All Classes

AbortProcessingException
AbstractMarshallerImpl
AbstractUnmarshalerImpl
AccessLocalException
ActionEvent
ActionListener
ActionSource
ActionSource2
ActionType
ActivationConfigProperty
ActivationDataFlavor
ActivationSpec
Address
AddressException
AddressStringTerm
AddressTerm
AndTerm
Application
ApplicationException
ApplicationFactory
ApplicationServerInternalException
AroundInvoke
ArrayDataModel
ArrayELResolver
Association
AssociationOverride
AssociationOverrides
AsyncHandler
AttachmentMarshaller
AttachmentPart
AttachmentUnmarshaler
Attribute
AttributeOverride
AttributeOverrides
AttributeTag
AuditableEvent
AuthenticationFailedException
Authenticator
Basic
BeanELResolver
BeanELResolver.BeanProperties
BeanELResolver.BeanProperty
BeanNotFoundException
BigDecimalConverter
BigDecimalHolder
BigIntegerConverter
BigIntegerHolder
Binder
Binding
BindingProvider
BindingType
BodyContent
BodyPart
BodyTag
BodyTagSupport
BodyTerm
BooleanConverter
BooleanHolder
BooleanWrapperHolder
BootstrapContext
BoundaryStatistic
BoundedRangeStatistic
BulkResponse
BusinessLifeCycleManager
BusinessQueryManager
ByteArrayDataSource
ByteArrayHolder
ByteConverter
ByteHolder
BytesMessage
ByteWrapperHolder
CalendarHolder
Call
CapabilityProfile
CascadeType
CharacterConverter
Characters
Classification
ClassificationScheme
ClassTransformer
ClientConfiguration
ClientExecuteException
CollapsedStringAdapter
Column
ColumnResult
CommandInfo
CommandMap
CommandObject
CommandType
Comment
CommException
ComparisonTerm
CompositeELResolver
Concept
ConcurrentAccessException
ConfigurationException
Connection
Connection
Connection
ConnectionAdapter
ConnectionFactory
ConnectionConsumer
ConnectionEvent
ConnectionEvent
ConnectionEventListener
ConnectionFactory
ConnectionFactory
ConnectionFactory
ConnectionListener
ConnectionManager
ConnectionMetaData
ConnectionMetaData
ConnectionRequestInfo
ConnectionSpec
ContentDisposition
ContentType
ContextCallback
Converter
ConverterELTag
ConverterException
ConverterTag
Cookie
CountStatistic
CreateException
DataContentHandler
DataContentHandlerFactory
DataHandler
DataModel
DataModelEvent
DataModelListener
DataSource
DatatypeConverter
DatatypeConverterInterface
DateTerm
DateTimeConverter
DConfigBean
DConfigBeanRoot
DConfigBeanVersionType
DConfigBeanVersionUnsupportedException
DDBean
DDBeanCreateException
DDBeanRoot
DeclarativeQueryManager
DeclareRoles
DefaultValidationEventHandler
DeleteException
DeliveryMode
DenyAll
DeployableObject
DeploymentConfiguration
DeploymentFactory
DeploymentFactoryManager
DeploymentManager
DeploymentManagerCreationException
DeploymentStatus
DeserializationContext
Deserializer
DeserializerFactory
Destination
Detail
DetailEntry
DiscriminatorColumn
DiscriminatorType
DiscriminatorValue
Dispatch
DissociatableManagedConnection
DomHandler
DoubleConverter
DoubleHolder
DoubleRangeValidator
DoubleWrapperHolder
DTD
DuplicateKeyException
DynamicAttributes
EditableValueHolder
EISSystemException
EJB
EJBAccessException
EJBContext
EJBException
EJBHome
EJBLocalHome
EJBLocalObject
EJBMetaData
EJBMethodPermission
EJBObject
EJBRoleRefPermission
EJBs
EJBStats
EJBTransactionRequiredException
EJBTransactionRolledbackException
ELContext
ELContextEvent
ELContextListener
Element
ELException
ELParseException
ELResolver
EmailAddress
Embeddable
Embedded
EmbeddedId
EndDocument
EndElement
Endpoint
EnterpriseBean
Entity
EntityBean
EntityBeanStats
EntityContext
EntityDeclaration
EntityExistsException
EntityListeners
EntityManager
EntityManagerFactory
EntityNotFoundException
EntityReference
EntityResult
EntityTransaction
EnumConverter
Enumerated
EnumType
ErrorData
EvaluationException
EventFilter
EventReaderDelegate
ExceptionListener
ExcludeClassInterceptors
ExcludeDefaultInterceptors
ExcludeDefaultListeners
ExcludeSuperclassListeners
ExecutionContext
Expression
Expression
ExpressionEvaluator
ExpressionFactory
ExtensibleObject
ExternalContext
ExternalIdentifier
ExternalLink
ExtrinsicObject
FacesContext
FacesContextFactory
FacesEvent
FacesException
FacesListener
FacesMessage
FacesMessage.Severity
FacesServlet
FacetTag
FactoryConfigurationError
FactoryFinder
FederatedConnection
FetchProfile
FetchProfile.Item
FetchType
FieldResult
FileDataSource
FileTypeMap
Filter
FilterChain
FilterConfig
FinderException
FindException
FindQualifier
Flags
Flags.Flag
FlagTerm
FloatConverter
FloatHolder
FloatWrapperHolder
FlushModeType
Folder
FolderAdapter
FolderClosedException
FolderEvent
FolderListener
FolderNotFoundException
FromStringTerm
FromTerm
FunctionInfo
FunctionMapper
FunctionMapper
Generated
GeneratedValue
GenerationType
GenericCredential
GenericHandler
GenericServlet
Handle
HandleDelegate
Handler
Handler
HandlerChain
HandlerChain
HandlerInfo
HandlerRegistry
HandlerResolver
Header
HeaderTerm
HeaderTokenizer
HeaderTokenizer.Token
HeuristicCommitException
HeuristicMixedException
HeuristicRollbackException
HexBinaryAdapter
Holder
Holder
HomeHandle
HtmlColumn
HtmlCommandButton
HtmlCommandLink
HtmlDataTable
HtmlForm
HtmlGraphicImage
HtmlInputHidden
HtmlInputSecret
HtmlInputText
HtmlInputTextarea
HtmlMessage
HtmlMessages
HtmlOutputFormat
HtmlOutputLabel
HtmlOutputLink
HtmlOutputText
HtmlPanelGrid
HtmlPanelGroup
HtmlSelectBooleanCheckbox
HtmlSelectManyCheckbox
HtmlSelectManyListbox
HtmlSelectManyMenu
HtmlSelectOneListbox
HtmlSelectOneMenu
HtmlSelectOneRadio
HTTPBinding
HttpException
HttpJspPage
HttpServlet
HttpServletRequest
HttpServletRequestWrapper
HttpServletResponse
HttpServletResponseWrapper
HttpSession
HttpSessionActivationListener
HttpSessionAttributeListener
HttpSessionBindingEvent
HttpSessionBindingListener
HttpSessionContext
HttpSessionEvent
HttpSessionListener
HttpUtils
Id
IdClass
IllegalStateException
IllegalStateException
IllegalWriteException
ImplicitObjectELResolver
IndexedRecord
Inheritance
InheritanceType
Init
InitParam
IntegerComparisonTerm
IntegerConverter
IntegerWrapperHolder
Interaction
InteractionSpec
Interceptors
InternationalString
InternetAddress
InternetHeaders
InternetHeaders.InternetHeader
IntHolder
InvalidClientIDException
InvalidDestinationException
InvalidModuleException
InvalidPropertyException
InvalidRequestException
InvalidSelectorException
InvalidTransactionException
InvocationContext
IterationTag
J2eeApplicationObject
JavaMailStats
JAXBContext
JAXBElement
JAXBElement.GlobalScope
JAXBException
JAXBIntrospector
JAXBResult
JAXBSource
JAXRException
JAXRPCException
JAXRRResponse
JCACreationPoolStats
JCAConnectionStats
JCAStats
JDBCConnectionPoolStats
JDBCConnectionStats
JDBCStats
JMSConnectionStats
JMSCoreConsumerStats
JEMSEndpointStats
JMSEException
JMSPublisherStats
JMSStats
JoinColumn
JoinColumn
JoinColumns
JoinTable
JspApplicationContext
JspContext
JspEngineInfo
JspException
JspFactory
JspFragment
JspIdConsumer
JspPage
JspTag
JspTagException
JspWriter
JTAServerStats
JVMStats
Key
LazyAssociatableConnectionManager
LazyEnlistableConnectionManager
LazyEnlistableManagedConnection
LengthValidator
Lifecycle
LifecycleFactory
LifeCycleManager
ListDataModel
ListELResolver
ListenerRegistration
Lob
Local
LocalHome
LocalizedString
LocalTransaction
LocalTransaction
LocalTransactionException
Location
LockModeType
LogicalHandler
LogicalMessage
LogicalMessageContext
LongConverter
LongHolder
LongRangeValidator
LongWrapperHolder
MailcapCommandMap
MailDateFormat
MailEvent
ManagedConnection
ManagedConnectionFactory
ManagedConnectionMetaData
Management
ManagementHome
ManyToMany
ManyToMany
ManyToOne
MapELResolver
MapKey
MapMessage
MappedRecord
MappedSuperclass
MarshalException
Marshaller
Marshaller.Listener
Message
Message
Message.RecipientType
MessageAware
MessageChangedEvent
MessageChangedListener
MessageConsumer
MessageContext
MessageContext
MessageContext
MessageContext.Scope
MessageCountAdapter
MessageCountEvent
MessageCountListener
MessageDriven
MessageDrivenBean
MessageDrivenBeanStats
MessageDrivenContext
MessageEndpoint
MessageEndpointFactory
MessageEOFException
MessageFactory
MessageFormatException
MessageIDTerm
MessageListener
MessageListener
MessageNotReadableException
MessageNotWriteableException
MessageNumberTerm
MessageProducer
MessageRemovedException
MessagingException
MethodBinding
MethodExpression
MethodExpressionActionListener
MethodExpressionValidator
MethodExpressionValueChangeListener
MethodInfo
MethodNotFoundException
MethodNotFoundException
MethodNotSupportedException
MimeBodyPart
MimeHeader
MimeHeaders
MimeMessage
MimeMessage.RecipientType
MimeMultipart
MimePart
MimePartDataSource
MimeType
MimeTypeParameterList
MimeTypeParseException
MimetypesFileTypeMap
MimeUtility
ModuleType
Multipart
MultipartDataSource
Name
NamedNativeQueries
NamedNativeQuery
NamedQueries
NamedQuery
Namespace
NamespaceConstants
NamingContainer
NavigationHandler
NewsAddress
Node
NonUniqueResultException
NoResultException
NormalizedStringAdapter
NoSuchEJBException
NoSuchEntityException
NoSuchObjectLocalException
NoSuchProviderException
NotationDeclaration
NotIdentifiableEvent
NotIdentifiableEventImpl
NotSupportedException
NotSupportedException
NotTerm
NumberConverter
ObjectHolder
ObjectMessage
ObjectNotFoundException
OneToMany
OneToOne
Oneway
OperationUnsupportedException
OptimisticLockException
OrderBy
Organization
OrTerm
PageContext
PageData
ParameterList
ParameterMode
ParseConversionEvent
ParseConversionEventImpl
ParseException
Part
PasswordAuthentication
PasswordCredential
PermitAll
Persistence
PersistenceContext
PersistenceContexts
PersistenceContextType
PersistenceException
PersistenceProperty
PersistenceProvider
PersistenceUnit
PersistenceUnitInfo
PersistenceUnits
PersistenceUnitTransactionType
PersonName
PhaseEvent
Phaseld
PhaseListener
PolicyConfiguration
PolicyConfigurationFactory
PolicyContext
PolicyContextException
PolicyContextHandler
PortInfo
PostActivate
PostalAddress
PostConstruct
PostLoad
PostPersist
PostRemove
PostUpdate
PreDestroy
PreencodedMimeBodyPart
PrePassivate
PrePersist
PreRemove
PreUpdate
PrimaryKeyJoinColumn
PrimaryKeyJoinColumns
PrintConversionEvent
PrintConversionEventImpl
ProcessingInstruction
ProgressEvent
ProgressListener
ProgressObject
PropertyException
PropertyNotFoundException
PropertyNot FoundException
PropertyNotWritableException
PropertyResolver
ProtocolException
Provider
Provider
Provider
Provider.Type
QNameHolder
Query
Query
QueryHint
QueryManager
Queue
QueueBrowser
QueueConnection
QueueConnectionFactory
QueueReceiver
QueueRequestor
QueueSender
QueueSession
Quota
Quota.Resource
QuotaAwareStore
RangeStatistic
ReadOnlyFolderException
ReceivedDateTerm
RecipientStringTerm
RecipientTerm
Record
RecordFactory
Referenceable
ReferenceSyntaxException
RegistryEntry
RegistryException
RegistryObject
RegistryPackage
RegistryService
Remote
RemoteHome
Remove
RemoveException
Renderer
RenderKit
RenderKitFactory
RequestDispatcher
RequestWrapper
Resource
Resource.AuthenticationType
ResourceAdapter
ResourceAdapterAssociation
ResourceAdapterInternalException
ResourceAdapterMetaData
ResourceAllocationException
ResourceAllocationException
ResourceBundleELResolver
ResourceException
Resources
ResourceWarning
Response
ResponseStateManager
ResponseStream
ResponseWrapper
ResponseWriter
ResponseWriterWrapper
ResultDataModel
ResultSet
ResultSetDataModel
ResultSetInfo
RolesAllowed
RollbackException
RollbackException
RunAs
SAAJMetaFactory
SAAJResult
SaveException
ScalarDataModel
SchemaOutputResolver
ScopedAttributeELResolver
SearchException
SearchTerm
SecondaryTable
SecondaryTables
SecurityException
ServletResponseWrapper
ServletStats
Session
Session
SessionBean
SessionBeanStats
SessionContext
SessionSynchronization
SharedByteArrayInputStream
SharedFileInputStream
SharedInputStream
SharingViolationException
ShortConverter
ShortHolder
ShortWrapperHolder
SimpleTag
SimpleTagSupport
SimpleThreadModel
SizeTerm
SkipPageException
Slot
SOAPBinding
SOAPBinding
SOAPBinding.ParameterStyle
SOAPBinding.Style
SOAPBinding.Use
SOAPBody
SOAPBodyElement
SOAPConnection
SOAPConnectionFactory
SOAPConstants
SOAPElement
SOAPElementFactory
SOAPEnvelope
SOAPException
SOAPFactory
SOAPFault
SOAPFaultElement
SOAPFaultException
SOAPFaultException
SOAPHandler
SOAPHeader
SOAPHeaderElement
SOAPMessage
SOAPMessageContext
SOAPMessageContext
SOAPMessageHandler
SOAPMessageHandlers
SOAPPart
SpecificationLink
SqlResultSetMapping
SqlResultSetMappings
StartDocument
StartElement
Stateful
StatefulSessionBeanStats
StateHolder
Stateless
StatelessSessionBeanStats
StateManager
StateManagerWrapper
StateType
Statistic
Stats
Status
Store
StoreClosedException
StoreEvent
StoreListener
Streamable
StreamFilter
StreamMessage
StreamReaderDelegate
StringHolder
StringTerm
Stub
SubjectTerm
Synchronization
SystemException
Table
TableGenerator
Tag
TagAdapter
TagAttributeInfo
TagData
TagExtraInfo
TagFileInfo
TagInfo
TagLibraryInfo
TagLibraryValidator
TagSupport
TagVariableInfo
Target
TargetException
TargetModuleID
TelephoneNumber
Temporal
TemporalType
TemporaryQueue
TemporaryTopic
Text
TextMessage
TimedObject
Timeout
Timer
TimerHandle
TimerService
TimeStatistic
Topic
TopicConnection
TopicConnectionFactory
TopicPublisher
TopicRequestor
TopicSession
TopicSubscriber
Transaction
TransactionAttribute
TransactionAttributeType
TransactionInProgressException
TransactionManagement
TransactionManagementType
TransactionManager
TransactionRequiredException
TransactionRequiredException
TransactionRequiredLocalException
TransactionRolledBackException
TransactionRolledbackException
TransactionRolledbackLocalException
TransactionSynchronizationRegistry
Transient
Transport
TransportAdapter
TransportEvent
TransportListener
TryCatchFinally
TypeConstraintException
TypeMapping
TypeMappingRegistry
UIColumn
UICommand
UIComponent
UIComponentBase
UIComponentBodyTag
UIComponentClassicTagBase
UIComponentELTag
UIComponentTag
UIComponentTagBase
UIData
UIDFolder
UIDFolder.FetchProfileItem
UIForm
UIGraphic
UIInput
UIMessage
UIMessages
UINamingContainer
ValueBinding
ValueChangeEvent
ValueChangeListener
ValueExpression
ValueHolder
VariableInfo
VariableMapper
VariableResolver
VariableResolver
Version
Versionable
ViewExpiredException
ViewHandler
ViewHandlerWrapper
W3CDomHandler
WebEndpoint
WebFault
WebMethod
WebParam
WebParam.Mode
WebResourcePermission
WebResult
WebRoleRefPermission
WebService
WebServiceClient
WebServiceContext
WebServiceException
WebServicePermission
WebServiceProvider
WebServiceRef
WebServiceRefs
WebUserDataPermission
Work
WorkAdapter
WorkCompletedException
WorkEvent
WorkException
WorkListener
WorkManager
WorkRejectedException
XAConnection
XAConnectionFactory
XAException
XAQueueConnection
XAQueueConnectionFactory
XAQueueSession
XAResource
XASession
XATerminator
XATopicConnection
XATopicConnectionFactory
XATopicSession
Xid
XmlAccessOrder
XmlAccessorOrder
XmlAccessorType
XmlAccessType
XmlAdapter
XmlAnyAttribute
XmlAnyElement
XmlAttachmentRef
XmlAttribute
XmlElement
XmlElement.DEF
XmlElementDecl
XmlElementDecl.GLOBAL
XmlElementRef
XmlElementRef.DEF
XmlElementRefs
XmlElement
XmlElements
XmlElementWrapper
XmlEnum
XmlEnumValue
XMLEvent
XMLEventAllocator
XMLEventConsumer
XMLEventFactory
XMLEventReader
XMLEventWriter
XmlID
XmlIDREF
XmlInlineBinaryData
XMLInputFactory
XmlJavaTypeAdapter
XmlJavaTypeAdapter.DEFAULT
XmlJavaTypeAdapters
XmlList
XmlMimeType
XmlMixed
XmlNs
XmlNsForm
XMLOutputFactory
XmlRegistry
XMLReporter
XMLResolver
XmlRootElement
XmlSchema
XmlSchemaType
XmlSchemaType.DEFAULT
XmlSchemaTypes
XMLStreamConstants
XMLStreamException
XMLStreamReader
XMLStreamWriter
XmlTransient
XmlType
XMLType
XmlType.DEFAULT
XmlValue
XpathEvent
XpathListener
Class AbortProcessingException

An exception that may be thrown by event listeners to terminate the processing of the current event.

See Also:
Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AbortProcessingException()</td>
<td>Construct a new exception with no detail message or root cause.</td>
</tr>
<tr>
<td>AbortProcessingException(String message)</td>
<td>Construct a new exception with the specified detail message and no root cause.</td>
</tr>
<tr>
<td>AbortProcessingException(String message, Throwable cause)</td>
<td></td>
</tr>
</tbody>
</table>
Construct a new exception with the specified detail message and root cause.

**AbortProcessingException** *(Throwable cause)*

Construct a new exception with the specified root cause.

### Method Summary

Methods inherited from class `javax.faces.FacesException`

- `getCause`

Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Constructor Detail

**public AbortProcessingException()**

**AbortProcessingException**

**public AbortProcessingException()**

Construct a new exception with no detail message or root cause.
public AbortProcessingException(String message)

message

AbortProcessingException

public AbortProcessingException(String message)

Construct a new exception with the specified detail message and no root cause.

Parameters:
message - The detail message for this exception

public AbortProcessingException(Throwable cause)

cause

AbortProcessingException

public AbortProcessingException(Throwable cause)

Construct a new exception with the specified root cause.

Parameters:
cause - The root cause for this exception

public AbortProcessingException(String message, Throwable cause)
AbortProcessingException

public AbortProcessingException(String message, Throwable cause)

Construct a new exception with the specified detail message and root cause.

Parameters:

message - The detail message for this exception
cause - The root cause for this exception

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.bind.helpers **Class AbstractMarshallerImpl**

**java.lang.Object**  
└── javax.xml.bind.helpers.AbstractMarshallerImpl

**All Implemented Interfaces:**  
Marshaller

---

```java
public abstract class AbstractMarshallerImpl
extends Object
implements Marshaller
```

**Implements:** Marshaller

```java
Marshaller
```

JAXB  
```
marshal(Object, javax.xml.transform.Result)  
marshal(Object, javax.xml.stream.XMLStreamWriter)  
marshal(Object, javax.xml.stream.XMLEventWriter)
```

**version**  
$Revision: 1.7 $ $Date: 2006/03/08 17:00:39$

**since**  
JAXB1.0

**See also**  
javax.xml.bind.Marshaller

---

Partial default Marshaller implementation.

This class provides a partial default implementation for the Marshaller interface.

The only methods that a JAXB Provider has to implement are  
marshal(Object, javax.xml.transform.Result), marshal(Object, javax.xml.stream.XMLStreamWriter), and marshal(Object, javax.xml.stream.XMLEventWriter).
Nested Class Summary

Nested classes/interfaces inherited from interface javax.xml.bind.Marshaller
Marshaller.Listener

Field Summary

Fields inherited from interface javax.xml.bind.Marshaller
JAXB_ENCODING, JAXB_FORMATTED_OUTPUT, JAXB_FRAGMENT,
JAXB_NO_NAMESPACE_SCHEMA_LOCATION, JAXB_SCHEMA_LOCATION

Constructor Summary

AbstractMarshallerImpl()

Method Summary

getAdapter(Class type)
    Gets the adapter associated with the specified type.

AttachmentMarshaller getAttachmentMarshaller()
    protected String getEncoding()
    Convenience method for getting the current output encoding.
ValidationEventHandler getEventHandler()
Return the current event handler or the default event handler if one hasn't been set. protected String getJavaEncoding(String encoding)
Gets the corresponding Java encoding name from an IANA name.
Marshaller.Listener getListener()
Return Marshaller.Listener registered with this Marshaller.

getNode(Object obj)
By default, the getNode method is unsupported and throw an UnsupportedOperationException. protected String getNoNSSchemaLocation()
Convenience method for getting the current noNamespaceSchemaLocation. Object getProperty(String name)
Default implementation of the getProperty method handles the four defined properties in Marshaller. Schema getSchema()
Get the JAXP 1.3 Schema object being used to perform marshal-time validation. protected String getSchemaLocation()
Convenience method for getting the current schemaLocation. protected boolean isFormattedOutput()
Convenience method for getting the formatted output flag. protected boolean isFragment()
Convenience method for getting the fragment flag.

void marshal(Object obj, ContentHandler handler)
Marshal the content tree rooted at jaxbElement into SAX2 events.
void marshal(Object obj, Node node)
Marshal the content tree rooted at jaxbElement into a DOM tree.
void marshal(Object obj, OutputStream os)
Marshal the content tree rooted at jaxbElement into an output stream. void marshal(Object obj, Writer w)
Marshal the content tree rooted at jaxbElement into a Writer.
void marshal(Object obj, XMLEventWriter writer)
Marshal the content tree rooted at jaxbElement into a XMLEventWriter.
void marshal(Object obj, XMLStreamWriter writer)
Marshal the content tree rooted at jaxbElement into a XMLStreamWriter.

void setAdapter(Class<A> type, A adapter)
Associates a configured instance of XmlAdapter with this marshaller.
Associates a configured instance of `XmlAdapter` with this marshaller. void `setAttachmentMarshaller(AttachmentMarshaller am)`
Associate a context that enables binary data within an XML document to be transmitted as XML-binary optimized attachment. protected void `setEncoding(String encoding)`
Convenience method for setting the output encoding. void `setEventHandler(ValidationEventHandler handler)`
Allow an application to register a validation event handler. protected void `setFormattedOutput(boolean v)`
Convenience method for setting the formatted output flag. protected void `setFragment(boolean v)`
Convenience method for setting the fragment flag. void `setListener(Marshaller.Listener listener)`
Register marshal event callback `Marshaller.Listener` with this `Marshaller`. protected void `setNoNSSchemaLocation(String location)`
Convenience method for setting the noNamespaceSchemaLocation. void `setProperty(String name, Object value)`
Default implementation of the setProperty method handles the four defined properties in Marshaller. void `setSchema(Schema schema)`
Specify the JAXP 1.3 `Schema` object that should be used to validate subsequent marshal operations against. protected void `setSchemaLocation(String location)`
Convenience method for setting the schemaLocation.

### Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Methods inherited from interface javax.xml.bind.Marshaller

`marshal`

### Constructor Detail

`public AbstractMarshallerImpl()`
AbstractMarshallerImpl

public AbstractMarshallerImpl()

### Method Detail

**final public void marshal(Object obj, java.io.OutputStream os) throws JAXBException**

**marshal**

```java
public final void marshal(Object obj, OutputStream os)
throws JAXBException
```

Description copied from interface: Marshaller

Marshal the content tree rooted at jaxbElement into an output stream.

**Specified by:**

marshaled in interface Marshaller

**Parameters:**

- **obj** - The root of content tree to be marshalled.
- **os** - XML will be added to this stream.

**Throws:**

- **JAXBException** - If any unexpected problem occurs during the marshalling.
- **MarshallerException** - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.

**final public void marshal(Object obj, java.io.Writer w) throws JAXBException**
marshal

```java
public final void marshal(Object obj,
                          Writer w)
    throws JAXBException
```

**Description copied from interface:** Marshaller
Marshal the content tree rooted at jaxbElement into a Writer.

**Specified by:**
- `marshal` in interface Marshaller

**Parameters:**
- `obj` - The root of content tree to be marshalled.
- `w` - XML will be sent to this writer.

**Throws:**
- `JAXBException` - If any unexpected problem occurs during the marshalling.
- `MarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Marshaller` is unable to marshal `obj` (or any object reachable from `obj`). See [Marshalling a JAXB element](https://docs.oracle.com/javase/8/docs/api/javax/xml/bind/Marshalling.html).

---

**final public void marshal(Object obj,**
```java
org.xml.sax.ContentHandler handler)
    throws JAXBException
```

**marshal**

```java
public final void marshal(Object obj,
                          ContentHandler handler)
    throws JAXBException
```

**Description copied from interface:** Marshaller
Marshal the content tree rooted at jaxbElement into SAX2 events.

**Specified by:**
- `marshal` in interface Marshaller
Parameters:
obj - The root of content tree to be marshalled.
handler - XML will be sent to this handler as SAX2 events.

Throws:
JAXBException - If any unexpected problem occurs during the marshalling.
MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.

final public void marshal(Object obj, org.w3c.dom.Node node) throws JAXBException

marshal

public final void marshal(Object obj,
Node node)
throws JAXBException

Description copied from interface: Marshaller
Marshal the content tree rooted at jaxbElement into a DOM tree.

Specified by:
marshal in interface Marshaller

Parameters:
obj - The content tree to be marshalled.
node - DOM nodes will be added as children of this node. This parameter must be a Node that accepts children (Document, DocumentFragment, or Element)

Throws:
JAXBException - If any unexpected problem occurs during the marshalling.
MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal jaxbElement (or any object reachable from jaxbElement). See Marshalling a JAXB element.
public org.w3c.dom.Node getNode(Object obj) throws JAXBException
getNode

By default, the getNode method is unsupported and throw an UnsupportedOperationException. Implementations that choose to support this method must override this method.

Specified by: getNode in interface Marshaller
Parameters: obj -- JAXB Java representation of XML content
Returns: the DOM tree view of the contentTree
Throws: JAXBException - If any unexpected problem occurs

protected String getEncoding()

return "UTF-8"

getEncoding

protected String getEncoding()

Convenience method for getting the current output encoding.

Returns: the current encoding or "UTF-8" if it hasn't been set.
protected void setEncoding(String encoding)

    encoding Marshaller

setEncoding

protected void setEncoding(String encoding)

    Convenience method for setting the output encoding.

    Parameters:
    encoding - a valid encoding as specified in the Marshaller class documentation

protected String getSchemaLocation()

    return schemaLocation schemaLocation null

getSchemaLocation

protected String getSchemaLocation()

    Convenience method for getting the current schemaLocation.

    Returns:
    the current schemaLocation or null if it hasn't been set

protected void setSchemaLocation(String location)

    location schemaLocation
setSchemaLocation

protected void setSchemaLocation(String location)

   Convenience method for setting the schemaLocation.

   **Parameters:**
   location - the schemaLocation value

getNoNSSchemaLocation

protected String getNoNSSchemaLocation()

   return noNamespaceSchemaLocation
   noNamespaceSchemaLocation null

getNoNSSchemaLocation

protected String getNoNSSchemaLocation()

   Convenience method for getting the current noNamespaceSchemaLocation.

   **Returns:**
   the current noNamespaceSchemaLocation or null if it hasn't been set

setNoNSSchemaLocation

protected void setNoNSSchemaLocation(String location)

   location noNamespaceSchemaLocation

setNoNSSchemaLocation

protected void setNoNSSchemaLocation(String location)

   Convenience method for setting the noNamespaceSchemaLocation.
protected boolean isFormattedOutput()

    return false

isFormattedOutput

protected boolean isFormattedOutput()

    Convenience method for getting the formatted output flag.

    Returns:
    the current value of the formatted output flag or false if it hasn't been set.

protected void setFormattedOutput(boolean v)

    v

setFormattedOutput

protected void setFormattedOutput(boolean v)

    Convenience method for setting the formatted output flag.

    Parameters:
    v - value of the formatted output flag.
**isFragment**

protected boolean isFragment()

Convenience method for getting the fragment flag.

**Returns:**
the current value of the fragment flag or false if it hasn't been set.

---

**protected void setFragment(boolean v)**

v

**setFragment**

protected void setFragment(boolean v)

Convenience method for setting the fragment flag.

**Parameters:**
v - value of the fragment flag.

---

protected String getJavaEncoding(String encoding) throws java.io.UnsupportedEncodingException
IANA Java helper

Throws java.io.UnsupportedEncodingException: Java

**getJavaEncoding**

protected String getJavaEncoding(String encoding)
throws UnsupportedEncodingException

Gets the corresponding Java encoding name from an IANA name. This method is a helper method for the derived class to convert encoding names.

**Throws:**

UnsupportedEncodingException - If this implementation couldn't find the Java encoding name.

---

```java
public void setProperty(String name, Object value) throws PropertyException

setProperty Marshaller 4
```

**setProperty**

```java
public void setProperty(String name, Object value)
    throws PropertyException
```

Default implementation of the setProperty method handles the four defined properties in Marshaller. If a provider needs to handle additional properties, it should override this method in a derived class.

**Specified by:**

setProperty in interface Marshaller

**Parameters:**

- **name** - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
- **value** - the value of the property to be set

**Throws:**

PropertyException - when there is an error processing the given property or value
public Object getProperty(String name) throws PropertyException

getProperty

public Object getProperty(String name)
    throws PropertyException

Default implementation of the getProperty method handles the four defined properties in Marshaller. If a provider needs to support additional provider specific properties, it should override this method in a derived class.

Specified by:
    getProperty in interface Marshaller

Parameters:
    name - the name of the property to retrieve

Returns:
    the value of the requested property

Throws:
    PropertyException - when there is an error retrieving the given property or value property name

------------------------------------------------------------------------------

public ValidationEventHandler getEventHandler() throws JAXBException

See also
    getEventHandler()

ggetEventHandler

public ValidationEventHandler getEventHandler()
    throws JAXBException

Description copied from interface: Marshaller
Return the current event handler or the default event handler if one
hasn't been set.

Specified by:
  getEventHandler in interface Marshaller

Returns:
  the current ValidationEventHandler or the default event handler
  if it hasn't been set

Throws:
  JAXBException - if an error was encountered while getting the
  current event handler

See Also:
  Marshaller.getEventHandler()

public void setEventHandler(ValidationEventHandler handler) throws JAXBException
See also  setEventHandler(ValidationEventHandler)

setEventHandler

public void setEventHandler(ValidationEventHandler handler) throws JAXBException

Description copied from interface: Marshaller
Allow an application to register a validation event handler.

The validation event handler will be called by the JAXB Provider if
any validation errors are encountered during calls to any of the
marshal API's. If the client application does not register a validation
event handler before invoking one of the marshal methods, then
validation events will be handled by the default event handler which
will terminate the marshal operation after the first error or fatal error
is encountered.

Calling this method with a null parameter will cause the Marshaller to
revert back to the default default event handler.

Specified by:
**setEventHandler** in interface `Marshaller`

**Parameters:**
- handler - the validation event handler

**Throws:**
- `JAXBException` - if an error was encountered while setting the event handler

**See Also:**
- `Marshaller.setEventHandler(ValidationEventHandler)`

---

```java
public void marshal(Object obj, XMLEventWriter writer) throws JAXBException
```

**marshal**

```java
public void marshal(Object obj,
                    XMLEventWriter writer)
    throws JAXBException
```

**Description copied from interface:** `Marshaller`

Marshal the content tree rooted at `jaxbElement` into a `XMLEventWriter`.

**Specified by:**
- `marshal` in interface `Marshaller`

**Parameters:**
- `obj` - The content tree rooted at `jaxbElement` to be marshalled.
- `writer` - XML will be sent to this writer.

**Throws:**
- `JAXBException` - If any unexpected problem occurs during the marshalling.
- `MarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Marshaller` is unable to marshal `obj` (or any object reachable from `obj`). See [Marshalling a JAXB element](https://docs.oracle.com/javase/8/docs/api/javax/xml/bind/Marshaller.html).

---

```java
public void marshal(Object obj, XMLStreamWriter writer)
```
public void marshal(Object obj, XMLStreamWriter writer) throws JAXBException

Description copied from interface: Marshaller
Marshals the content tree rooted at jaxbElement into a XMLStreamWriter.

Specified by:
marshal in interface Marshaller
Parameters:
   obj - The content tree to be marshalled.
   writer - XML will be sent to this writer.
Throws:
   JAXBException - If any unexpected problem occurs during the marshalling.
   MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.

public void setSchema(javax.xml.validation.Schema schema)

setSchema

public void setSchema(Schema schema)

Description copied from interface: Marshaller
Specify the JAXP 1.3 Schema object that should be used to validate subsequent marshal operations against. Passing null into this method will disable validation.
This method allows the caller to validate the marshalled XML as it's marshalled.

Initially this property is set to null.

**Specified by:**

```
setSchema in interface Marshaller
```

**Parameters:**

- `schema` - Schema object to validate marshal operations against or null to disable validation

---

```
public javax.xml.validation.Schema getSchema()
```

**getSchema**

```
public Schema getSchema()
```

**Description copied from interface:** [Marshaller](http://docs.oracle.com/javase/8/docs/api/javax/xml/validation/Schema.html)

Get the JAXP 1.3 [Schema](http://docs.oracle.com/javase/8/docs/api/javax/xml/validation/Schema.html) object being used to perform marshal-time validation. If there is no Schema set on the marshaller, then this method will return null indicating that marshal-time validation will not be performed.

**Specified by:**

```
getSchema in interface Marshaller
```

**Returns:**

- the Schema object being used to perform marshal-time validation or null if not present.

---

```
public void setAdapter("">XmlAdapter adapter)
```

**setAdapter**

```
public void setAdapter(XmlAdapter adapter)
```
Description copied from interface: *Marshaller*
Associates a configured instance of *XmlAdapter* with this marshaller.

This is a convenience method that invokes
setAdapter(adapter.getClass(), adapter);

Specified by:
  *setAdapter* in interface *Marshaller*

See Also:
  *Marshaller.setAdapter(Class,XmlAdapter)*

---

setAdapter

```java
public <A extends XmlAdapter> void setAdapter(Class<A> type, A adapter)
```

Description copied from interface: *Marshaller*
Associates a configured instance of *XmlAdapter* with this marshaller.

Every marshaller internally maintains a *Map<Class,XmlAdapter>*,
which it uses for marshalling classes whose fields/methods are
annotated with *XmlJavaTypeAdapter*.

This method allows applications to use a configured instance of
*XmlAdapter*. When an instance of an adapter is not given, a
marshaller will create one by invoking its default constructor.

Specified by:
  *setAdapter* in interface *Marshaller*

Parameters:
  - type - The type of the adapter. The specified instance will be
  used when *XmlJavaTypeAdapter.value()* refers to this type.
  - adapter - The instance of the adapter to be used. If null, it will
  un-register the current adapter set for this type.
getAdapter

public <A extends XmlAdapter> A getAdapter(Class<A> type)

Description copied from interface: Marshaller
Gets the adapter associated with the specified type. This is the reverse operation of the Marshaller.setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter) method.

Specified by:
getAdapter in interface Marshaller

public void setAttachmentMarshaller(AttachmentMarshaller am)

setAttachmentMarshaller

public void setAttachmentMarshaller(AttachmentMarshaller am)

Description copied from interface: Marshaller
Associate a context that enables binary data within an XML document to be transmitted as XML-binary optimized attachment. The attachment is referenced from the XML document content model by content-id URIs(cid) references stored within the xml document.

Specified by:
setAttachmentMarshaller in interface Marshaller

public AttachmentMarshaller getAttachmentMarshaller()

getAttachmentMarshaller

public AttachmentMarshaller getAttachmentMarshaller()
public void setListener(Marshaller.Listener listener)

setListener

public void setListener(Marshaller.Listener listener)

Description copied from interface: Marshaller

Register marshal event callback Marshaller.Listener with this Marshaller.

There is only one Listener per Marshaller. Setting a Listener replaces the previous set Listener. One can unregister current Listener by setting listener to null.

Specified by:
setListener in interface Marshaller

Parameters:
listener - an instance of a class that implements Marshaller.Listener

public Marshaller.Listener getListener()

getListener

public Marshaller.Listener getListener()

Description copied from interface: Marshaller

Return Marshaller.Listener registered with this Marshaller.

Specified by:
getListener in interface Marshall

Returns:
registered Marshall.Listener or null if no Listener is registered with this Marshaller.
javax.xml.bind.helpers  **Class AbstractUnmarshallerImpl**

**java.lang.Object**  
   - javax.xml.bind.helpers.AbstractUnmarshallerImpl

**All Implemented Interfaces:**  
   - Unmarshaller

---

```java
public abstract class AbstractUnmarshallerImpl
extends Object
implements Unmarshaller

Implements: Unmarshaller

Unmarshaller

javax.xml.bind.Unmarshaller

JAXB 5 getUnmarshallerHandler
unmarshal(Node)
unmarshal(XMLReader, InputSource)
unmarshal(XMLStreamReader)
unmarshal(XMLEventReader)
```

**version**  
$Revision: 1.14 $  
$Date: 2006/03/08 17:01:00 $

**since**  
JAXB1.0

**See also**  
javax.xml.bind.Unmarshaller

---

Partial default Unmarshaller implementation.

This class provides a partial default implementation for the Unmarshaller interface.

A JAXB Provider has to implement five methods  
(getUnmarshallerHandler, unmarshal(Node),
unmarshal(XMLReader, InputSource), unmarshal(XMLStreamReader),
and unmarshal(XMLEventReader).
## Nested Class Summary

### Nested classes/interfaces inherited from interface javax.xml.bind.Unmarshaller

- Unmarshaller.Listener

## Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected boolean</td>
<td><code>validating</code></td>
<td>whether or not the unmarshaller will validate</td>
</tr>
</tbody>
</table>

## Constructor Summary

- `AbstractUnmarshallerImpl()`  

## Method Summary

- `createUnmarshalException(SAXException e)`  
  Creates an UnmarshalException from a SAXException.

- `getAdapter(Class<A> type)`  
  Gets the adapter associated with the specified type.

- `getAttachmentUnmarshaller()`  
  AttachmentUnmarshaller
ValidationEventHandler getEventHandler()  
Return the current event handler or the default event handler if one 
hasn't been set. Unmarshaller.Listener getListener()  
Return Unmarshaller.Listener registered with this Unmarshaller.
Object getProperty(String name)  
Default implementation of the getProperty method always throws PropertyException since there are no required properties. Schema
getSchema()  
Get the JAXP 1.3 Schema object being used to perform unmarshal-
time validation. protected XMLReader getXMLReader()  
Obtains a configured XMLReader. boolean isValidating()  
Indicates whether or not the Unmarshaller is configured to validate 
during unmarshal operations.

<A extends XmlAdapter>
void setAdapter(Class<A> type, A adapter)  
Associates a configured instance of XmlAdapter with this
unmarshaller. void setAdapter(XmlAdapter adapter)  
Associates a configured instance of XmlAdapter with this
unmarshaller. void setAttachmentUnmarshaller(AttachmentUnmarshaller au)  
Associate a context that resolves cid's, content-id URIs, to binary
data passed as attachments. void setEventHandler(ValidationEventHandler handler)  
Allow an application to register a validation event handler. void setListener(Unmarshaller.Listener listener)  
Register unmarshal event callback Unmarshaller.Listener with this Unmarshaller. void setProperty(String name, Object value)  
Default implementation of the setProperty method always throws PropertyException since there are no required properties. void setSchema(Schema schema)  
Specify the JAXP 1.3 Schema object that should be used to validate 
subsequent unmarshal operations against. void setValidating(boolean validating)  
Specifies whether or not the Unmarshaller should validate during 
unmarshal operations. Object unmarshal(File f)  
Unmarshal XML data from the specified file and return the resulting content tree. Object unmarshal(InputSource source)
Unmarshal XML data from the specified SAX InputSource and return the resulting content tree. 

Unmarshal XML data from the specified InputStream and return the resulting content tree.

```java
<T> JAXBElement<T> unmarshal(Node node, Class<T> expectedType)
```

Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.

```java
<T> JAXBElement<T> unmarshal(Reader reader)
```

Unmarshal XML data from the specified Reader and return the resulting content tree.

```java
<T> JAXBElement<T> unmarshal(Source source)
```

Unmarshal XML data from the specified XML Source and return the resulting content tree.

```java
<T> JAXBElement<T> unmarshal(Source source, Class<T> expectedType)
```

Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.

```java
URL url
```

Unmarshal XML data from the specified URL and return the resulting content tree.

```java
<T> JAXBElement<T> unmarshal(XMLEventReader reader)
```

Unmarshal XML data from the specified pull parser and return the resulting content tree.

```java
protected abstract Object unmarshal(XMLStreamReader reader, InputSource source)
```

Unmarshals an object by using the specified XMLReader and the InputSource.

```java
<T> JAXBElement<T> unmarshal(XMLStreamReader reader, Class<T> expectedType)
```

Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

**Methods inherited from interface javax.xml.bind.Unmarshaller**
Field Detail

protected boolean validating

whether or not the unmarshaller will validate

Constructor Detail

public AbstractUnmarshallerImpl()

Method Detail

protected org.xml.sax.XMLReader getXMLReader() throws JAXBException

XMLReader SAXSource XMLReader Unmarshaller XMLReader
Obtains a configured XMLReader. This method is used when the client-specified SAXSource object doesn't have XMLReader. Unmarshaller is not re-entrant, so we will only use one instance of XMLReader.

**Throws:**  
*JAXBException*

```java
public Object unmarshal(javax.xml.transform.Source source) throws JAXBException
```

**unmarshal**

```java
public Object unmarshal(Source source) throws JAXBException
```

**Description copied from interface:** Unmarshaller

Unmarshal XML data from the specified XML Source and return the resulting content tree.

Implements Unmarshal Global Root Element.

**SAX 2.0 Parser Pluggability**

A client application can choose not to use the default parser mechanism supplied with their JAXB provider. Any SAX 2.0 compliant parser can be substituted for the JAXB provider's default mechanism. To do so, the client application must properly configure a SAXSource containing an XMLReader implemented by the SAX 2.0 parser provider. If the XMLReader has an org.xml.sax.ErrorHandler registered on it, it will be replaced by the JAXB Provider so that validation errors can be reported via the ValidationEventHandler mechanism of JAXB. If the SAXSource does not contain an XMLReader, then the JAXB provider's default parser mechanism will be used.

This parser replacement mechanism can also be used to replace the JAXB provider's unmarshal-time validation engine. The client
application must properly configure their SAX 2.0 compliant parser to perform validation (as shown in the example above). Any SAXParserExceptions encountered by the parser during the unmarshal operation will be processed by the JAXB provider and converted into JAXB ValidationEvent objects which will be reported back to the client via the ValidationEventHandler registered with the Unmarshaller. Note: specifying a substitute validating SAX 2.0 parser for unmarshalling does not necessarily replace the validation engine used by the JAXB provider for performing on-demand validation.

The only way for a client application to specify an alternate parser mechanism to be used during unmarshal is via the unmarshal(SAXSource) API. All other forms of the unmarshal method (File, URL, Node, etc) will use the JAXB provider's default parser and validator mechanisms.

Specified by:  
unmarshal in interface Unmarshaller  

Parameters:  
source - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and StreamSource)

Returns:  
the newly created root object of the java content tree  

Throws:  
JAXBException - If any unexpected errors occur while unmarshalling  
UnmarshallerException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data  

See Also:  
Unmarshaller.unmarshal(javax.xml.transform.Source, Class)  

abstract protected Object  
unmarshal(org.xml.sax.XMLReader reader,  
org.xml.sax.InputSource source) throws JAXBException  
XMLReader  InputSource
Unmarshals an object by using the specified XMLReader and the InputSource. The callee should call the setErrorHandler method of the XMLReader so that errors are passed to the client-specified ValidationEventHandler.

**Throws:**

*JAXBException*
Throws:

- `JAXBException` - If any unexpected errors occur while unmarshalling
- `UnmarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Unmarshaller` is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#).

```java
final public Object unmarshal(java.net.URL url) throws JAXBException
```

**unmarshal**

```java
public final Object unmarshal(URL url)
```

**Description copied from interface: Unmarshaller**

Unmarshal XML data from the specified URL and return the resulting content tree.

Implements [Unmarshal Global Root Element](#).

**Specified by:**

`unmarshal` in interface `Unmarshaller`

**Parameters:**

- `url` - the url to unmarshal XML data from

**Returns:**

the newly created root object of the java content tree

**Throws:**

- `JAXBException` - If any unexpected errors occur while unmarshalling
- `UnmarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Unmarshaller` is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#)

```java
final public Object unmarshal(java.io.File f) throws
```
JAXBException

unmarshal

public final Object unmarshal(File f) throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal XML data from the specified file and return the resulting content tree.

Implements Unmarshal Global Root Element.

Specified by:
unmarshal in interface Unmarshaller

Parameters:
f - the file to unmarshal XML data from

Returns:
the newly created root object of the java content tree

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshallerException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

final public Object unmarshal(java.io.InputStream is) throws JAXBException

unmarshal

public final Object unmarshal(InputStream is) throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal XML data from the specified InputStream and return the
resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API.

Implements Unmarshal Global Root Element.

**Specified by:**
unmarshal in interface Unmarshaller

**Parameters:**
is - the InputStream to unmarshal XML data from

**Returns:**
the newly created root object of the java content tree

**Throws:**
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

final public Object unmarshal(java.io.Reader reader)
throws JAXBException

unmarshal

public final Object unmarshal(Reader reader)
throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal XML data from the specified Reader and return the resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API, because a Reader does not provide the system ID.

Implements Unmarshal Global Root Element.

Specified by:
unmarshal in interface Unmarshaller

Parameters:
reader - the Reader to unmarshal XML data from

**Returns:**
the newly created root object of the java content tree

**Throws:**
*JAXBException* - If any unexpected errors occur while unmarshalling
*UnmarshallerException* - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Unmarshaller` is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#)

---

### public boolean isValidating() throws JAXBException

Unmarshaller

```java
JAXP isValidating() getValidating()
```

```java
return Unmarshaller true false
```

**Throws**
*JAXBException*:

---

### isValidating

public boolean `isValidating()`

```java
throws JAXBException
```

Indicates whether or not the Unmarshaller is configured to validate during unmarshal operations.

**Note:** I named this method `isValidating()` to stay in-line with JAXP, as opposed to naming it `getValidating()`.

**Specified by:**
*`isValidating`* in interface `Unmarshaller`

**Returns:**
true if the Unmarshaller is configured to validate during unmarshal operations, false otherwise

**Throws:**
*JAXBException* - if an error occurs while retrieving the validating
public void setEventHandler(ValidationEventHandler handler) throws JAXBException

Allow an application to register a validation event handler.

The validation event handler will be called by the JAXB Provider if any validation errors are encountered during calls to any of the unmarshal methods. If the client application does not register a validation event handler before invoking the unmarshal methods, then all validation events will be silently ignored and may result in unexpected behaviour.

Specified by:
setEventHandler in interface Unmarshaller

Parameters:
handler - the validation event handler

Throws:
JAXBException - if an error was encountered while setting the event handler

public void setValidating(boolean validating) throws JAXBException
Unmarshaller

**validating**  Unmarshaller  true false

**Throws**  JAXBException:

setValidating

```java
public void setValidating(boolean validating)
  throws JAXBException
```

Specifies whether or not the Unmarshaller should validate during unmarshal operations. By default, the Unmarshaller does not validate.

This method may only be invoked before or after calling one of the unmarshal methods.

**Specified by:**

setValidating in interface Unmarshaller

**Parameters:**

validating - true if the Unmarshaller should validate during unmarshal, false otherwise

**Throws:**

JAXBException - if an error occurred while enabling or disabling validation at unmarshal time

---

public **ValidationEventHandler** getEventHandler() throws JAXBException

```java
return ValidationEventHandler
  JAXBException:
```
getEventHandler

public ValidationEventHandler getEventHandler()
    throws JAXBException

    Return the current event handler or the default event handler if one
    hasn't been set.

    Specified by: getEventHandler in interface Unmarshaller

    Returns: the current ValidationEventHandler or the default event handler
    if it hasn't been set

    Throws: JAXBException - if an error was encountered while getting the
    current event handler

protected Unmarshalexception createUnmarshalException(org.xml.sax.SAXException e)
SAXException Unmarshalexception

    ContentHandler JAXBException SAXException unmarshaller JAXBException SAXException
    JAXBException SAXException JAXBException

    SAXException
    return Unmarshalexception

createUnmarshalException

protected Unmarshalexception createUnmarshalException(SAXException e)

    Creates an UnmarshalException from a SAXException. This is an
    utility method provided for the derived classes.
When a provider-implemented ContentHandler wants to throw a JAXBException, it needs to wrap the exception by a SAXException. If the unmarshaller implementation blindly wrap SAXException by JAXBException, such an exception will be a JAXBException wrapped by a SAXException wrapped by another JAXBException. This is silly.

This method checks the nested exception of SAXException and reduce those excessive wrapping.

**Returns:**
the resulting UnmarshalException

```java
public void setProperty(String name, Object value) throws PropertyException
```

**setProperty**

Default implementation of the setProperty method always throws PropertyException since there are no required properties. If a provider needs to handle additional properties, it should override this method in a derived class.

**Specified by:**
setProperty in interface Unmarshalled

**Parameters:**
- `name` - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
- `value` - the value of the property to be set

**Throws:**
- `PropertyException` - when there is an error processing the given
public Object getProperty(String name) throws PropertyException
getProperty  PropertyException

getProperty

public Object getProperty(String name) throws PropertyException

Default implementation of the getProperty method always throws PropertyException since there are no required properties. If a provider needs to handle additional properties, it should override this method in a derived class.

Specified by:
   getProperty in interface Unmarshaller
Parameters:
   name - the name of the property to retrieve
Returns:
   the value of the requested property
Throws:
   PropertyException - when there is an error retrieving the given property or value property name

public Object unmarshal(XMLEventReader reader) throws JAXBException

unmarshal

public Object unmarshal(XMLEventReader reader) throws JAXBException
Description copied from interface: Unmarshaller
Unmarshal XML data from the specified pull parser and return the resulting content tree.

This method is an Unmarshal Global Root method.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

Specified by:
unmarshal in interface Unmarshaller

Parameters:
reader - The parser to be read.

Returns:
the newly created root object of the java content tree.

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshallerException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

See Also:
Unmarshaller.unmarshal(javax.xml.stream.XMLStreamReader, Class)

public Object unmarshal(XMLStreamReader reader) throws JAXBException

unmarshal

public Object unmarshal(XMLStreamReader reader) throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal XML data from the specified pull parser and return the resulting content tree.

Implements Unmarshal Global Root Element.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

Specified by: unmarshal in interface Unmarshaller

Parameters:
reader - The parser to be read.

Returns:
the newly created root object of the java content tree.

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshallerException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

See Also:
Unmarshaller.unmarshal(javax.xml.stream.XMLStreamReader, Class)

unmarshal

public <T> JAXBElement<T> unmarshal(Node node, Class<T> expectedType)
throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.
Implements **Unmarshal by Declared Type**

**Specified by:**
*unmarshal* in interface *Unmarshaller*

**Parameters:**
- node - the document/element to unmarshal XML data from. The caller must support at least Document and Element.
- expectedType - appropriate JAXB mapped class to hold node's XML data.

**Returns:**
*JAXB Element* representation of node

**Throws:**
- *JAXBException* - If any unexpected errors occur while unmarshalling
- *UnmarshalException* - If the *ValidationEventHandler* returns false from its *handleEvent* method or the Unmarshaller is unable to perform the XML to Java binding. See **Unmarshalling XML Data**

```java
public <T> JAXBElement<T> unmarshal(Source source, Class<T> expectedType) throws JAXBException
```

**Description copied from interface: Unmarshaller**
Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.

**Specified by:** **Unmarshal by Declared Type**

See **SAX 2.0 Parser Pluggability**

**Parameters:**
- source - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and
StreamSource)

expectedType - appropriate JAXB mapped class to hold source's xml root element

Returns:
Java content rooted by JAXB Element

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshallerException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

unmarshal

public &lt;T&gt; JAXBElement&lt;T&gt; unmarshal(XMLStreamReader reader, Class&lt;T&gt; expectedType)
throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

This method implements unmarshal by declaredType.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

Specified by:
unmarshal in interface Unmarshaller

Parameters:
reader - The parser to be read.
expectedType - appropriate JAXB mapped class to hold reader's START_ELEMENT XML data.

Returns:
unmarshal

class <T>  JAXBElement< T> unmarshal(XMLEventReader reader, 
                                      Class< T> expectedType)
                                      throws JAXBException

Description copied from interface: Unmarshaller
Unmarshal root element to JAXB mapped declaredType and return 
the resulting content tree.

This method implements unmarshal by declaredType.

This method assumes that the parser is on a START_DOCUMENT 
or START_ELEMENT event. Unmarshalling will be done from this 
start event to the corresponding end event. If this method returns 
successfully, the reader will be pointing at the token right after the 
end event.

Specified by: 
unmarshal in interface Unmarshaller

Parameters:
reader - The parser to be read.
expectedType - appropriate JAXB mapped class to hold reader's 
START_ELEMENT XML data.

Returns:
content tree rooted by JAXB Element representation

Throws:
JAXBException - If any unexpected errors occur while 
unmarshalling
UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data

public void setSchema(javax.xml.validation.Schema schema)

setSchema

public void setSchema(Schema schema)

Description copied from interface: Unmarshaller
Specify the JAXP 1.3 Schema object that should be used to validate subsequent unmarshal operations against. Passing null into this method will disable validation.

This method replaces the deprecated setValidating(boolean) API.

Initially this property is set to null.

Specified by:
setSchema in interface Unmarshaller

Parameters:
schema - Schema object to validate unmarshal operations against or null to disable validation

public javax.xml.validation.Schema getSchema()

getSchema

public Schema getSchema()

Description copied from interface: Unmarshaller
Get the JAXP 1.3 Schema object being used to perform unmarshal-time validation. If there is no Schema set on the unmarshaller, then
this method will return null indicating that unmarshal-time validation will not be performed.

This method provides replacement functionality for the deprecated Unmarshaller.isValidating() API as well as access to the Schema object. To determine if the Unmarshaller has validation enabled, simply test the return type for null:

```java
boolean is_validating = u.getSchema() != null;
```

**Specified by:**
getSchema in interface Unmarshaller

**Returns:**
the Schema object being used to perform unmarshal-time validation or null if not present

---

```java
public void setAdapter("XmlAdapter adapter)
```

**setAdapter**

```java
public void setAdapter(XmlAdapter adapter)
```

**Description copied from interface:** Unmarshaller
Associates a configured instance of XmlAdapter with this unmarshaller.

This is a convenience method that invokes setAdapter(adapter.getClass(), adapter);

**Specified by:**
setAdapter in interface Unmarshaller

**See Also:**
Unmarshaller.setAdapter(Class, XmlAdapter)
**setAdapter**

```java
public <A extends XmlAdapter> void setAdapter(Class<A> type, A adapter)
```

**Description copied from interface: Unmarshaller**
Associates a configured instance of XmlAdapter with this unmarshaller.

Every unmarshaller internally maintains a `Map<Class, XmlAdapter>`, which it uses for unmarshalling classes whose fields/methods are annotated with `XmlJavaTypeAdapter`.

This method allows applications to use a configured instance of `XmlAdapter`. When an instance of an adapter is not given, an unmarshaller will create one by invoking its default constructor.

**Specified by:**
- `setAdapter` in interface `Unmarshaller`

**Parameters:**
- `type` - The type of the adapter. The specified instance will be used when `XmlJavaTypeAdapter.value()` refers to this type.
- `adapter` - The instance of the adapter to be used. If null, it will un-register the current adapter set for this type.

---

**getAdapter**

```java
public <A extends XmlAdapter> A getAdapter(Class<A> type)
```

**Description copied from interface: Unmarshaller**
Gets the adapter associated with the specified type. This is the reverse operation of the `Unmarshaller.setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter)` method.

**Specified by:**
- `getAdapter` in interface `Unmarshaller`
public void setAttachmentUnmarshaller(AttachmentUnmarshaller au)

setAttachmentUnmarshaller

public void setAttachmentUnmarshaller(AttachmentUnmarshaller au)

Description copied from interface: Unmarshaller

Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments.

Unmarshal time validation, enabled via Unmarshaller.setSchema(Schema), must be supported even when unmarshaller is performing XOP processing.

Specified by:
  setAttachmentUnmarshaller in interface Unmarshaller

public AttachmentUnmarshaller getAttachmentUnmarshaller()

getAttachmentUnmarshaller

public AttachmentUnmarshaller getAttachmentUnmarshaller()

  Specified by:
    getAttachmentUnmarshaller in interface Unmarshaller

public void setListener(Unmarshaller.Listener listener)

setListener
public void setListener(Unmarshaller.Listener listener)

Description copied from interface: Unmarshaller

Register unmarshal event callback Unmarshaller.Listener with this Unmarshaller.

There is only one Listener per Unmarshaller. Setting a Listener replaces the previous set Listener. One can unregister current Listener by setting listener to null.

Specified by:

setListener in interface Unmarshaller

Parameters:

listener - provides unmarshal event callbacks for this Unmarshaller

public Unmarshaller.Listener getListener()

getListener

public Unmarshaller.Listener getListener()

Description copied from interface: Unmarshaller

Return Unmarshaller.Listener registered with this Unmarshaller.

Specified by:

getListener in interface Unmarshaller

Returns:

registered Unmarshaller.Listener or null if no Listener is registered with this Unmarshaller.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  Class AccessLocalException

java.lang.Object  
  java.lang.Throwables  
    java.lang.Exception  
      java.lang.RuntimeException  
        javax.ejb.EJBException  
          javax.ejb.AccessLocalException

All Implemented Interfaces:
  Serializable

public class AccessLocalException
  extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

AccessLocalException

An AccessLocalException is thrown to indicate that the caller does not have permission to call the method. This exception is thrown to local clients.

See Also:
  Serialized Form

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessLocalException()</td>
</tr>
<tr>
<td>Constructs an AccessLocalException with no detail message.</td>
</tr>
<tr>
<td>AccessLocalException(String message)</td>
</tr>
<tr>
<td>Constructs an AccessLocalException with the specified detail message.</td>
</tr>
<tr>
<td>AccessLocalException(String message, Exception ex)</td>
</tr>
<tr>
<td>Constructs an AccessLocalException with the specified detail</td>
</tr>
</tbody>
</table>

message and a nested exception.

## Method Summary

### Methods inherited from class `javax.ejb.EJBException`
- `getCausedByException`, `getMessage`, `printStackTrace`, `printStackTrace`

### Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `setStackTrace`, `toString`

### Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

### public AccessLocalException()

**AccessLocalException**

### public AccessLocalException()

Constructs an AccessLocalException with no detail message.

### public AccessLocalException(String message)

**AccessLocalException**
public AccessLocalException(String message)

   Constructs an AccessLocalException with the specified detail message.

public AccessLocalException(String message, Exception ex)
AccessLocalException

AccessLocalException

public AccessLocalException(String message, Exception ex)

   Constructs an AccessLocalException with the specified detail message and a nested exception.
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
javax.faces.event Class ActionEvent

java.lang.Object
    ↓ java.util.EventObject
        ↓ javax.faces.event.FacesEvent
            ↓ javax.faces.event.ActionEvent

All Implemented Interfaces:
    Serializable

public class ActionEvent
    extends FacesEvent

Extends: java.util.EventObject > FacesEvent

UICommand   ActionEvent

An ActionEvent represents the activation of a user interface component (such as a UICommand).

See Also:
    Serialized Form

---

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class java.util.EventObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>ActionEvent(UIComponent component)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct a new event object from the specified source</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean isAppropriateListener(FacesListener listener)</code></td>
<td>Return true if this <code>FacesListener</code> is an instance of a listener class that this event supports.</td>
</tr>
<tr>
<td><code>void processListener(FacesListener listener)</code></td>
<td>Broadcast this <code>FacesEvent</code> to the specified <code>FacesListener</code>, by whatever mechanism is appropriate.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.event.FacesEvent`:
- `getComponent`, `getPhaseId`, `queue`, `setPhaseId`

Methods inherited from class `java.util.EventObject`:
- `getSource`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

```java
public ActionEvent(UIComponent component)
```

**Throws**
- `IllegalArgumentException`: `component` null

ActionEvent

```java
public ActionEvent(UIComponent component)
```
Construct a new event object from the specified source component and action command.

**Parameters:**
- **component** - Source `UIComponent` for this event

**Throws:**
- `IllegalArgumentException` - if component is null

### Method Detail

```java
public boolean isAppropriateListener(FacesListener listener)
```

**isAppropriateListener**

```java
public boolean isAppropriateListener(FacesListener listener)
```

**Description copied from class:** [FacesEvent](#)

Return true if this `FacesListener` is an instance of a listener class that this event supports. Typically, this will be accomplished by an "instanceof" check on the listener class.

**Specified by:**
- `isAppropriateListener` in class [FacesEvent](#)

**Parameters:**
- **listener** - `FacesListener` to evaluate

```java
public void processListener(FacesListener listener)
```

**Throws**
- `AbortProcessingException`: NotSupportedException  JavaServer Face

**processListener**
public void processListener(FacesListener listener)

Description copied from class: FacesEvent

Broadcast this FacesEvent to the specified FacesListener, by whatever mechanism is appropriate. Typically, this will be accomplished by calling an event processing method, and passing this FacesEvent as a parameter.

Specified by:  
processListener in class FacesEvent

Parameters:
- listener - FacesListener to send this FacesEvent to

Throws:
- AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.event Interface ActionListener

All Superinterfaces:
   EventListener, FacesListener

All Known Implementing Classes:
   MethodExpressionActionListener

public interface ActionListener
extends FacesListener

Implements: FacesListener
Implemented by: MethodExpressionActionListener

ActionEvent addActionListener()
UIComponent

A listener interface for receiving ActionEvents. A class that is interested in receiving such events implements this interface, and then registers itself with the source UIComponent of interest, by calling addActionListener().

Method Summary

<table>
<thead>
<tr>
<th>processAction</th>
<th>(ActionEvent event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Invoked when the action described by the specified(ActionEvent occurs.</td>
</tr>
</tbody>
</table>

Method Detail

public void processAction(ActionEvent event) throws AbortProcessingException
**ActionEvent**

```java
void processAction(ActionEvent event)
```

Invoked when the action described by the specified `ActionEvent` occurs.

**Parameters:**
- `event` - The `ActionEvent` that has occurred

**Throws:**
- `AbortProcessingException` - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

---

**Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.**

**PS:**
javax.faces.component Interface ActionSource

All Known Subinterfaces:  

ActionSource2

All Known Implementing Classes:  

HtmlCommandButton, HtmlCommandLink, UICommand

public interface ActionSource

Implemented by: ActionSource2

ActionSource is an interface that may be implemented by any concrete UIComponent that wishes to be a source of ActionEvent s, including the ability to invoke application actions via the default ActionListener mechanism.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addActionListener(ActionListener listener)</td>
<td>Add a new ActionListener to the set of listeners interested in being notified when ActionEvent s occur.</td>
</tr>
<tr>
<td>MethodBinding getAction()</td>
<td>Deprecated. This has been replaced by ActionSource2.getActionExpression().</td>
</tr>
<tr>
<td>MethodBinding getActionListener()</td>
<td>Deprecated. Use getActionListeners() instead.</td>
</tr>
<tr>
<td>ActionListener[] getActionListeners()</td>
<td>Return the set of registered ActionListeners for this</td>
</tr>
</tbody>
</table>
boolean isImmediate()  
Return a flag indicating that the default ActionListener provided by the JavaServer Faces implementation should be executed immediately (that is, during Apply Request Value phase of the request processing lifecycle), rather than wait until the Invoke Application phase.

void removeActionListener(ActionListener listener)  
Remove an existing ActionListener (if any) from the listeners interested in being notified when ActionEvents occur.

void setAction(MethodBinding action)  
**Deprecated. This has been replaced by ActionSource2.setActionExpression(javax.el.MethodExpression)**

void setActionListener(MethodBinding actionListener)  
**Deprecated. This has been replaced by addActionListener(javax.faces.event.ActionListener).**

void setImmediate(boolean immediate)  
Set the "immediate execution" flag for this UIComponent.

---

**Method Detail**

public MethodBinding getAction()

```
ActionSource2    ActionSource2#getActionExpression
#setAction       MethodBinding     MethodBinding
javax.el.MethodExpression

ActionSource2    MethodBinding    MethodBinding
UIComponent      immediate

deprecated       ActionSource2#getActionExpression
```

getAction
MethodBinding  getAction()

**Deprecated.** This has been replaced by 
*ActionSource2.getActionExpression()*.

If the implementing class also implements *ActionSource2*, the implementation of this method must call through to 
*ActionSource2.getActionExpression()* and examine the result. If the result came from a previous call to 
*setAction(javax.faces.el.MethodBinding)*, extract the MethodBinding from it and return it. Otherwise, wrap the returned *MethodExpression* in a MethodBinding implementation, and return it.

If the implementing class does not implement *ActionSource2*, return the MethodBinding pointing at the application action to be invoked, if this *UIComponent* is activated by the user, during the *Apply Request Values* or *Invoke Application* phase of the request processing lifecycle, depending on the value of the *immediate* property.

```
public void setAction(*MethodBinding* action)

    *ActionSource2*    *javax.el.MethodExpression*  action
    *ActionSource2#setActionExpression*  action

    *ActionSource2*    *MethodBinding*    *MethodBinding*
    *UIComponent*        immediate

    String

    action  MethodBinding
    deprecated  *setActionExpression(javax.el.MethodExpression)*
```

**setAction**

```
void setAction(*MethodBinding* action)
```
Deprecated. This has been replaced by `ActionSource2.setActionExpression(javax.el.MethodExpression)`.

If the implementing class also implements `ActionSource2`, the implementation of this method must wrap the argument `action` in a class that implements `MethodExpression` and call through to `ActionSource2.setActionExpression(javax.el.MethodExpression)`, passing the wrapped `action`.

If the implementing class does not implement `ActionSource2`, set the `MethodBinding` pointing at the application action to be invoked, if this `UIComponent` is activated by the user, during the `Apply Request Values` or `Invoke Application` phase of the request processing lifecycle, depending on the value of the `immediate` property.

Any method referenced by such an expression must be public, with a return type of `String`, and accept no parameters.

**Parameters:**

- `action` - The new `MethodBinding` expression

```java
public MethodBinding getActionListener()

#setActionListener null #setActionListener
#setActionListener MethodBinding

UIComponent immediate

deprecated #getActionListeners

g fåActionListener

MethodBinding getActionListener()

Deprecatd. Use `getActionListeners()` instead.

If `setActionListener(javax.faces.el.MethodBinding)` was not
previously called for this instance, this method must return `null`. If it was called, this method must return the exact `MethodBinding` instance that was passed to

```java
setActionListener(javax.faces.el.MethodBinding).
```

The method to be invoked, if this `UIComponent` is activated by the user, will be called during the `Apply Request Values` or `Invoke Application` phase of the request processing lifecycle, depending upon the value of the `immediate` property.

```java
public void setActionListener(MethodBinding actionListener)
```

- `ActionListener actionListener` #getActionListeners
- `setActionListener` #addActionListener(javax.faces.event.ActionListener)

```java
setActionListener
```

```java
void ActionEvent
```

- `actionListener` #addActionListener(javax.faces.event.ActionListener)
- `deprecated`

```java
setActionListener
```

```java
void setActionListener(MethodBinding actionListener)
```

**Deprecated.** This has been replaced by `addActionListener(javax.faces.event.ActionListener).

Wrap the argument `actionListener` in an implementation of `ActionListener` and store it in the internal data structure that backs the `getActionListeners()` method, taking care to over-write any instance that was stored by a previous call to `setActionListener`.

Any method referenced by such an expression must be public, with a return type of `void`, and accept a single parameter of type
Parameters:

- `actionListener` - The new method binding expression

---

**public boolean isImmediate()**

**JavaServer Faces**

- `ActionListener`

- `false`

**isImmediate**

**boolean isImmediate()**

Return a flag indicating that the default `ActionListener` provided by the JavaServer Faces implementation should be executed immediately (that is, during *Apply Request Values* phase of the request processing lifecycle), rather than waiting until the *Invoke Application* phase. The default value for this property must be `false`.

---

**public void setImmediate(boolean immediate)**

**UIComponent**

"immediate execution"

**boolean immediate**

**setImmediate**

**void setImmediate(boolean immediate)**

Set the "immediate execution" flag for this `UIComponent`.

**Parameters:**
The new immediate execution flag

**public void addActionListener(ActionListener listener)**

ActionEvent ActionListener

listener ActionListener

**Throws**

NullPointerException: listener null

**addActionListener**

void addActionListener(ActionListener listener)

Add a new ActionListener to the set of listeners interested in being notified when ActionEvents occur.

**Parameters:**

listener - The ActionListener to be added

**Throws:**

NullPointerException - if listener is null

**public ActionListener[] getActionListeners()**

ActionSource ActionListener 0

**getActionListeners**

ActionListener[] getActionListeners()

Return the set of registered ActionListeners for this ActionSource instance. If there are no registered listeners, a zero-length array is returned.
public void removeActionListener(ActionListener listener)

removeActionListener
void removeActionListener(ActionListener listener)

Remove an existing ActionListener (if any) from the set of listeners interested in being notified when ActionEvent s occur.

Parameters:
listener - The ActionListener to be removed

Throws:
NullPointerException - if listener is null

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface ActionSource2
extends ActionSource

Implements: ActionSource
Implemented by: UICommand

ActionSource2 extends ActionSource and provides a JavaBeans property analogous to the "action" property on ActionSource. The difference is the type of this property is a MethodExpression rather than a MethodBinding. This allows the ActionSource concept to leverage the new Unified EL API.

Since: 1.2

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getActionExpression()</td>
<td>Return the MethodExpression pointing at the</td>
</tr>
</tbody>
</table>
application action to be invoked, if this `UIComponent` is activated by the user, during the `Apply Request Values` or `Invoke Application` phase of the request processing lifecycle, depending on the value of the `immediate` property.

```java
void setActionExpression(MethodExpression action)
```

Set the `MethodExpression` pointing at the application action to be invoked, if this `UIComponent` is activated by the user, during the `Apply Request Values` or `Invoke Application` phase of the request processing lifecycle, depending on the value of the `immediate` property.

Methods inherited from interface `javax.faces.component.ActionSource`:
- `addActionListener`, `getAction`, `getActionListener`, `getActionListeners`, `isImmediate`, `removeActionListener`, `setAction`, `setActionListener`, `setImmediate`

**Method Detail**

```java
public MethodExpression getActionExpression()
```

```java
MethodExpression MethodExpression UIComponent
immediate
```

```java
MethodExpression ActionSource#setAction
MethodBinding ActionListener
```

**getActionExpression**

```java
MethodExpression getActionExpression()
```

Return the `MethodExpression` pointing at the application action to be
invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

Note that it's possible that the returned MethodExpression is just a wrapper around a MethodBinding instance wihch was set by a call to ActionSource.setAction(javax.faces.el.MethodBinding). This makes it possible for the default ActionListener to continue to work properly with older components.

---

```java
public void setActionExpression(MethodExpression action)
```

```
MethodExpression   MethodExpression   UIComponent
immediate

String

action
```

**setActionExpression**

```java
void setActionExpression(MethodExpression action)
```

Set the MethodExpression pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

Any method referenced by such an expression must be public, with a return type of String, and accept no parameters.

**Parameters:**

- **action** - The new method expression
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.enterprise.deploy.shared  **Class ActionType**

**java.lang.Object**

java.enterprise.deploy.shared.ActionType

---

public class **ActionType**

extends **Object**

---

**ActionType**  J2EE DeploymentStatus

Class ActionTypes defines enumeration values for the J2EE DeploymentStatus actions.

**Author:**

rsearls

---

### Field Summary

<table>
<thead>
<tr>
<th>Static ActionType</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CANCEL</strong></td>
<td></td>
<td>A cancel operation is being preformed on the DeploymentManager action command.</td>
</tr>
<tr>
<td><strong>EXECUTE</strong></td>
<td></td>
<td>The DeploymentManager action command is executing.</td>
</tr>
<tr>
<td><strong>STOP</strong></td>
<td></td>
<td>A stop operation is being preformed on the DeploymentManager action command.</td>
</tr>
</tbody>
</table>

---

### Constructor Summary

protected **ActionType**(int value)

Construct a new enumeration value with the given integer value.
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static ActionType getActionType(int value)</td>
<td>Return an object of the specified value.</td>
</tr>
<tr>
<td>protected ActionType[] getEnumValueTable()</td>
<td>Returns the enumeration value table for class ActionType</td>
</tr>
<tr>
<td>protected int getOffset()</td>
<td>Returns the lowest integer value used by this enumeration value's enumeration class.</td>
</tr>
<tr>
<td>protected String[] getStringTable()</td>
<td>Returns the string table for class ActionType</td>
</tr>
<tr>
<td>int getValue()</td>
<td>Returns this enumeration value's integer value.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Return the string name of this ActionType or the integer value if outside the bounds of the table</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang. Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

### Field Detail

**EXECUTE**

public static final ActionType EXECUTE

The DeploymentManager action command is executing.
CANCEL

public static final ActionType CANCEL

   A cancel operation is being preformed on the DeploymentManager action command.

STOP

public static final ActionType STOP

   A stop operation is being preformed on the DeploymentManager action command.

Constructor Detail

protected ActionType(int value)

   value

ActionType

protected ActionType(int value)

   Construct a new enumeration value with the given integer value.

   Parameters:
   value - Integer value.

Method Detail

public int getValue()
getValue

public int getValue()

    Returns this enumeration value's integer value.

    Returns: the value

protected String[] getStringTable()

getStringTable

protected String[] getStringTable()

    Returns the string table for class ActionType

protected ActionType[] getEnumValueTable()

getEnumValueTable

protected ActionType[] getEnumValueTable()

    Returns the enumeration value table for class ActionType

public static ActionType getActionType(int value)

value
getActionType

public static ActionType getActionType(int value)

    Return an object of the specified value.

    Parameters:
    value - a designator for the object.

public String toString()

ActionType

toString

public String toString()

    Return the string name of this ActionType or the integer value if outside the bounds of the table

    Overrides:
    toString in class Object

protected int getOffset()

    0
    return

g Moffet

protected int getOffset()
Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0.

**Returns:**
the offset of the lowest enumeration value.
javax.ejb Annotation Type ActivationConfigProperty

@Target(value={})
@Retention(value=RUNTIME)
public @interface ActivationConfigProperty

Implements: Annotation
@Target(value={})
@Retention(value=RUNTIME)

Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>propertyName</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>PropertyValue</td>
</tr>
</tbody>
</table>

Element Detail

abstract public String propertyName()

propertyName

public abstract String propertyName

abstract public String PropertyValue()

PropertyValue
public abstract String propertyValue

PS:
<table>
<thead>
<tr>
<th>Summary:</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail:</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>

---

PREV CLASS   NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTRUCTOR | METHOD
FRAMES  NO FRAMES
DETAIL: FIELD | CONSTRUCTOR | METHOD
javax.activation  **Class ActivationDataFlavor**

java.lang.Object  
   ▼ java.awt.datatransfer.DataFlavor  
      ▼ javax.activation.ActivationDataFlavor

**All Implemented Interfaces:**  
   Externalizable, Serializable, Cloneable

---

public class **ActivationDataFlavor**  
extends DataFlavor

**Extends:**  
java.awt.datatransfer.DataFlavor

ActivationDataFlavor  
java.awt.datatransfer.DataFlavor  
JAF  
DataFlavor  
equals MIME  
JDél

The ActivationDataFlavor class is a special subclass of  
java.awt.datatransfer.DataFlavor. It allows the JAF to set all three  
values stored by the DataFlavor class via a new constructor. It also  
contains improved MIME parsing in the equals method. Except for the  
improved parsing, its semantics are identical to that of the JDK's  
DataFlavor class.

**See Also:**  
Serialized Form

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class java.awt.datatransfer.DataFlavor</th>
</tr>
</thead>
<tbody>
<tr>
<td>imageFlavor, javaFileListFlavor, javaJVMLocalObjectMimeType,</td>
</tr>
<tr>
<td>javaRemoteObjectMimeType, javaSerializedObjectMimeType,</td>
</tr>
<tr>
<td>plainTextFlavor, stringFlavor</td>
</tr>
</tbody>
</table>
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ActivationDataFlavor(Class representationClass, String humanPresentableName)</code></td>
<td>Construct a DataFlavor that represents a MimeType.</td>
</tr>
<tr>
<td><code>ActivationDataFlavor(Class representationClass, String mimeType, String humanPresentableName)</code></td>
<td>Construct a DataFlavor that represents an arbitrary Java object.</td>
</tr>
<tr>
<td><code>ActivationDataFlavor(String mimeType, String humanPresentableName)</code></td>
<td>Construct a DataFlavor that represents a MimeType.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(DataFlavor dataFlavor)</code></td>
<td>Compares the DataFlavor passed in with this DataFlavor; calls the isMimeTypeEqual method.</td>
</tr>
<tr>
<td><code>getHumanPresentableName()</code></td>
<td>Return the Human Presentable name.</td>
</tr>
<tr>
<td><code>getMimeType()</code></td>
<td>Return the MIME type for this DataFlavor.</td>
</tr>
<tr>
<td><code>getRepresentationClass()</code></td>
<td>Return the representation class.</td>
</tr>
<tr>
<td><code>isMimeTypeEqual(String mimeType)</code></td>
<td>Is the string representation of the MIME type passed in equivalent to the MIME type of this DataFlavor.</td>
</tr>
<tr>
<td><code>normalizeMimeType(String mimeType)</code></td>
<td>Deprecated.</td>
</tr>
<tr>
<td><code>normalizeMimeTypeParameter(String parameterName, String parameterValue)</code></td>
<td>Deprecated.</td>
</tr>
<tr>
<td><code>setHumanPresentableName(String humanPresentableName)</code></td>
<td>Set the human presentable name.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.awt.datatransfer.DataFlavor

`clone`, `equals`, `equals`, `getDefaultRepresentationClass`, `getDefaultRepresentationClassAsString`, `getParameter`, `getPrimaryType`, `getReaderForText`, `getSubType`,...
public ActivationDataFlavor(Class<T> representationClass, String mimeType, String humanPresentableName)

Java DataFlavor JDK DataFlavor DataFlavor

representationClass = representationClass
mimeType = mimeType
humanName = humanName

ActivationDataFlavor

public ActivationDataFlavor(Class representationClass, String mimeType, String humanPresentableName)
Construct a DataFlavor that represents an arbitrary Java object. This constructor is an extension of the JDK’s DataFlavor in that it allows the explicit setting of all three DataFlavor attributes.

The returned DataFlavor will have the following characteristics:

representationClass = representationClass
mimeType = mimeType
humanName = humanName

**Parameters:**
- representationClass - the class used in this DataFlavor
- mimeType - the MIME type of the data represented by this class
- humanPresentableName - the human presentable name of the flavor

```java
public ActivationDataFlavor(Class<T> representationClass, String humanPresentableName)
```

DataFlavor

mimeType "application/x-java-serialized-object;
class=" DataFlavor(Class:forName())

representationClass = InputStream

mimeType = mimeType

<table>
<thead>
<tr>
<th>representationClass</th>
<th>DataFlavor</th>
</tr>
</thead>
<tbody>
<tr>
<td>humanPresentableName</td>
<td>flavor</td>
</tr>
</tbody>
</table>
ActivationDataFlavor

```
public ActivationDataFlavor(Class representationClass,
                             String humanPresentableName)
```

Construct a DataFlavor that represents a MimeType.

The returned DataFlavor will have the following characteristics:

If the mimeType is "application/x-java-serialized-object; class="", the result is the same as calling new DataFlavor(Class.forName()) as above.

otherwise:

representationClass = InputStream

mimeType = mimeType

**Parameters:**
- representationClass - the class used in this DataFlavor
- humanPresentableName - the human presentable name of the flavor

```
public ActivationDataFlavor(String mimeType, String humanPresentableName)
MimeType DataFlavor

DataFlavor

mimeType "application/x-java-serialized-object; class=" DataFlavor(Class.forName())

representationClass = InputStream

mimeType = mimeType
```
ActivationDataFlavor

public ActivationDataFlavor(String mimeType, String humanPresentableName)

Construct a DataFlavor that represents a MimeType.

The returned DataFlavor will have the following characteristics:

If the mimeType is "application/x-java-serialized-object; class=", the result is the same as calling new DataFlavor(Class.forName()) as above, otherwise:

representationClass = InputStream
mimeType = mimeType

Parameters:
  mimeType - the MIME type of the data represented by this class
  humanPresentableName - the human presentable name of the flavor

Method Detail

public String getMimeType()

  DataFlavor MIME
  return MIME

getMimeType

public String getMimeType()

  Return the MIME type for this DataFlavor.
Overrides:
getMimeType in class DataFlavor

Returns:
the MIME type

public Class<T> getRepresentationClass()
    return

getRepresentationClass

public Class getRepresentationClass()
    Return the representation class.

Overrides:
getRepresentationClass in class DataFlavor

Returns:
the representation class

public String getHumanPresentableName()
    return

getHumanPresentableName

public String getHumanPresentableName()
    Return the Human Presentable name.

Overrides:
getHumanPresentableName in class DataFlavor

Returns:
the human presentable name
public void setHumanPresentableName(String humanPresentableName)

    humanPresentableName

setHumanPresentableName

public void setHumanPresentableName(String humanPresentableName)

    Set the human presentable name.

    Overrides:
    setHumanPresentableName in class DataFlavor

    Parameters:
    humanPresentableName - the name to set

public boolean equals(java.awt.datatransfer.DataFlavor dataFlavor)

    DataFlavor DataFlavor

        isMimeTypeEqual

        dataFlavor DataFlavor

        return MIME true

equals

public boolean equals(DataFlavor dataFlavor)

    Compares the DataFlavor passed in with this DataFlavor; calls the
    isMimeTypeEqual method.

    Overrides:
    equals in class DataFlavor

    Parameters:
    dataFlavor - the DataFlavor to compare with
public boolean isMimeTypeEqual(String mimeType)

MIME  DataFlavor  MIME

ActivationDataFlavor  MIME  JavaBeans Activation Framework  MimeType  DataFlavor

mimeType

return

MIME  true

isMimeTypeEqual

public boolean isMimeTypeEqual(String mimeType)

Is the string representation of the MIME type passed in equivalent to the MIME type of this DataFlavor.

ActivationDataFlavor delegates the comparison of MIME types to the MimeType class included as part of the JavaBeans Activation Framework. This provides a more robust comparison than is normally available in the DataFlavor class.

Overrides:

isMimeTypeEqual in class DataFlavor

Parameters:

mimeType - the MIME type

Returns:

true if the same MIME type

protected String normalizeMimeTypeParameter(String parameterName, String parameterValue)

MIME  DataFlavor  DataFlavor / charset
normalizeMimeTypeParameter

protected String normalizeMimeTypeParameter(String mimeType)

Deprecated.

Called on DataFlavor for every MIME Type parameter to allow DataFlavor subclasses to handle special parameters like the text/plain charset parameters, whose values are case insensitive. (MIME type parameter values are supposed to be case sensitive).

This method is called for each parameter name/value pair and should return the normalized representation of the parameter value.

This method is never invoked by this implementation.

Overrides: normalizeMimeTypeParameter in class DataFlavor

Parameters:
  parameterName - the parameter name
  parameterValue - the parameter value

Returns:
  the normalized parameter value
normalizeMimeType

protected String normalizeMimeType(String mimeType)

Deprecated.

Called for each MIME type string to give DataFlavor subtypes the opportunity to change how the normalization of MIME types is accomplished. One possible use would be to add default parameter/value pairs in cases where none are present in the MIME type string passed in. This method is never invoked by this implementation.

Overrides:
    normalizeMimeType in class DataFlavor

Parameters:
    mimeType - the MIME type

Returns:
    the normalized MIME type
javax.resource.spi  Interface ActivationSpec

All Superinterfaces:  
   ResourceAdapterAssociation

public interface ActivationSpec  
extends ResourceAdapterAssociation

Implements: ResourceAdapterAssociation

ActivationSpec  JavaBean  
   version  1.0

This interface serves as a marker. An instance of an ActivationSpec must be a JavaBean and must be serializable. This holds the activation configuration information for a message endpoint.

Version:  
   1.0

Author:  
   Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| void validate()  
   This method may be called by a deployment tool to validate the overall activation configuration information provided by the endpoint deployer. |

Methods inherited from interface javax.resource.spi.ResourceAdapterAssociation  
getResourceAdapter, setResourceAdapter
public void validate() throws InvalidPropertyException

Throws

InvalidPropertyException:

validate

void validate()
throws InvalidPropertyException

This method may be called by a deployment tool to validate the overall activation configuration information provided by the endpoint deployer. This helps to catch activation configuration errors earlier on without having to wait until endpoint activation time for configuration validation. The implementation of this self-validation check behavior is optional.

Throws:

InvalidPropertyException - indicates invalid configuration property settings.

InvalidPropertyException

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.mail Class Address

java.lang.Object
   \- javax.mail.Address

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   InternetAddress, NewsAddress

public abstract class Address

extends Object
implements Serializable

Implements: java.io.Serializable
Extended by: InternetAddress, NewsAddress

Address

This abstract class models the addresses in a message. Subclasses provide specific implementations. Subclasses will typically be serializable so that (for example) the use of Address objects in search terms can be serialized along with the search terms.

Author:
   John Mani, Bill Shannon

See Also:
   Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address ()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract boolean equals(Object address)</td>
<td>The equality operator.</td>
</tr>
<tr>
<td>abstract String getType()</td>
<td>Return a type string that identifies this address type.</td>
</tr>
<tr>
<td>abstract String toString()</td>
<td>Return a String representation of this address object.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- clone
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

**public Address()**

**Address**

**public Address()**

### Method Detail

**abstract public String getType()**

```java
    return
```

See also  

```
javax.mail.internet.InternetAddress
```

**getType**

**public abstract String getType()**
Return a type string that identifies this address type.

**Returns:**
address type

**See Also:**
[InternetAddress](#)

---

```java
abstract public String toString()
Address String
    return
```

### toString

```java
generic method java lang Object
toString() String
```

**toString**

```
public abstract String toString()
```

Return a String representation of this address object.

**Overrides:**
`toString` in class [Object](#)

**Returns:**
String representation of this address

---

```java
abstract public boolean equals(Object address)
Address hashCode - hashCode
    address Address
```

### equals

```
public abstract boolean equals(Object address)
```

The equality operator. Subclasses should provide an implementation of this method that supports value equality (do the two Address objects represent the same destination?), not object reference...
equality. A subclass must also provide a corresponding implementation of the `hashCode` method that preserves the general contract of `equals` and `hashCode` - objects that compare as equal must have the same `hashCode`.

**Overrides:**

`equals` in class `Object`

**Parameters:**

`address` - Address object
public class AddressException

extends ParseException

Extends: Throwable > Exception > MessagingException > ParseException

The exception thrown when a wrongly formatted address is encountered.

Author: Bill Shannon, Max Spivak

See Also: Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected int pos</td>
<td>The index in the string where the error occurred, or -1 if not known.</td>
<td></td>
</tr>
<tr>
<td>protected String ref</td>
<td>The string being parsed.</td>
<td></td>
</tr>
</tbody>
</table>
**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressException()</td>
<td>Constructs an AddressException with no detail message.</td>
</tr>
<tr>
<td>AddressException(String s)</td>
<td>Constructs an AddressException with the specified detail message.</td>
</tr>
<tr>
<td>AddressException(String s, String ref)</td>
<td>Constructs an AddressException with the specified detail message and reference info.</td>
</tr>
<tr>
<td>AddressException(String s, String ref, int pos)</td>
<td>Constructs an AddressException with the specified detail message and reference info.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getPos()</td>
<td>Get the position with the reference string where the error was detected (-1 if not relevant).</td>
</tr>
<tr>
<td>String getRef()</td>
<td>Get the string that was being parsed when the error was detected (null if not relevant).</td>
</tr>
<tr>
<td>String toString()</td>
<td>Override toString method to provide information on nested exceptions.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.MessagingException
- getCause, getNextException, setNextException

Methods inherited from class java.lang.Throwable
- fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll,
Field Detail

ref

protected String ref

The string being parsed.

pos

protected int pos

The index in the string where the error occurred, or -1 if not known.

Constructor Detail

public AddressException()

AddressException

AddressException

public AddressException()

Constructs an AddressException with no detail message.

public AddressException(String s)

AddressException
AddressException

public AddressException(String s)

Constructs an AddressException with the specified detail message.

Parameters:
   s - the detail message

public AddressException(String s, String ref)

AddressException

public AddressException(String s, String ref, int pos)

AddressException

public AddressException(String s,
Constructs an AddressException with the specified detail message and reference info.

**Parameters:**
- s - the detail message

### Method Detail

#### public String getRef()

```java
String getRef()
null
```

**getRef**

```java
public String getRef()
```

Get the string that was being parsed when the error was detected (null if not relevant).

#### public int getPos()

```java
int getPos()
-1
```

**getPos**

```java
public int getPos()
```

Get the position with the reference string where the error was detected (-1 if not relevant).

#### public String toString()
public String toString()

Description copied from class: MessagingException
Override toString method to provide information on nested exceptions.

Overrides:
<code>toString</code> in class MessagingException
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS  FRAMES  NO FRAMES
SUMMARY: NESTED  FIELD  CONSTR  METHOD  DETAIL: FIELD  CONSTR  METHOD
javax.mail.search  Class AddressStringTerm

java.lang.Object
   javax.mail.search.SearchTerm
      javax.mail.search.StringTerm
         javax.mail.search.AddressStringTerm

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   FromStringTerm, RecipientStringTerm

public abstract class AddressStringTerm
   extends StringTerm

Extends: SearchTerm > StringTerm
Extended by: FromStringTerm, RecipientStringTerm

Message
   AddressTerm Address
   since JavaMail 1.1

This abstract class implements string comparisons for Message addresses.

Note that this class differs from the AddressTerm class in that this class
does comparisons on address strings rather than Address objects.

Since:
   JavaMail 1.1

See Also:
   Serialized Form

en
### Field Summary

Fields inherited from class `javax.mail.search.StringTerm`
- `ignoreCase`, `pattern`

### Constructor Summary

<table>
<thead>
<tr>
<th>protected</th>
<th><code>AddressStringTerm</code>&lt;br&gt;<code>(String pattern)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Constructor.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>equals(Object obj)</code></th>
<th>Equality comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected</td>
<td><code>match(Address a)</code></td>
<td>Check whether the address pattern specified in the constructor is a substring of the string representation of the given Address object.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.mail.search.StringTerm`
- `getIgnoreCase`, `getPattern`, `hashCode`, `match`

Methods inherited from class `javax.mail.search.SearchTerm`
- `match`

Methods inherited from class `java.lang.Object`
- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

### Constructor Detail

protected `AddressStringTerm(String pattern)`
protected AddressStringTerm(String pattern)

Constructor.

Parameters:
  pattern - the address pattern to be compared.

Method Detail

protected boolean match(Address a)

Check whether the address pattern specified in the constructor is a substring of the string representation of the given Address object.

Note that if the string representation of the given Address object contains charset or transfer encodings, the encodings must be accounted for, during the match process.

Parameters:
The comparison is applied to this Address object.

**Returns:**
true if the match succeeds, otherwise false.

**public boolean equals(Object obj)**

equals

public boolean equals(Object obj)

Equality comparison.

**Overrides:**
equals in class `StringTerm`
javax.mail.search Class AddressTerm

java.lang.Object
  javax.mail.search.SearchTerm
   javax.mail.search.AddressTerm

All Implemented Interfaces: Serializable

Direct Known Subclasses: FromTerm, RecipientTerm

public abstract class AddressTerm extends SearchTerm

Extends: SearchTerm
Extended by: FromTerm, RecipientTerm

Message Address

This class implements Message Address comparisons.

Author:
  Bill Shannon, John Mani

See Also:
  Serialized Form

---

Field Summary

<table>
<thead>
<tr>
<th>protected Address</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>The address.</td>
<td></td>
</tr>
</tbody>
</table>

---

Constructor Summary
protected AddressTerm(Address address)

# Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>Address</td>
<td>getAddress()</td>
<td>Return the address to match with.</td>
</tr>
<tr>
<td>int</td>
<td>hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>protected boolean</td>
<td>match(Address a)</td>
<td>Match against the argument Address.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.search.SearchTerm
- match

Methods inherited from class java.lang.Object
- clone, finalize, getClass, notify, notifyAll, toString, wait, wait

# Field Detail

address

protected Address address

The address.

# Constructor Detail

protected AddressTerm(Address address)
public Address getAddress()

getAddress

public Address getAddress()

Return the address to match with.

protected boolean match(Address a)

match

protected boolean match(Address a)

Match against the argument Address.

public boolean equals(Object obj)

equals
public boolean equals(Object obj)

Equality comparison.

Overrides:
equals in class Object

public int hashCode()
hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

Overrides:
hashCode in class Object

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.mail.search Class AndTerm

java.lang.Object
   ▼ javax.mail.search.SearchTerm
      ▼ javax.mail.search.AndTerm

All Implemented Interfaces:
   Serializable

public final class AndTerm
  extends SearchTerm

Extends: SearchTerm

SearchTermsThis AND

This class implements the logical AND operator on individual SearchTerms.

Author:
   Bill Shannon, John Mani
See Also:
   Serialized Form

---

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected terms</td>
<td>SearchTerm[]</td>
<td>The array of terms on which the AND operator should be applied.</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AndTerm(SearchTerm[] t)</td>
<td>Constructor that takes an array of SearchTerms.</td>
</tr>
</tbody>
</table>
AndTerm\((\text{SearchTerm} \ t1, \ \text{SearchTerm} \ t2)\)

Constructor that takes two terms.

### Method Summary

<table>
<thead>
<tr>
<th>boolean equals(\text{Object} \ obj)</th>
<th>Equality comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>searchTerm[] getTerms()</td>
<td>Return the search terms.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean match(Message msg)</td>
<td>The AND operation.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.\text{Object}

\text{clone, \ finalize, \ getClass, \ notify, \ notifyAll, \ toString, \ wait, \ wait}

### Field Detail

terms

protected \text{SearchTerm}[] \ terms

The array of terms on which the AND operator should be applied.

### Constructor Detail

public AndTerm(\text{SearchTerm} \ t1, \ \text{SearchTerm} \ t2)

\( t1 \)

\( t2 \)
AndTerm

class AndTerm/SearchTerm(t1, SearchTerm t2)

Constructor that takes two terms.

Parameters:
- t1 - first term
- t2 - second term

public AndTerm/SearchTerm[] t)

SearchTerm

Constructor that takes an array of SearchTerms.

Parameters:
- t - array of terms

Method Detail

public SearchTerm[] getTerms()
public boolean match(Message msg)

AND

AND

msg SearchTerm MessageAND

return AND true false

match

public boolean match(Message msg)

The AND operation.

The terms specified in the constructor are applied to the given object and the AND operator is applied to their results.

Specified by:

match in class SearchTerm

Parameters:

msg - The specified SearchTerms are applied to this Message and the AND operator is applied to their results.

Returns:

true if the AND succeeds, otherwise false

equals

public boolean equals(Object obj)
Equality comparison.

**Overrides:**

equals in class Object

```java
public int hashCode()
```

**hashCode**

```java
public int hashCode()
```

Compute a hashCode for this object.

**Overrides:**

hashCode in class Object

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.application Class Application

java.lang.Object
   \javax.faces.application.Application

public abstract class Application
extends Object

Application  Web  JavaServer Faces  JavaServer Faces

   ApplicationFactory  getApplication()  Application

Faces  Object
Application#createComponentApplication#createConverter
Application#createValidator

Application represents a per-web-application singleton object where applications based on JavaServer Faces (or implementations wishing to provide extended functionality) can register application-wide singletons that provide functionality required by JavaServer Faces. Default implementations of each object are provided for cases where the application does not choose to customize the behavior.

The instance of Application is created by calling the getApplication() method of ApplicationFactory. Because this instance is shared, it must be implemented in a thread-safe manner.

The application also acts as a factory for several types of Objects specified in the Faces Configuration file. Please see createComponent(java.lang.String), createConverter(java.lang.String), and createValidator(java.lang.String).
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract void addComponent(String componentType, String componentClass)</code></td>
<td>Register a new mapping of component type to the class.</td>
</tr>
<tr>
<td><code>abstract void addConverter(Class targetClass, String converterClass)</code></td>
<td>Register a new converter class that is capable of performing conversions for objects of the specified type.</td>
</tr>
<tr>
<td><code>abstract void addConverter(String converterId, String converterClass)</code></td>
<td>Register a new mapping of converter id to the name of the class.</td>
</tr>
<tr>
<td><code>void addELContextListener(ELContextListener listener)</code></td>
<td>Provide a way for Faces applications to register an ELContextListener.</td>
</tr>
<tr>
<td><code>void addELResolver(ELResolver resolver)</code></td>
<td>Cause an the argument resolver to be added to the <code>FacesContext ELResolvers</code> list.</td>
</tr>
<tr>
<td><code>abstract void addValidator(String validatorId, String validatorClass)</code></td>
<td>Register a new mapping of validator id to the name of the class.</td>
</tr>
<tr>
<td><code>abstract UIComponent createComponent(String componentType)</code></td>
<td>Instantiate and return a new UIComponent instance for the specified component type.</td>
</tr>
<tr>
<td><code>abstract UIComponent createComponent(ValueBinding componentBinding, FacesContext facesContext)</code></td>
<td>Call the <code>getValue()</code> method on the specified ValueBinding.</td>
</tr>
<tr>
<td><code>abstract UIComponent createComponent(ValueExpression componentExpression, String componentType)</code></td>
<td>Call the <code>getValue()</code> method on the specified ValueExpression.</td>
</tr>
<tr>
<td><code>abstract Converter createConverter(Class targetClass)</code></td>
<td>Instantiate and return a new Converter instance of the specified type.</td>
</tr>
<tr>
<td><code>abstract Converter createConverter(String converterId)</code></td>
<td>Instantiate and return a new Converter instance of the specified type.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>abstract <code>Converter</code></strong></td>
<td>Instantiate and return a new <code>Converter</code> instance of <code>addConverter()</code> for the specified converter id.</td>
</tr>
<tr>
<td><strong>abstract <code>MethodBinding</code></strong></td>
<td><code>createMethodBinding(String ref, Class[] params)</code> <strong>Deprecated. This has been replaced by calling</strong> <code>ExpressionFactory.createMethodExpression(javax.el.ELContext, java.lang.String, java.lang.Class, java.lang.Class[])</code>.</td>
</tr>
<tr>
<td><strong>abstract <code>Validator</code></strong></td>
<td><code>createValidator(String validatorId)</code> Instantiate and return a new <code>Validator</code> instance of <code>addValidator()</code> for the specified validator id.</td>
</tr>
<tr>
<td><strong>abstract <code>ValueBinding</code></strong></td>
<td><code>createValueBinding(String ref)</code> <strong>Deprecated. This has been replaced by calling</strong> <code>ExpressionFactory.createValueExpression(javax.el.ELContext, java.lang.String, java.lang.Class)</code>.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
<td><code>evaluateExpressionGet(FacesContext context, String expression)</code> Get a value by evaluating an expression.</td>
</tr>
<tr>
<td><strong>abstract <code>ActionListener</code></strong></td>
<td><code>getActionListener()</code> Return the default <code>ActionListener</code> to be registered for the application.</td>
</tr>
<tr>
<td><strong>abstract <code>Iterator&lt;String&gt;</code></strong></td>
<td><code>getComponentTypes()</code> Return an <code>Iterator</code> over the set of currently defined components.</td>
</tr>
<tr>
<td><strong>abstract <code>Iterator&lt;String&gt;</code></strong></td>
<td><code>getConverterIds()</code> Return an <code>Iterator</code> over the set of currently registered converter ids.</td>
</tr>
<tr>
<td><strong>abstract <code>Iterator&lt;Class&gt;</code></strong></td>
<td><code>getConverterTypes()</code> Return an <code>Iterator</code> over the set of class instances for which explicitly registered.</td>
</tr>
<tr>
<td><strong>abstract <code>Locale</code></strong></td>
<td><code>getDefaultLocale()</code> Return the default <code>Locale</code> for this application.</td>
</tr>
<tr>
<td><strong>abstract <code>String</code></strong></td>
<td><code>getDefaultRenderKitId()</code> Return the <code>renderKitId</code> to be used for rendering the application.</td>
</tr>
<tr>
<td><strong>ELContextListener[]</strong></td>
<td><code>getELContextListeners()</code> If no calls have been made to <code>addELContextListener</code> method, the method must return an empty array.</td>
</tr>
<tr>
<td><strong>ELResolver</strong></td>
<td><code>getELResolver()</code> Return the singleton <code>ELResolver</code> instance to be used by the application.</td>
</tr>
<tr>
<td><strong>ExpressionFactory</strong></td>
<td><code>getExpressionFactory()</code> Return the <code>ExpressionFactory</code> instance for this application.</td>
</tr>
</tbody>
</table>
abstract String getMessageBundle()  
Return the fully qualified class name of the ResourceBundle messages for this application.

abstract NavigationHandler getNavigationHandler()  
Return the NavigationHandler instance that will be invoked application action for this web application.

abstract PropertyResolver getPropertyResolver()  
Deprecated. This has been replaced by getELResolver()

ResourceBundle getResourceBundle(FacesContext ctx, String name)  
Find a ResourceBundle as defined in the application configuration resources under the specified name.

abstract StateManager getStateManager()  
Return the StateManager instance that will be utilized for Response phases of the request processing lifecycle.

abstract Iterator<Locale> getSupportedLocales()  
Return an Iterator over the supported Locales for this application.

abstract Iterator<String> getValidatorIds()  
Return an Iterator over the set of currently registered validators.

abstract VariableResolver getVariableResolver()  
Deprecated. This has been replaced by getELResolver()

abstract ViewHandler getViewHandler()  
Return the ViewHandler instance that will be utilized for Response phases of the request processing lifecycle.

void removeELContextListener(ELContextListener listener)  
Remove the argument listener from the list of ELContextListener.

abstract void setActionListener(ActionListener listener)  
Set the default ActionListener to be registered for all actions.

abstract void setDefaultLocale(Locale locale)  
Set the default Locale for this application.

abstract void setDefaultRenderKitId(String renderKitId)  
Set the renderKitId to be used to render this application.

abstract void setMessageBundle(String bundle)  
Set the fully qualified class name of the ResourceBundle messages for this application.

abstract void setNavigationHandler(NavigationHandler handler)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract void void Set the NavigationHandler instance that will be passed as the application action for this web application.</code></td>
<td></td>
</tr>
<tr>
<td><code>setPropertyResolver(PropertyResolver resolver)</code></td>
<td><strong>Deprecated.</strong> The recommended way to affect the <code>resolver&gt;</code> element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. The argument resolver to be wrapped inside an implementation of a resolution system as if the user had called <code>addELResolver</code></td>
</tr>
<tr>
<td><code>setStateManager(StateManager manager)</code></td>
<td>Set the StateManager instance that will be utilized during the <code>Response</code> phases of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>setSupportedLocales(Collection&lt;Locale&gt; locales)</code></td>
<td>Set the Locale instances representing the supported locales</td>
</tr>
<tr>
<td><code>setVariableResolver(VariableResolver resolver)</code></td>
<td><strong>Deprecated.</strong> The recommended way to affect the <code>resolver&gt;</code> element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. The argument resolver to be wrapped inside an implementation of a resolution system as if the user had called <code>addELResolver</code></td>
</tr>
<tr>
<td><code>setViewHandler(ViewHandler handler)</code></td>
<td>Set the ViewHandler instance that will be utilized during the <code>Response</code> phases of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Constructor Detail**

**public Application()**

**Application**

**public Application()**
abstract public **ActionListener** getActionListener()  

```java
javax.faces.component.ActionSource  ActionListener
```

- `processAction()`  `FacesContext.renderResponse()`
- `processAction()`
  - null `action`  `MethodBinding`  `invoke()`  `String`
  - null
- `processAction()`  `NavigationHandler`  `NavigationHandler#handleNavigation`
  - `FacesContext`
  - `action`  `MethodBinding`
    - `MethodBinding#getExpressionString`  `null`
  - `null`

```java
ActionListener ( action) ( MethodBinding)
ActionListener  javax.faces.component.ActionSource2
javax.faces.component.ActionSource
```

**getActionListener**

public abstract **ActionListener** getActionListener()  

Return the default **ActionListener** to be registered for all **ActionSource** components in this application. If not explicitly set, a default implementation must be provided that performs the following functions:

- The `processAction()` method must first call  
  `FacesContext.renderResponse()` in order to bypass any
The processAction() method must next determine the logical outcome of this event, as follows:

- If the originating component has a non-null action property, retrieve the MethodBinding from the property, and call invoke() on it. Convert the returned value (if any) to a String, and use it as the logical outcome.
- Otherwise, the logical outcome is null.

The processAction() method must finally retrieve the NavigationHandler instance for this application and call NavigationHandler.handleNavigation(javax.faces.context.FacesContext, java.lang.String, java.lang.String) passing:

- the FacesContext for the current request
- If there is a MethodBinding instance for the action property of this component, the result of calling MethodBinding.getExpressionString() on it, null otherwise
- the logical outcome as determined above

Note that the specification for the default ActionListener continues to call for the use of a deprecated property (action) and class (MethodBinding). Unfortunately, this is necessary because the default ActionListener must continue to work with components that do not implement ActionSource2, and only implement ActionSource.

---

abstract public void setActionListener(ActionListener listener)

javax.faces.component.ActionSource ActionListener

listener ActionListener

.Throws NullPointerException: listener null

setActionListener

public abstract void setActionListener(ActionListener listener)

Set the default ActionListener to be registered for all ActionSource
components.

**Parameters:**
- listener - The new default `ActionListener`

**Throws:**
- `NullPointerException` - if listener is null

```
abstract public java.util.Locale getDefaultLocale()
```

```
Locale null
```

### `getDefaultLocale`

```
public abstract Locale getDefaultLocale()
```

Return the default `Locale` for this application. If not explicitly set, `null` is returned.

```
abstract public void setDefaultLocale(java.util.Locale locale)
```

```
Locale
locale

Throws
Locale
NullPointerException: locale null
```

### `setDefaultLocale`

```
public abstract void setDefaultLocale(Locale locale)
```

Set the default `Locale` for this application.

**Parameters:**
locale - The new default Locale

Throws:

NullPointerException - if locale is null

abstract public String getDefaultRenderKitId()

renderKitId null

getDefaultRenderKitId

public abstract String getDefaultRenderKitId()

Return the renderKitId to be used for rendering this application. If not explicitly set, null is returned.

abstract public void setDefaultRenderKitId(String renderKitId)

renderKitId javax.faces.render.RenderKit ViewHandler Faces

setDefaultRenderKitId

public abstract void setDefaultRenderKitId(String renderKitId)

Set the renderKitId to be used to render this application. Unless the client has provided a custom ViewHandler that supports the use of multiple RenderKit instances in the same application, this method must only be called at application startup, before any Faces requests have been processed. This is a limitation of the current Specification, and may be lifted in a future release.
abstract public String getMessageBundle()

JavaServer Faces ResourceBundle null

getMessageBundle

public abstract String getMessageBundle()

Return the fully qualified class name of the ResourceBundle to be used for JavaServer Faces messages for this application. If not explicitly set, null is returned.

abstract public void setMessageBundle(String bundle)

JavaServer Faces ResourceBundle java.util.ResourceBundle JavaDocs

bundle

Throws NullPointerException: bundle null

setMessageBundle

public abstract void setMessageBundle(String bundle)

Set the fully qualified class name of the ResourceBundle to be used for JavaServer Faces messages for this application. See the JavaDocs for the java.util.ResourceBundle class for more information about the syntax for resource bundle names.

Parameters:
    bundle - Base name of the resource bundle to be used

Throws:
    NullPointerException - if bundle is null
abstract public NavigationHandler getNavigationHandler()

getNavigationHandler

public abstract NavigationHandler getNavigationHandler()

Return the NavigationHandler instance that will be passed the outcome returned by any invoked application action for this web application. If not explicitly set, a default implementation must be provided that performs the functions described in the NavigationHandler class description.

abstract public void setNavigationHandler(NavigationHandler handler)

setNavigationHandler

public abstract void setNavigationHandler(NavigationHandler handler)

Set the NavigationHandler instance that will be passed the outcome returned by any invoked application action for this web application.

Parameters:
	handler - The new NavigationHandler instance

Throws:

NullPointerException - if handler is null
abstract public PropertyResolver getPropertyResolver()

 ELResolver  PropertyResolver  Faces  ELResolver
 EL

 PropertyResolver  PropertyResolver  EL  (no-op)

 deprecated

getPropertyResolver

public abstract PropertyResolver getPropertyResolver()

 Deprecated. This has been replaced by getELResolver().

Return a PropertyResolver instance that wraps the ELResolver instance that Faces provides to the unified EL for the resolution of expressions that appear programmatically in an application.

Note that this no longer returns the default PropertyResolver since that class is now a no-op that aids in allowing custom PropertyResolverS to affect the EL resolution process.

abstract public void setPropertyResolver(PropertyResolver resolver)

PropertyResolver

 resolver

Throws

NullPointerException: resolver null
setPropertyResolver

public abstract void setPropertyResolver(PropertyResolver resolver)

**Deprecated.** The recommended way to affect the execution of the EL is to provide an `<el-resolver>` element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. This method now will cause the argument resolver to be wrapped inside an implementation of `ELResolver` and exposed to the EL resolution system as if the user had called `addELResolver(javax.el.ELResolver)`.

Set the `PropertyResolver` instance that will be utilized to resolve method and value bindings.

This method is now deprecated but the implementation must cause the argument to be set as the head of the legacy `PropertyResolver` chain, replacing any existing value that was set from the application configuration resources.

It is illegal to call this method after the application has received any requests from the client. If an attempt is made to register a listener after that time it must have no effect.

**Parameters:**
- resolver - The new `PropertyResolver` instance

**Throws:**
- `NullPointerException` - if resolver is null
- `IllegalStateException` - if called after the first request to the `FacesServlet` has been serviced.
getResourceBundle(FacesContext ctx, String name)

ResourceBundle ResourceBundle
javax.faces.component.UIViewRoot

UnsupportedOperationException

return UIViewRoot ResourceBundle null
Throws FacesException:
Throws NullPointerException: ctx == null || name == null
since 1.2

getResourceBundle

public ResourceBundle getResourceBundle(FacesContext ctx,
                                        String name)

Find a ResourceBundle as defined in the application configuration
resources under the specified name. If a ResourceBundle was defined
for the name, return an instance that uses the locale of the current
UIViewRoot.

The default implementation throws UnsupportedOperationException
and is provided for the sole purpose of not breaking existing
applications that extend this class.

Returns:
    ResourceBundle for the current UIViewRoot, otherwise null
Throws:
    FacesException - if a bundle was defined, but not resolvable
    NullPointerException - if ctx == null || name == null
Since:
    1.2

abstract public VariableResolver getVariableResolver()
getVariableResolver

public abstract VariableResolver getVariableResolver()

Deprecated. This has been replaced by getELResolver().

Return the VariableResolver that wraps the ELResolver instance that Faces provides to the unified EL for the resolution of expressions that appear programmatically in an application. The implementation of the VariableResolver must pass null as the base argument for any methods invoked on the underlying ELResolver.

Note that this method no longer returns the default VariableResolver, since that class now is a no-op that aids in allowing custom VariableResolvers to affect the EL resolution process.

abstract public void setVariableResolver(VariableResolver resolver)

Throws NullPointerException: resolver null

<el-resolver>
deprecated

```java
public abstract void setVariableResolver(VariableResolver resolver)
```

**Deprecated.** The recommended way to affect the execution of the EL is to provide an `<el-resolver>` element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. This method now will cause the argument `resolver` to be wrapped inside an implementation of `ELResolver` and exposed to the EL resolution system as if the user had called `addELResolver(javax.el.ELResolver)`.

Set the `VariableResolver` instance that will be consulted to resolve method and value bindings.

This method is now deprecated but the implementation must cause the argument to be set as the head of the legacy `VariableResolver` chain, replacing any existing value that was set from the application configuration resources.

It is illegal to call this method after the application has received any requests from the client. If an attempt is made to register a listener after that time it must have no effect.

**Parameters:**
- `resolver` - The new `VariableResolver` instance

**Throws:**
- `NullPointerException` - if `resolver` is `null`
- `IllegalStateException` - if called after the first request to the `FacesServlet` has been serviced.

```java
public void addELResolver(ELResolver resolver)
```
addELResolver

public void addELResolver(ELResolver resolver)

Cause an the argument resolver to be added to the resolver chain as specified in section 5.5.1 of the JavaServer Faces Specification.

It is not possible to remove an ELResolver registered with this method, once it has been registered.

It is illegal to register an ELResolver after the application has received any requests from the client. If an attempt is made to register a listener after that time, an IllegalStateException must be thrown. This restriction is in place to allow the JSP container to optimize for the common case where no additional ELResolverS are in the chain, aside from the standard ones. It is permissible to add ELResolverS before or after initialization to a CompositeELResolver that is already in the chain.

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend Application.

Since:

1.2
public ELResolver getELResolver()

EL ELResolver javax.el.CompositeELResolver
ELResolver

1. <el-resolver> ELResolver

2. 5 VariableResolver ChainWrapper
VariableResolver implementation

3. 5 PropertyResolver ChainWrapper
PropertyResolver implementation

4. #addELResolver ELResolver

UnsupportedOperationException Application

since 1.2

getELResolver

public ELResolver getELResolver()

Return the singleton ELResolver instance to be used for all EL resolution. This is actually an instance of CompositeELResolver that must contain the following ELResolver instances in the following order:

1. ELResolver instances declared using the <el-resolver> element in the application configuration resources.

2. An implementation that wraps the head of the legacy VariableResolver chain, as per section VariableResolver ChainWrapper in Chapter 5 in the spec document.
3. An implementation that wraps the head of the legacy PropertyResolver chain, as per section PropertyResolver ChainWrapper in Chapter 5 in the spec document.

4. Any ELResolver instances added by calls to addELResolver(javax.el.ELResolver).

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend Application.

Since:
1.2

abstract public ViewHandler getViewHandler()

getViewHandler

public abstract ViewHandler getViewHandler()

Return the ViewHandler instance that will be utilized during the Restore View and Render Response phases of the request processing lifecycle. If not explicitly set, a default implementation must be provided that performs the functions described in the ViewHandler description in the JavaServer Faces Specification.

abstract public void setViewHandler(ViewHandler handler)

setDefaultHandler
**setViewHandler**

```java
public abstract void setViewHandler(ViewHandler handler)
```

Set the `ViewHandler` instance that will be utilized during the `Restore View` and `Render Response` phases of the request processing lifecycle.

**Parameters:**
- `handler` - The new `ViewHandler` instance

**Throws:**
- `IllegalStateException` - if this method is called after at least one request has been processed by the `Lifecycle` instance for this application.
- `NullPointerException` - if `handler` is null

---

**abstract public StateManager getStateManager()**

```java
StateManager JavaServer Faces
```

**getStateManager**

```java
public abstract StateManager getStateManager()
```

Return the `StateManager` instance that will be utilized during the `Restore View` and `Render Response` phases of the request processing lifecycle. If not explicitly set, a default implementation must be provided that performs the functions described in the `StateManager` description in the JavaServer Faces Specification.
abstract public void setStateManager(StateManager manager)

StateManager

manager
Throws  IllegalArgumentException: Lifecycle
Throws  NullPointerException: manager null

setStateManager

public abstract void setStateManager(StateManager manager)

Set the StateManager instance that will be utilized during the Restore View and Render Response phases of the request processing lifecycle.

Parameters:
manager - The new StateManager instance

Throws:
IllegalArgumentException - if this method is called after at least one request has been processed by the Lifecycle instance for this application.
NullPointerException - if manager is null

abstract public void addComponent(String componentType, String componentClass)

UIComponent

createComponent()

UIComponent

componentType
componentClass
Throws
NullPointerException: componentType
NullPointerException: componentClass null
addComponent

public abstract void addComponent(String componentType, String componentClass)

Register a new mapping of component type to the name of the corresponding UIComponent class. This allows subsequent calls to createComponent() to serve as a factory for UIComponent instances.

Parameters:
componentType - The component type to be registered
componentClass - The fully qualified class name of the corresponding UIComponent implementation

Throws:
NullPointerException - if componentType or componentClass is null

abstract public UIComponent createComponent(String componentType) throws FacesException

UIComponent addComponent()

Throws
FacesException: UIComponent

Throws
NullPointerException: componentType null

createComponent

public abstract UIComponent createComponent(String componentType) throws FacesException

Instantiate and return a new UIComponent instance of the class specified by a previous call to addComponent() for the specified component type.

Parameters:
componentType - The component type for which to create and
return a new `UIComponent` instance

**Throws:**
- `FacesException` - if a `UIComponent` of the specified type cannot be created
- `NullPointerException` - if `componentType` is null

```java
abstract public UIComponent createComponent(ValueBinding componentBinding,
                                              FacesContext context,
                                              String componentType)
throws FacesException
```

```java
ValueExpression componentBinding
#createComponent(javax.el.ValueExpression, javax.faces.context.FacesContext, java.lang.String)
```

```java
componentBinding
context
componentType
Throws
Throws
FacesException: UIComponent
NullPointerException:
```

deprecated `#createComponent(javax.el.ValueExpression, javax.faces.context.FacesContext, java.lang.String)`

---

**createComponent**

```java
public abstract UIComponent createComponent(ValueBinding componentBinding,
                                             FacesContext context,
                                             String componentType)
throws FacesException
```

**Deprecated. This has been replaced by**

```java
createComponent(javax.el.ValueExpression, javax.faces.context.FacesContext, java.lang.String)
```

Wrap the argument `componentBinding` in an implementation of `ValueExpression` and call through to

```java
createComponent(javax.el.ValueExpression, javax.faces.context.FacesContext, java.lang.String)
```
Parameters:

- `componentBinding` - ValueBinding representing a component value binding expression (typically specified by the component attribute of a custom tag)
- `context` - FacesContext for the current request
- `componentType` - Component type to create if the ValueBinding does not return a component instance

Throws:

- FacesException - if a UIComponent cannot be created
- NullPointerException - if any parameter is null

```java
public UIComponent createComponent(ValueExpression componentExpression, FacesContext context, String componentType) throws FacesException
```

<table>
<thead>
<tr>
<th>ValueExpression</th>
<th>getValue()</th>
<th>UIComponent</th>
<th>setValue()</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIComponent</td>
<td>UIComponent</td>
<td>ValueExpression</td>
<td></td>
</tr>
</tbody>
</table>

Throws

- FacesException: component
- UnsupportedOperationException: Application
- NullPointerException: null

since 1.2

createComponent

```java
public UIComponent createComponent(ValueExpression componentExpression, FacesContext context, String componentType)
```
Call the `getValue()` method on the specified `ValueExpression`. If it returns a `UIComponent` instance, return it as the value of this method. If it does not, instantiate a new `UIComponent` instance of the specified component type, pass the new component to the `setValue()` method of the specified `ValueExpression`, and return it.

**Parameters:**
- `componentExpression` - `ValueExpression` representing a component value expression (typically specified by the component attribute of a custom tag)
- `context` - `FacesContext` for the current request
- `componentType` - Component type to create if the `ValueExpression` does not return a component instance

**Throws:**
- `FacesException` - if a `UIComponent` cannot be created
- `NullPointerException` - if any parameter is null

A default implementation is provided that throws `UnsupportedOperationException` so that users that decorate `Application` can continue to function.

Since: 1.2

```java
abstract public java.util.Iterator<E> getComponentTypes()

Application  Iterator

gGetComponentTypes

public abstract Iterator<String> getComponentTypes()

Return an `Iterator` over the set of currently defined component
types for this Application.

abstract public void addConverter(String converterId, String converterClass)

ID Converter createConverter() Converter

converterId converterClass
converterClass

Throws NullPointerException: converterId converterClass null

addConverter

public abstract void addConverter(String converterId, String converterClass)

Register a new mapping of converter id to the name of the corresponding Converter class. This allows subsequent calls to createConverter() to serve as a factory for Converter instances.

Parameters:
converterId - The converter id to be registered
converterClass - The fully qualified class name of the corresponding Converter implementation

Throws:
NullPointerException - if converterId or converterClass is null

abstract public void addConverter(Class<T> targetClass, String converterClass)

targetClass
addConverter

public abstract void addConverter(Class targetClass, String converterClass)

Register a new converter class that is capable of performing conversions for the specified target class.

Parameters:
  targetClass - The class for which this converter is registered
  converterClass - The fully qualified class name of the corresponding Converter implementation

Throws:
  NullPointerException - if targetClass or converterClass is null

abstract public Converter createConverter(String converterId)

   Converter ID addConverter() ID
null

createConverter

public abstract Converter createConverter(String converterId)

Instantiate and return a new Converter instance of the class specified by a previous call to addConverter() for the specified
converter id. If there is no such registration for this converter id, return null.

**Parameters:**
- `converterId` - The converter id for which to create and return a new `Converter` instance

**Throws:**
- `FacesException` - if the `Converter` cannot be created
- `NullPointerException` - if `converterId` is null

```java
abstract public Converter createConverter(Class<T> targetClass)
```

- `Converter`
- `Converter`
- `Converter`

Instantiate and return a new `Converter` instance of the class that has registered itself as capable of performing conversions for objects of the specified type. If no such `Converter` class can be identified,
return null.

To locate an appropriate Converter class, the following algorithm is performed, stopping as soon as an appropriate Converter class is found:

- Locate a Converter registered for the target class itself.
- Locate a Converter registered for interfaces that are implemented by the target class (directly or indirectly).
- Locate a Converter registered for the superclass (if any) of the target class, recursively working up the inheritance hierarchy.

If the Converter has a single argument constructor that accepts a Class, instantiate the Converter using that constructor, passing the argument targetClass as the sole argument. Otherwise, simply use the zero-argument constructor.

**Parameters:**

- targetClass - Target class for which to return a Converter

**Throws:**

- FacesException - if the Converter cannot be created
- NullPointerException - if targetClass is null

---

abstract public java.util.Iterator<E> getConverterIds()

**getConverterIds**

public abstract Iterator<String> getConverterIds()

Return an Iterator over the set of currently registered converter ids for this Application.

---

abstract public java.util.Iterator<E> getConverterTypes()
getConverterTypes

public abstract Iterator<Class> getConverterTypes()

Return an Iterator over the set of Class instances for which Converter classes have been explicitly registered.

public ExpressionFactory getExpressionFactory()

ExpressionFactory #evaluateExpressionGet

JspFactory.getDefaultFactory().getJspApplicationContext(servletContext).getExpressionFactory()

UnsupportedOperationException Application since 1.2

getExpressionFactory

public ExpressionFactory getExpressionFactory()

Return the ExpressionFactory instance for this application. This instance is used by the convenience method 


The implementation must return the ExpressionFactory from the JSP container by calling

JspFactory.getDefaultFactory().getJspApplicationContext(servletContext).getExpressionFactory()
An implementation is provided that throws UnsupportedOperationException so that users that decorate the Application continue to work.

Since:
1.2

```java
public Object evaluateExpressionGet(FacesContext context, String expression, Class<T> expectedType) throws ELException
```

```java
getExpressionFactory
ExpressionFactory#createValueExpression expression
expectedType FacesContext#getELContext
ValueExpression#getValue

UnsupportedOperationException Application
```

evaluateExpressionGet

```java
public Object evaluateExpressionGet(FacesContext context, String expression, Class<T> expectedType) throws ELException
```

Get a value by evaluating an expression.

Call `getExpressionFactory()` then call `ExpressionFactory#createValueExpression(javax.el.ELContext, java.lang.String, java.lang.Class)` passing the argument `expression` and `expectedType`. Call `FacesContext#getELContext()` and pass it to `ValueExpression#getValue(javax.el.ELContext)`, returning the result.
An implementation is provided that throws 
UnsupportedOperationException so that users that decorate the 
Application continue to work.

**Throws:**
- `ELException`

abstract public `MethodBinding` createMethodBinding(String ref, Class<T>[] params)
throws `ReferenceSyntaxException`

```java
#getExpressionFactory
ExpressionFactory#createMethodExpression
```

```java
ref
params 0
Throws NullPointerException: ref null
Throws ReferenceSyntaxException: ref
```  

deprecated
```java
ExpressionFactory#createMethodExpression
```

createMethodBinding

public abstract `MethodBinding` createMethodBinding(String ref, Class[] params)
throws `ReferenceSyntaxException`

**Deprecated. This has been replaced by calling**
```java
getExpressionFactory() then
ExpressionFactory#createMethodExpression(javax.el.ELContext,
java.lang.String, java.lang.Class, java.lang.Class[]).
```

Call `getExpressionFactory()` then call
```java
ExpressionFactory#createMethodExpression(javax.el.ELContext,
java.lang.String, java.lang.Class, java.lang.Class[]),
```  
passing the given arguments, and wrap the result in a `MethodBinding`
implementation, returning it.

**Parameters:**
- `ref` - Method binding expression for which to return a `MethodBinding` instance
- `params` - Parameter signatures that must be compatible with those of the method to be invoked, or a zero-length array or `null` for a method that takes no parameters

**Throws:**
- `NullPointerException` - if `ref` is `null`
- `ReferenceSyntaxException` - if the specified `ref` has invalid syntax

```
abstract public java.util.Iterator<E> getSupportedLocales()

Locale    Iterator
```

**getSupportedLocales**

```
public abstract Iterator<Locale> getSupportedLocales()

Return an Iterator over the supported `Locale` instances for this application.
```

```
setSupportedLocales

public abstract void setSupportedLocales(Collection<Locale> locales)

Set the `Locale` instances representing the supported `Locale` instances for this application.

**Parameters:**
- `locales` - The set of supported `Locale` instances for this application

**Throws:**
NullPointerException - if the argument newLocales is null.

**public void addELContextListener(ELContextListener listener)**

<table>
<thead>
<tr>
<th>Faces</th>
<th>ELContextListener</th>
<th>ELContext</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UnsupportedOperationException</td>
<td>Application</td>
</tr>
<tr>
<td></td>
<td>since</td>
<td>1.2</td>
</tr>
</tbody>
</table>

**addELContextListener**

public void addELContextListener(ELContextListener listener)

Provide a way for Faces applications to register an ELContextListener that will be notified on creation of ELContext instances. This listener will be called once per request.

An implementation is provided that throws UnsupportedOperationException so that users that decorate the Application continue to work.

**Since:**

1.2

**public void removeELContextListener(ELContextListener listener)**

ELContextListener listener listener null

UnsupportedOperationException Application

since 1.2
removeELContextListener

public void removeELContextListener(ELContextListener listener)

Remove the argument listener from the list of ELContextListenerS. If listener is null, no exception is thrown and no action is performed. If listener is not in the list, no exception is thrown and no action is performed.

An implementation is provided that throws
UnsupportedOperationException so that users that decorate the Application continue to work.

Since:
1.2

getELContextListeners

public ELContextListener[] getELContextListeners()

If no calls have been made to addELContextListener(javax.el.ELContextListener), this method must return an empty array.
Otherwise, return an array representing the list of listeners added by calls to `addELContextListener(javax.el.ELContextListener)`.

An implementation is provided that throws `UnsupportedOperationException` so that users that decorate the Application continue to work.

**Since:**
1.2

---

**abstract public void addValidator(String validatorId, String validatorClass)**

<table>
<thead>
<tr>
<th>ID</th>
<th><strong>Validator</strong></th>
<th>createValidator()</th>
<th><strong>Validator</strong></th>
</tr>
</thead>
</table>

`validatorId`  
`validatorClass`  

**Throws**  
`NullPointerException: validatorId or validatorClass is null`

**addValidator**

```
public abstract void addValidator(String validatorId, String validatorClass)
```

Register a new mapping of validator id to the name of the corresponding **Validator** class. This allows subsequent calls to `createValidator()` to serve as a factory for **Validator** instances.

**Parameters:**
- `validatorId` - The validator id to be registered
- `validatorClass` - The fully qualified class name of the corresponding **Validator** implementation

**Throws:**
- `NullPointerException` - If `validatorId` or `validatorClass` is null
abstract public **Validator** createValidator(String validatorId) throws **FacesException**

<table>
<thead>
<tr>
<th><strong>Validator</strong></th>
<th>ID</th>
<th>addValidator()</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>validatorId</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Throws</td>
<td><strong>FacesException</strong>: ID</td>
<td></td>
</tr>
<tr>
<td>Throws</td>
<td><strong>NullPointerException</strong>: validatorId null</td>
<td></td>
</tr>
</tbody>
</table>

**createValidator**

public abstract **Validator** createValidator(String validatorId) throws **FacesException**

Instantiate and return a new **validator** instance of the class specified by a previous call to addValidator() for the specified validator id.

**Parameters:**
validatorId - The validator id for which to create and return a new **Validator** instance

**Throws:**
**FacesException** - if a **Validator** of the specified id cannot be created
**NullPointerException** - if validatorId is null

**abstract public java.util.Iterator<E> getValidatorIds()**

<table>
<thead>
<tr>
<th>Application ID</th>
<th>Iterator</th>
</tr>
</thead>
</table>

**getValidatorIds**

public abstract **Iterator<String>** getValidatorIds()
Return an Iterator over the set of currently registered validator ids for this Application.

abstract public ValueBinding createValueBinding(String ref) throws ReferenceSyntaxException

#getExpressionFactory
ExpressionFactory#createValueExpression ref
expectedType Object.class fnMapper null

createValueBinding

public abstract ValueBinding createValueBinding(String ref) throws ReferenceSyntaxException

Deprecated. This has been replaced by calling
getExpressionFactory() then
ExpressionFactory#createValueExpression(javax.el.ELContext,
java.lang.String, java.lang.Class).

Call getExpressionFactory() then call
ExpressionFactory#createValueExpression(javax.el.ELContext,
java.lang.String, java.lang.Class), passing the argument ref, Object.class for the expectedType, and null, for the fnMapper.

Parameters:
ref - Value binding expression for which to return a ValueBinding instance

Throws:
NullPointerException - if ref is null
ReferenceSyntaxException - if the specified ref has invalid syntax

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  

**Annotation Type ApplicationException**

@Target(value=TYPE)  
@Retention(value=RUNTIME)  

public interface ApplicationException

**Implements**: Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

Applied to an exception to denote that it is an application exception and should be reported to the client directly (i.e., unwrapped).

### Optional Element Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th>rollback</th>
</tr>
</thead>
</table>

Indicates whether the container should cause the transaction to rollback when the exception is thrown.

**abstract public boolean rollback()**

**rollback**

public abstract boolean rollback

Indicates whether the container should cause the transaction to rollback when the exception is thrown.

**Default:**

false
javax.faces.application Class ApplicationFactory

java.lang.Object
   \ javax.faces.application.ApplicationFactory

public abstract class ApplicationFactory
    extends Object

ApplicationFactory Application JavaServer Faces

Application ApplicationFactory

JavaServer Faces  Web ApplicationFactory

ApplicationFactory factory = (ApplicationFactory)
FactoryFinder.getFactory(FactoryFinder.APPLICATION_FACTORY);

ApplicationFactory is a factory object that creates (if needed) and returns Application instances. Implementations of JavaServer Faces must provide at least a default implementation of Application.

There must be one ApplicationFactory instance per web application that is utilizing JavaServer Faces. This instance can be acquired, in a portable manner, by calling:

    ApplicationFactory factory = (ApplicationFactory)
        FactoryFinder.getFactory(FactoryFinder.APPLICATION_FACTORY);

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationFactory()</td>
<td></td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract Application getApplication()</td>
<td>Create (if needed) and return an Application instance for this web application.</td>
</tr>
<tr>
<td>abstract void setApplication(Application application)</td>
<td>Replace the Application instance that will be returned for this web application.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ApplicationFactory()

ApplicationFactory

public ApplicationFactory()

Method Detail

abstract public Application getApplication()

Web Application

getApplication

public abstract Application getApplication()
Create (if needed) and return an Application instance for this web application.

abstract public void setApplication(Application application)

Web Application

Throws NullPointerException: application null

setApplication

public abstract void setApplication(Application application)

Replace the Application instance that will be returned for this web application.

Parameters:
  application - The replacement Application instance

Throws:
  NullPointerException - if application is null.
javax.resource.spi  Class
ApplicationServerInternalException

java.lang.Object
  ↓ java.lang.Throwable
      ↓ java.lang.Exception
          ↓ javax.resource.ResourceException
              ↓ javax.resource.spi.ApplicationServerInternalException

All Implemented Interfaces:
  Serializable

public class ApplicationServerInternalException
  extends ResourceException

Extends: Throwable > Exception > ResourceException

ApplicationServerInternalException
  version 1.0

An ApplicationServerInternalException is thrown by an application server to indicate error conditions specific to an application server. These error conditions can be related to either configuration related errors or implementation of mechanisms internal to an application server (example: connection pooling, thread management).

Version:
  1.0

Author:
  Rahul Sharma, Ram Jeyaraman

See Also:
  Serialized Form

-------------------

Constructor Summary

**ApplicationServerInternalException()**

Constructs a new instance with null as its detail message.

**ApplicationServerInternalException(String message)**

Constructs a new instance with the specified detail message.

**ApplicationServerInternalException(String message, String errorCode)**

Constructs a new throwable with the specified detail message and an error code.

**ApplicationServerInternalException(String message, Throwable cause)**

Constructs a new throwable with the specified detail message and cause.

**ApplicationServerInternalException(Throwable cause)**

Constructs a new throwable with the specified cause.

---

**Method Summary**

Methods inherited from class javax.resource.ResourceException
- getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException

Methods inherited from class java.lang.Throwable
- fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

**Constructor Detail**

public ApplicationServerInternalException()
null
public ApplicationServerInternalException()
    Constructs a new instance with null as its detail message.

public ApplicationServerInternalException(String message)
    Constructs a new instance with the specified detail message.
    Parameters:
    message - the detail message.

public ApplicationServerInternalException(Throwable cause)
    Constructs a new throwable with the specified cause.
    Parameters:
public ApplicationServerInternalException(String message, Throwable cause)
\hspace{1cm} cause \hspace{1cm} throwable

message
cause

ApplicationServerInternalException

public ApplicationServerInternalException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public ApplicationServerInternalException(String message, String errorCode)
\hspace{1cm} throwable

message
errorCode

ApplicationServerInternalException

public ApplicationServerInternalException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.
Parameters:

message - a description of the exception.
errorCode - a string specifying the vendor specific error code.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.interceptor Annotation Type AroundInvoke

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface AroundInvoke

Implements: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  FRAMES  NO FRAMES
SUMMARY: REQUIRED | OPTIONAL  DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
</tr>
</tbody>
</table>
public class ArrayDataModel extends DataModel

ArrayDataModel is a convenience implementation of DataModel that wraps an array of Java objects.

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayDataModel()</td>
<td>Construct a new ArrayDataModel with no specified wrapped data.</td>
</tr>
<tr>
<td>ArrayDataModel(Object[] array)</td>
<td>Construct a new ArrayDataModel wrapping the specified array.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getRowCount()</td>
<td>If there is wrappedData available, return the length of the array.</td>
</tr>
<tr>
<td>Object getRowData()</td>
<td>If row data is available, return the array element at the index specified by rowIndex.</td>
</tr>
<tr>
<td>getRowIndex()</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>Return the zero-relative index of the currently selected row.</td>
</tr>
<tr>
<td>------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>Object</td>
<td><strong>getWrappedData()</strong> Return the object representing the data wrapped by this DataModel, if any.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isRowAvailable()</strong> Return true if there is wrappedData available, and the current value of rowIndex is greater than or equal to zero, and less than the length of the array.</td>
</tr>
<tr>
<td>void</td>
<td><strong>setRowIndex(int rowIndex)</strong> Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row.</td>
</tr>
<tr>
<td>void</td>
<td><strong>setWrappedData(Object data)</strong> Set the object representing the data collection wrapped by this DataModel.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.model.DataModel
- addDataModelListener, getDataModelListeners, removeDataModelListener

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

```java
public ArrayDataModel()
```

ArrayDataModel
public ArrayDataModel()

Construct a new ArrayDataModel with no specified wrapped data.

public ArrayDataModel(Object[] array)

ArrayDataModel

array

ArrayDataModel

public ArrayDataModel(Object[] array)

Construct a new ArrayDataModel wrapping the specified array.

Parameters:
array - Array to be wrapped (if any)

Method Detail

public boolean isRowAvailable()

wrappedData rowIndex 0 true false

Throws FacesException:

isRowAvailable

public boolean isRowAvailable()

Return true if there is wrappedData available, and the current value of rowIndex is greater than or equal to zero, and less than the length of the array. Otherwise, return false.
public int getRowCount()

wrappedData  wrappedData -1

Throws: FacesException

getRowCount

public int getRowCount()

If there is wrappedData available, return the length of the array. If no wrappedData is available, return -1.

Specified by: getRowCount in class DataModel
Throws: FacesException - if an error occurs getting the row count

public Object getRowData()

rowIndex  null

Throws: FacesException
Throws: IllegalArgumentException:

getRowData

public Object getRowData()
If row data is available, return the array element at the index specified by `rowIndex`. If no wrapped data is available, return `null`.

**Specified by:**

`getRowData` in class `DataModel`

**Throws:**

- `FacesException` - if an error occurs getting the row data
- `IllegalArgumentException` - if now row data is available at the currently specified row index

---

```java
public int getRowIndex()

Throws  FacesException: NullPointerException
```

**getRowIndex**

```java
public int getRowIndex()

Description copied from class: `DataModel`

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no `wrappedData` is available, return `-1`.

**Specified by:**

`getRowIndex` in class `DataModel`

**Throws:**

- `FacesException` - if an error occurs getting the row index

---

```java
public void setRowIndex(int rowIndex)

Throws  FacesException: NullPointerException
        IllegalArgumentException: NullPointerException rowIndex
```

**setRowIndex**

```java
public void setRowIndex(int rowIndex)

Throws  FacesException: NullPointerException
        IllegalArgumentException: NullPointerException rowIndex
```
public void setRowIndex(int rowIndex)

Description copied from class: DataModel

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the isRowAvailable() method to detect whether row data will be available for use by the getRowData() method.

If there is no wrappedData available when this method is called, the specified rowIndex is stored (and may be retrieved by a subsequent call to getRowData()), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a DataModelEvent will be sent to the rowSelected() method of all registered DataModelListeners.

Specified by:
setRowIndex in class DataModel

Parameters:
   rowIndex - The new zero-relative index (must be non-negative)

Throws:
   FacesException - if an error occurs setting the row index
   IllegalArgumentException - if rowIndex is less than -1

---

public Object getWrappedData()

getWrappedData

public Object getWrappedData()

Description copied from class: DataModel

Return the object representing the data wrapped by this DataModel, if any.
public void setWrappedData(Object data)

Throws  ClassCastException:  data null Java

setWrappedData

Descripion copied from class: DataModel

Set the object representing the data collection wrapped by this DataModel. If the specified data is null, detach this DataModel from any previously wrapped data collection instead.

If data is non-null, the currently selected row index must be set to zero, and a DataModelEvent must be sent to the rowSelected() method of all registered DataModelListener indicating that this row is now selected.

Specified by:
  setWrappedData in class DataModel

Parameters:
  data - Data collection to be wrapped, or null to detach from any previous data collection

Throws:
  ClassCastException - if data is non-null and is not an array of Java objects.
to license terms.

PS:
javax.el Class ArrayELResolver

java.lang.Object
  └─javax.el.ELResolver
      └─javax.el.ArrayELResolver

public class ArrayELResolver
  extends ELResolver

Extends: ELResolver

Java base

#isReadOnly   true#setValue   PropertyNotWritableException

ELResolver    CompositeELResolver    ELResolver
javadoc

since    JSP 2.1
See also  javax.el.CompositeELResolver, javax.el.ELResolver

Defines property resolution behavior on arrays.

This resolver handles base objects that are Java language arrays. It accepts any object as a property and coerces that object into an integer index into the array. The resulting value is the value in the array at that index.

This resolver can be constructed in read-only mode, which means that #isReadOnly will always return true and setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always throw PropertyNotWritableException.

ELResolvers are combined together using CompositeELResolver, to
define rich semantics for evaluating an expression. See the javadocs for
ELResolver for details.

Since:
JSP 2.1
See Also:
CompositeELResolver, ELResolver

---

Field Summary

**Fields inherited from class javax.el.ELResolver**

<table>
<thead>
<tr>
<th>RESOLVABLE_AT_DESIGN_TIME, TYPE</th>
</tr>
</thead>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th><strong>ArrayELResolver()</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates a new read/write ArrayELResolver.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>ArrayELResolver(boolean isReadOnly)</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates a new ArrayELResolver whose read-only status is determined by the given parameter.</td>
</tr>
</tbody>
</table>

Method Summary

| **Class<?>**
| `getCommonPropertyType(ELContext context, Object base)` |
| If the base object is a Java language array, returns the most general type that this resolver accepts for the property argument. |

| **Iterator<FeatureDescriptor>**
| `getFeatureDescriptors(ELContext context, Object base)` |
| Always returns null, since there is no reason to iterate through set set of all integers. |

| **Class<?>**
| `getType(ELContext context, Object base, Object property)` |
| If the base object is an array, returns the most general acceptable type for a value in this
If the base object is a Java language array, returns the value at the given index.

If the base object is a Java language array, returns whether a call to `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` will always fail.

If the base object is a Java language array, attempts to set the value at the given index with the given value.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ArrayELResolver()

/  

ArrayELResolver

public ArrayELResolver()

Creates a new read/write ArrayELResolver.

public ArrayELResolver(boolean isReadOnly)
ArrayELResolver

```
ArrayELResolver

isReadOnly
true  false
```

ArrayELResolver

public ArrayELResolver(boolean isReadOnly)

Creates a new ArrayELResolver whose read-only status is determined by the given parameter.

Parameters:
- isReadOnly - true if this resolver cannot modify arrays; false otherwise.

Method Detail

public Class<T> getType(ELContext context, Object base, Object property)

```
base
array  ELContext  propertyResolved  true
true

base  array  base.getClass().getComponentType()
context  Java  base
property
return  ELContext  propertyResolved  true

Throws  PropertyNotFoundException:
Throws  NullPointerException: context null
Throws  ELException: cause
```
getType

public Class<?〉 getType(ELContext context, Object base, Object property)

If the base object is an array, returns the most general acceptable type for a value in this array.

If the base is an array, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Assuming the base is an array, this method will always return base.getClass().getComponentType(), which is the most general type of component that can be stored at any given index in the array.

Specified by:
   getType in class ELResolver
Parameters:
   context - The context of this evaluation.
   base - The array to analyze. Only bases that are Java language arrays are handled by this resolver.
   property - The index of the element in the array to return the acceptable type for. Will be coerced into an integer, but otherwise ignored by this resolver.
Returns:
   If the propertyResolved property of ELContext was set to true, then the most general acceptable type; otherwise undefined.
Throws:
   PropertyNotFoundException - if the given index is out of bounds for this array.
   NullPointerException - if context is null
   ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.
public Object getValue(ELContext context, Object base, Object property)

base  Java  property
IllegalArgumentException  null

base  Java  ELContext  propertyResolved
true  true

collection  context
base  Java  base
property
return  ELContext  propertyResolved  true  null
Threws  IllegalArgumentException:
Threws  NullPointerException:  context  null
Threws  ELException:  cause

getValue

public Object getValue(ELContext context, Object base, Object property)

If the base object is a Java language array, returns the value at the given index. The index is specified by the property argument, and coerced into an integer. If the coercion could not be performed, an IllegalArgumentException is thrown. If the index is out of bounds, null is returned.

If the base is a Java language array, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Specified by:
getValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - The array to analyze. Only bases that are Java language arrays are handled by this resolver.

property - The index of the value to be returned. Will be coerced into an integer.

Returns:
If the propertyResolved property of ELContext was set to true, then the value at the given index or null if the index was out of bounds. Otherwise, undefined.

Throws:
- `IllegalArgumentException` - if the property could not be coerced into an integer.
- `NullPointerException` - if context is null.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public void setValue(ELContext context, Object base, Object property, Object val) {
    base - Java
    propertyResolved - true
    ELContext - true
    propertyNotWritableException

    context - Java base
    property - val

    Throws: ClassCastException:
    Throws: NullPointerException: context null
    Throws: IllegalArgumentException:
    Throws: PropertyNotWritableException:
    Throws: PropertyNotFoundException:
    Throws: ELException: cause
```
setValue

public void setValue(ELContext context,
    Object base,
    Object property,
    Object val)

If the base object is a Java language array, attempts to set the value at the given index with the given value. The index is specified by the property argument, and coerced into an integer. If the coercion could not be performed, an IllegalArgumentException is thrown. If the index is out of bounds, a PropertyNotFoundException is thrown.

If the base is a Java language array, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller can safely assume no value was set.

If this resolver was constructed in read-only mode, this method will always throw PropertyNotWritableException.

Specified by: setValue in class ELResolver

Parameters:
    context - The context of this evaluation.
    base - The array to be modified. Only bases that are Java language arrays are handled by this resolver.
    property - The index of the value to be set. Will be coerced into an integer.
    val - The value to be set at the given index.

Throws:
    ClassCastException - if the class of the specified element prevents it from being added to this array.
    NullPointerException - if context is null.
    IllegalArgumentException - if the property could not be coerced into an integer, or if some aspect of the specified element prevents it from being added to this array.
    PropertyNotWritableException - if this resolver was constructed
in read-only mode.

*PropertyNotFoundException* - if the given index is out of bounds for this array.

*ELException* - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public boolean isReadOnly(ELContext context, Object base, Object property)
```

If the base object is a Java language array, returns whether a call to `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` will always fail.

If the base is a Java language array, the `propertyResolved` property
of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

If this resolver was constructed in read-only mode, this method will always return true. Otherwise, it returns false.

Specified by:
   isReadOnly in class ELResolver

Parameters:
   context - The context of this evaluation.
   base - The array to analyze. Only bases that are a Java language array are handled by this resolver.
   property - The index of the element in the array to return the acceptable type for. Will be coerced into an integer, but otherwise ignored by this resolver.

Returns:
   If the propertyResolved property of ELContext was set to true, then true if calling the setValue method will always fail or false if it is possible that such a call may succeed; otherwise undefined.

Throws:
   PropertyNotFoundException - if the given index is out of bounds for this array.
   NullPointerException - if context is null
   ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

public java.util.Iterator<E>
getFeatureDescriptors(ELContext context, Object base)
null

#getCommonPropertyType

context
base Java base
getFeatureDescriptors

public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext context, Object base)

Always returns null, since there is no reason to iterate through set of all integers.

The getCommonPropertyType(javax.el.ELContext, java.lang.Object) method returns sufficient information about what properties this resolver accepts.

Specified by:
   getFeatureDescriptors in class ELResolver

Parameters:
   context - The context of this evaluation.
   base - The array to analyze. Only bases that are a Java language array are handled by this resolver.

Returns:
   null.

See Also:
   FeatureDescriptor

public Class<T> getCommonPropertyType(ELContext context, Object base)

<table>
<thead>
<tr>
<th>base</th>
<th>Java</th>
<th>property</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>context</td>
<td></td>
<td>null</td>
<td></td>
</tr>
<tr>
<td>base</td>
<td></td>
<td>base</td>
<td>null</td>
</tr>
<tr>
<td>return</td>
<td>base</td>
<td>Java</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>null</td>
<td>Integer.class</td>
</tr>
</tbody>
</table>
getCommonPropertyType

public Class<?>
getCommonPropertyType(ELContext context, Object base)

If the base object is a Java language array, returns the most general type that this resolver accepts for the property argument. Otherwise, returns null.

Assuming the base is an array, this method will always return Integer.class. This is because arrays accept integers for their index.

Specified by:
getCommonPropertyType in class ELResolver

Parameters:
context - The context of this evaluation.
base - The array to analyze. Only bases that are a Java language array are handled by this resolver.

Returns:
null if base is not a Java language array; otherwise Integer.class.
javax.xml.registry.infomodel Interface Association

All Superinterfaces:
   ExtensibleObject, RegistryObject

public interface Association

extends RegistryObject

Implements: RegistryObject

RegistryObject 0 RegistryObject Association RegistryObject
Association

ClassificationScheme ClassificationScheme
ClassificationScheme 1 ClassificationScheme 1
NAICS ClassificationScheme NAICS ClassificationScheme
Association

1. RegistryObject Association
Association RegistryObject RegistryObject Association sourceObject targetObject Association sourceObject targetObject 1 NAICS ClassificationScheme sourceObject NAICS ClassificationScheme targetObject associationType sourceObject targetObject
Association

Association associationType associationType Concept
Concept JAXR associationType ClassificationScheme
Supersedes associationType Concept
Association

Association User "u" RegistryObject "o1" "o2" Association "a" Association "a" RegistryObject "o1" "o2" User "u"
Association (intramural association) 2 1

2. Association
Association

User "u1" RegistryObject "o1" "o2" Association "a" Association "a" User "u1" RegistryObject "o1" "o2" User "u2" User "u3" Association User Association User (extramural association)Association isExtramural Association Association true 3 1 User 3 Association sourceObject targetObject

3. Association
Association
Association
Association
Association
User "u2" "u3" sourceObject targetObject
BusinessLifeCycleManager confirmAssociation
Association

Association  Association  Association
A RegistryObject instance may be associated with zero or more RegistryObject instances. The information model defines an Association interface, an instance of which may be used to associate any two RegistryObject instances.
Example of an Association

One example of such an association is between two ClassificationScheme instances, where one ClassificationScheme supersedes the other ClassificationScheme as shown in Figure 1. This may be the case when a new version of a ClassificationScheme is submitted. In Figure 1, we see how an Association is defined between a new version of the NAICS ClassificationScheme and an older version of the NAICS ClassificationScheme.

Figure 1. Example of RegistryObject Association
Source and Target Objects

An Association instance represents an association between a source RegistryObject and a target RegistryObject. These are referred to as sourceObject and targetObject for the Association instance. It is important which object is the sourceObject and which is the targetObject as it determines the directional semantics of an Association. In the example in Figure 1, it is important to make the newer version of NAICS ClassificationScheme be the sourceObject and the older version of NAICS be the targetObject because the associationType implies that the sourceObject supersedes the targetObject (and not the other way around).
Association Types

Each Association must have an associationType attribute that identifies the type of that association. The associationType attribute is a reference to an enumeration Concept as defined by the predefined associationType ClassificationScheme in the JAXR specification. Our example uses the pre-defined associationType Concept named Supersedes.
Intramural Associations

A common use case for the Association interface is when a User "u" creates an Association "a" between two RegistryObjects "o1" and "o2" where association "a" and RegistryObjects "o1" and "o2" are objects that were created by the same User "u". This is the simplest use case where the association is between two objects that are owned by same User that is defining the Association. Such associations are referred to as intramural associations. Figure 2 extends the previous example in Figure 1 for the intramural association case.

Figure 2. Example of Intramural Association
Extramural Association

The information model also allows a more sophisticated use case where a User "u1" creates an Association "a" between two RegistryObjects "o1" and "o2" where association "a" is owned by User "u1", but RegistryObjects "o1" and "o2" are owned by User "u2" and User "u3" respectively. In this use case the Association is being defined where either or both objects that are being associated are owned by a User different from the User defining the Association. Such associations are referred to as extramural associations. The Association interface provides a convenience method called isExtramural that returns true if the Association instance is an extramural Association. Figure 3 extends the previous example in Figure 1 for the extramural association case. Note that it is possible for an extramural association to have two distinct Users rather than three distinct Users as shown in Figure 3. In such case, one of the two users owns two of the three objects involved (Association, sourceObject and targetObject).

Figure 3. Example of Extramural Association
Confirmation of an Association

An association may need to be confirmed by the parties whose objects are involved in that Association. This section describes the semantics of confirmation of an association by the parties involved.

Confirmation of Intramural Associations

Intramural associations may be viewed as declarations of truth and do not require any explicit steps to confirm that Association as being true. In other words, intramural associations are implicitly considered be confirmed.

Confirmation of Extramural Associations

Extramural associations may be viewed as a unilateral assertion that may not be viewed as truth until it has been confirmed by the other (extramural) parties (Users "u2" and "u3" in example). The confirmAssociation method on the BusinessLifeCycleManager interface may be called by the extramural parties that own the sourceObject or targetObject.
Visibility of Unconfirmed Associations

Extramural associations require each extramural party to confirm the assertion being made by the extramural Association before the Association is visible to 3rd parties that are not involved in the Association. This ensures that unconfirmed Associations are not visible to 3rd party registry clients.
Possible Confirmation States

Assume the most general case where there are three distinct User instances as shown in Figure 23 for an extramural Association. The extramural Association needs to be confirmed by both the other (extramural) parties (Users "u2" and "u3" in example) in order to be fully confirmed. The methods `isConfirmedBySourceOwner` and `isConfirmedByTargetOwner` in the Association interface provide access to confirmation state for both the `sourceObject` and `targetObject`. A third convenience method called `isConfirmed` provides a way to determine whether the Association is fully confirmed or not. So there are the following four possibilities related to confirmation state of an extramural Association:

- The Association is confirmed neither by the owner of the `sourceObject` nor is it confirmed by owner of `targetObject`.
- The Association is confirmed by the owner of the `sourceObject` but it is not confirmed by owner of `targetObject`.
- The Association is not confirmed by the owner of the `sourceObject` but it is confirmed by owner of `targetObject`.
- The Association is confirmed by the owner of the `sourceObject` and it is confirmed by owner of `targetObject`. This is the only state where the Association is fully confirmed.

**Author:**
Farrukh S. Najmi

**See Also:**
*RegistryObject*

---

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Concept</strong></td>
</tr>
<tr>
<td><strong>RegistryObject</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### RegistryObject

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isConfirmed()</td>
<td>Determines whether an Association has been confirmed completely.</td>
</tr>
<tr>
<td>boolean isConfirmedBySourceOwner()</td>
<td>Determines whether an Association has been confirmed by the owner of the source object.</td>
</tr>
<tr>
<td>boolean isConfirmedByTargetOwner()</td>
<td>Determines whether an Association has been confirmed by the owner of the target object.</td>
</tr>
<tr>
<td>boolean isExtramural()</td>
<td>Determines whether an Association is extramural or not.</td>
</tr>
<tr>
<td>void setAssociationType(Concept associationType)</td>
<td>Sets the association type for this Association.</td>
</tr>
<tr>
<td>void setSourceObject(RegistryObject srcObject)</td>
<td>Sets the Object that is the source of this Association.</td>
</tr>
<tr>
<td>void setTargetObject(RegistryObject targetObject)</td>
<td>Sets the Object that is the target of this Association.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface**

**javax.xml.registry.infomodel.RegistryObject**

addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML

**Methods inherited from interface**

**javax.xml.registry.infomodel.ExtensibleObject**

addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots
### Method Detail

**public RegistryObject getSourceObject() throws JAXRException**

**Association Object**

0

```java
    return Association RegistryObject
```

**Throws**

- **JAXRException** - If the JAXR provider encounters an internal error

**getSourceObject**

**RegistryObject getSourceObject() throws JAXRException**

Gets the Object that is the source of this Association.

**Capability Level: 0**

**Returns:**

The RegistryObject that is the source object of this Association

**Throws:**

- **JAXRException** - If the JAXR provider encounters an internal error

---

**public void setSourceObject(RegistryObject srcObject) throws JAXRException**

**Association Object**

0

```java
    srcObject Association RegistryObject
```

**setSourceObject**

**RegistryObject setSourceObject(RegistryObject srcObject) throws JAXRException**

**Association Object**

0

```java
    srcObject Association RegistryObject
```
setSourceObject

```java
void setSourceObject(RegistryObject srcObject)
throws JAXRException
```

Sets the Object that is the source of this Association.

**Capability Level: 0**

**Parameters:**
- `srcObject` - the RegistryObject that is the source object of this Association

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

```java
public RegistryObject getTargetObject() throws JAXRException
```

**Association Object**

```java
return Association RegistryObject
```

**Throws**
- `JAXRException`: JAXR

---

getTargetObject

```java
RegistryObject getTargetObject()
throws JAXRException
```

Gets the Object that is the target of this Association.

**Capability Level: 0**
Returns:
The RegistryObject that is the target object of this Association

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setTargetObject(RegistryObject targetObject)
throws JAXRException

Association Object

0

targetObject Association RegistryObject
Throws JAXRException: JAXR

setTargetObject

void setTargetObject(RegistryObject targetObject)
throws JAXRException

Sets the Object that is the target of this Association.

Capability Level: 0

Parameters:
targetObject - the RegistryObject that is the target object of this Association

Throws:
JAXRException - If the JAXR provider encounters an internal error

public Concept getAssociationType() throws JAXRException

Association
getAssociationType

Concept getAssociationType()
throws JAXRException

Gets the association type for this Association.

Capability Level: 0

Returns:
The association type for this Association which is a Concept in
the AssociationType ClassificationScheme

Throws:
JAXRException - If the JAXR provider encounters an internal
error

public void setAssociationType(Concept associationType)
throws JAXRException

setAssociationType
Sets the association type for this Association.

**Capability Level: 0**

**Parameters:**

- `associationType` - the association type for this Association which is a Concept in the AssociationType ClassificationScheme

**Throws:**

- `JAXRException` - If the JAXR provider encounters an internal error

---

```java
public boolean isExtramural() throws JAXRException {
    Association sourceObject = User / targetObject
    Association targetObject = User

    return sourceObject / targetObject
```

**Determines whether an Association is extramural or not.**

An Extramural Association must be confirmed by the User(s) that own the source and/or target object, if they are different from the User who creates this extramural association. Both the sourceObject and targetObject owners must confirm an extramural Association, in order for it to be visible to third parties.
Capability Level: 0

Returns:
true if the sourceObject and/or the targetObject are owned by a User that is different from the User that created the Association;
false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error

public boolean isConfirmedBySourceOwner() throws JAXRException

Determines whether an Association has been confirmed by the owner of the source object.

Capability Level: 0

Returns:
true if the association has been confirmed by the owner of the sourceObject; false otherwise. For intramural Associations always return true

Throws:
JAXRException - If the JAXR provider encounters an internal error
public boolean isConfirmedByTargetOwner() throws JAXRException

Determines whether an Association has been confirmed by the owner of the target object.

**Capability Level:** 0

**Returns:**
- `true` if the association has been confirmed by the owner of the targetObject; `false` otherwise. For intramural Associations always return `true`

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

public boolean isConfirmed() throws JAXRException

isConfirmed  true Association

0
isConfirmed

boolean isConfirmed()

throws JAXRException

Determines whether an Association has been confirmed completely.

An association should only be visible to third parties (not involved with the Association) if isConfirmed returns true.

Capability Level: 0

Returns:
true if the isConfirmedBySourceOwner and isConfirmedByTargetOwner methods both return true; false otherwise. For intramural Associations always return true

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
isConfirmedBySourceOwner(), isConfirmedByTargetOwner()
PS:
javax.persistence Annotation Type AssociationOverride

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface AssociationOverride

This annotation is used to override a many-to-one or one-to-one mapping of property or field for an entity relationship.

The AssociationOverride annotation may be applied to an entity that extends a mapped superclass to override a many-to-one or one-to-one mapping defined by the mapped superclass. If the AssociationOverride annotation is not specified, the join column is mapped the same as in the original mapping.

Example:
@MappedSuperclass
public class Employee {
    ...
    @ManyToOne
    protected Address address;
    ...
}

@Entity
@AssociationOverride(name="address",
    joinColumns=@JoinColumn(name="ADDR_ID"))
// address field mapping overridden to ADDR_ID fk
public class PartTimeEmployee extends Employee {
    ...
}

Since:
Java Persistence 1.0

See Also:
OneToOne, ManyToOne, MappedSuperclass
### joinColumns

The join column that is being mapped to the persistent attribute.

### name

The name of the relationship property whose mapping is being overridden if property-based access is being used, or the name of the relationship field if field-based access is used.

---

#### Element Detail

**name**

public abstract `String` name

The name of the relationship property whose mapping is being overridden if property-based access is being used, or the name of the relationship field if field-based access is used.

#### joinColumns

public abstract `JoinColumn[]` joinColumns

The join column that is being mapped to the persistent attribute. The mapping type will remain the same as is defined in the mapped superclass.
javax.persistence Annotation Type AssociationOverrides

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface AssociationOverrides

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

@MappedSuperclass
public class Employee {

    @Id protected Integer id;
    @Version protected Integer version;
    @ManyToOne protected Address address;
    @OneToOne protected Locker locker;

    public Integer getId() { ... }
    public void setId(Integer id) { ... }
    public Address getAddress() { ... }
    public void setAddress(Address address) { ... }
    public Locker getLocker() { ... }
    public void setLocker(Locker locker) { ... }
}

@Entity
@AssociationOverrides({
    @AssociationOverride(name="address",
        joinColumns=@JoinColumn("ADDR_ID")),
    @AttributeOverride(name="locker",
        joinColumns=@JoinColumn("LCKR_ID")))
public PartTimeEmployee { ... }

since  Java Persistence 1.0  en
This annotation is used to override mappings of multiple many-to-one or one-to-one relationship properties or fields.

Example:
@MappedSuperclass
public class Employee {
    @Id protected Integer id;
    @Version protected Integer version;
    @ManyToOne protected Address address;
    @OneToOne protected Locker locker;

    public Integer getId() { ... }
    public void setId(Integer id) { ... }
    public Address getAddress() { ... }
    public void setAddress(Address address) { ... }
    public Locker getLocker() { ... }
    public void setLocker(Locker locker) { ... }
}

@Entity
@AssociationOverrides({
    @AssociationOverride(name="address",
            joinColumns=@JoinColumn("ADDR_ID")),
    @AttributeOverride(name="locker",
            joinColumns=@JoinColumn("LCKR_ID")))
public PartTimeEmployee { ... }

Since:
Java Persistence 1.0

---

**Required Element Summary**

<table>
<thead>
<tr>
<th>AssociationOverride[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mapping overrides of relationship properties or fields</td>
</tr>
</tbody>
</table>

**Element Detail**
abstract public AssociationOverride[] value()

value

public abstract AssociationOverride[] value

Mapping overrides of relationship properties or fields

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws Interface AsyncHandler<T>

public interface AsyncHandler<T>

AsyncHandler

since JAX-WS 2.0

The AsyncHandler interface is implemented by clients that wish to receive callback notification of the completion of service endpoint operations invoked asynchronously.

Since:
  JAX-WS 2.0

---

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>handleResponse(T res)</td>
</tr>
</tbody>
</table>

- **handleResponse**
  - Method: `void handleResponse(Response<T> res)`
  - Called when the response to an asynchronous operation is available.

---

Method Detail

handleResponse

void **handleResponse**(Response<T> res)

  Called when the response to an asynchronous operation is available.

  **Parameters:**
  
  res - The response to the operation invocation.
public abstract class AttachmentMarshaller
extends Object

JAXB

API JAXB MIME JAXB MIME MIME MIME

#isXOPPackage() true XOP
#addMtomAttachment(DataHandler, String, String)

#addSwaRefAttachment(DataHandler) WS-I Attachment Profile 1.0
{http://ws-i.org/profiles/basic/1.1/xsd}swaRef JAXB

since JAXB 2.0

See also setAttachmentMarshaller(AttachmentMarshaller), XML-binary Optimized Packaging, WS-I Attachments Profile Version 1.0.

Enable JAXB marshalling to optimize storage of binary data.

This API enables an efficient cooperative creation of optimized binary data formats between a JAXB marshalling process and a MIME-based package processor. A JAXB implementation marshals the root body of a MIME-based package, delegating the creation of referenceable MIME parts to the MIME-based package processor that implements this abstraction.

XOP processing is enabled when isXOPPackage() is true. See
addMtomAttachment(DataHandler, String, String) for details.

WS-I Attachment Profile 1.0 is supported by addSwaRefAttachment(DataHandler) being called by the marshaller for each JAXB property related to {http://ws-i.org/profiles/basic/1.1/xsd}swaRef.

Since:
    JAXB 2.0
Author:
    Marc Hadley, Kohsuke Kawaguchi, Joseph Fialli
See Also:
    Marshaller.setAttachmentMarshaller(AttachmentMarshaller), XML-binary Optimized Packaging, WS-I Attachments Profile Version 1.0.

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AttachmentMarshaller()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addMtomAttachment(byte[] data, int offset, int length, String mimeType, String elementNamespace, String elementLocalName)</td>
<td>Consider binary data for optimized binary storage as an attachment.</td>
</tr>
<tr>
<td>addMtomAttachment(DataHandler data, String elementNamespace, String elementLocalName)</td>
<td>Consider MIME content data for optimized binary storage as an attachment.</td>
</tr>
<tr>
<td>addSwaRefAttachment(DataHandler data)</td>
<td>Add MIME data as an attachment and return attachment's content-id, cid.</td>
</tr>
<tr>
<td>isXOPPackage()</td>
<td>Read-only property that returns true if JAXB marshaller should enable XOP creation.</td>
</tr>
</tbody>
</table>
### Constructor Detail

**public AttachmentMarshaller()**

### Method Detail

**abstract public String addMtomAttachment(DataHandler data, String elementNamespace, String elementLocalName)**

```
MIME data

Creating XOP Packages 3
true JAXB "base64Binary"

id cid MIME MIME cidcid JAXB
cid XOP null JAXB
base64binary

/ MIME
```

```
data infoset xmime:
data MIMI
```

```
data.getContentType() MIMI
```
Consider MIME content data for optimized binary storage as an attachment.

This method is called by JAXB marshal process when isXOPPackage() is true, for each element whose datatype is "base64Binary", as described in Step 3 in Creating XOP Packages.

The method implementor determines whether data shall be attached separately or inlined as base64Binary data. If the implementation chooses to optimize the storage of the binary data as a MIME part, it is responsible for attaching data to the MIME-based package, and then assigning an unique content-id, cid, that identifies the MIME part within the MIME message. This method returns the cid, which enables the JAXB marshaller to marshal a XOP element that refers to that cid in place of marshalling the binary data. When the method returns null, the JAXB marshaller inlines data as base64binary data.

The caller of this method is required to meet the following constraint. If the element infoset item containing data has the attribute xmime:contentType or if the JAXB property/field representing data is annotated with a known MIME type, data.getContentType() should be set to that MIME type.
The `elementNamespace` and `elementLocalName` parameters provide the context that contains the binary data. This information could be used by the MIME-based package processor to determine if the binary data should be inlined or optimized as an attachment.

**Parameters:**
- `data` - represents the data to be attached. Must be non-null.
- `elementNamespace` - the namespace URI of the element that encloses the base64Binary data. Can be empty but never null.
- `elementLocalName` - The local name of the element. Always a non-null valid string.

**Returns:**
- a valid content-id URI (see RFC 2387) that identifies the attachment containing `data`. Otherwise, null if the attachment was not added and should instead be inlined in the message.

**See Also:**
- XML-binary Optimized Packaging, Describing Media Content of Binary Data in XML

```java
abstract public String addMtomAttachment(byte[] data, int offset, int length, String mimeType, String elementNamespace, String elementLocalName)

data

MIME “application/octet-stream”

<table>
<thead>
<tr>
<th>elementNamespace</th>
<th>elementLocalName</th>
<th>MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td>null</td>
<td>(data,offset,length)</td>
</tr>
<tr>
<td><code>offset</code></td>
<td>array.length</td>
<td></td>
</tr>
<tr>
<td><code>length</code></td>
<td>array.length</td>
<td></td>
</tr>
<tr>
<td><code>mimeType</code></td>
<td>JAXB MIME “application/octet-stream” null</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>elementNamespace</th>
<th>elementLocalName</th>
<th>MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>base64Binary</td>
<td>URI</td>
<td>null</td>
</tr>
</tbody>
</table>

| return           | data, id, URI, cid | null |
```
See also addMtomAttachment(DataHandler, String, String)

**addMtomAttachment**

```java
public abstract String addMtomAttachment(byte[] data,
                                          int offset,
                                          int length,
                                          String mimeType,
                                          String elementNamespace,
                                          String elementLocalName)
```

Consider binary data for optimized binary storage as an attachment.

Since content type is not known, the attachment's MIME content type must be set to "application/octet-stream".

The `elementNamespace` and `elementLocalName` parameters provide the context that contains the binary data. This information could be used by the MIME-based package processor to determine if the binary data should be inlined or optimized as an attachment.

**Parameters:**
- `data` - represents the data to be attached. Must be non-null. The actual data region is specified by `(data,offset,length)` tuple. 
- `offset` - The offset within the array of the first byte to be read; must be non-negative and no larger than array.length 
- `length` - The number of bytes to be read from the given array; must be non-negative and no larger than array.length 
- `mimeType` - If the data has an associated MIME type known to JAXB, that is passed as this parameter. If none is known, "application/octet-stream". This parameter may never be null. 
- `elementNamespace` - the namespace URI of the element that encloses the base64Binary data. Can be empty but never null. 
- `elementLocalName` - The local name of the element. Always a non-null valid string.

**Returns:**
- content-id URI, cid, to the attachment containing data or null if data should be inlined.
public boolean isXOPPackage()

JAXB XOP true

true addMtomAttachment(...)

true XML Creating XOP Packages

http://www.w3.org/TR/2005/REC-xop10-20050125/#creating_xop_packages

Marshaller.marshal() IllegalStateException

"Ensure the Original XML Infoset contains no element information item with a [namespace name] of "http://www.w3.org/2004/08/xop/include" and a [local name] of Include"

true addMtomAttachment(...) id Creating XOP Packages 5 MIME application/xop+xml

return MIME XOP true

isXOPPackage

public boolean isXOPPackage()

Read-only property that returns true if JAXB marshaller should enable XOP creation.

This value must not change during the marshalling process. When this value is true, the addMtomAttachment(...) method is invoked when the appropriate binary datatypes are encountered by the marshal process.
Marshaller.marshal() must throw IllegalStateException if this value is true and the XML content to be marshalled violates Step 1 in Creating XOP Packages http://www.w3.org/TR/2005/REC-xop10-20050125/#creating_xop_packages. "Ensure the Original XML Infoset contains no element information item with a [namespace name] of "http://www.w3.org/2004/08/xop/include" and a [local name] of Include"

When this method returns true and during the marshal process at least one call to addMtomAttachment(...) returns a content-id, the MIME-based package processor must label the root part with the application/xop+xml media type as described in Step 5 of Creating XOP Packages.

Returns:
true when MIME context is a XOP Package.

abstract public String addSwaRefAttachment(DataHandler data)

MIME data id cid

JAXB {http://ws-i.org/profiles/basic/1.1/xsd}swaRef /
MIME MIME id cid MIME

id cid XML

data null

return cid URI

WS-I Attachments Profile Version 1.0
R2928

addSwaRefAttachment

public abstract String addSwaRefAttachment(DataHandler data)

Add MIME data as an attachment and return attachment's content-id,
This method is called by JAXB marshal process for each element/attribute typed as {http://ws-i.org/profiles/basic/1.1/xsd}swaRef. The MIME-based package processor implementing this method is responsible for attaching the specified data to a MIME attachment, and generating a content-id, cid, that uniquely identifies the attachment within the MIME-based package.

Caller inserts the returned content-id, cid, into the XML content being marshalled.

**Parameters:**
- `data` - represents the data to be attached. Must be non-null.

**Returns:**
- must be a valid URI used as cid. Must satisfy Conformance Requirement R2928 from [WS-I Attachments Profile Version 1.0](http://ws-i.org).
javax.xml.soap  Class AttachmentPart

java.lang.Object
- javax.xml.soap.AttachmentPart

public abstract class AttachmentPart
extends Object

SOAPMessage  SOAPMessage 0 1 AttachmentPart
AttachmentPart MIME MIME /

AttachmentPart

1. MIME [RFC2045]
2.
3.  
   - Content-Type
      AttachmentPart [RFC2045] Content-Type
      Content-Type: application/xml

     ap  AttachmentPart
     ap.setMimeHeader("Content-Type", "application/xml");

AttachmentPart XML

AttachmentPart SOAPMessage.createAttachmentPart MIME
SOAPMessage.addAttachmentPart AttachmentPart

m  SOAPMessage  contentString1  String AttachmentPart
AttachmentPart  AttachmentPart  SOAPMessage

AttachmentPart ap1 = m.createAttachmentPart();
ap1.setContent(contentString1, "text/plain");
m.addAttachmentPart(ap1);
AttachmentPart jpegData jpeg

AttachmentPart ap2 = m.createAttachmentPart();
byte[] jpegData = ...;
ap2.setContent(new ByteArrayInputStream(jpegData), "image/jpeg");
m.addAttachmentPart(ap2);

getContent AttachmentPart DataContentHandler Object
MIME Java

String content1 = ap1.getContent();
java.io.InputStream content2 = ap2.getContent();

clearContent AttachmentPart
ap1.clearContent();

A single attachment to a SOAPMessage object. A SOAPMessage object may contain zero, one, or many AttachmentPart objects. Each AttachmentPart object consists of two parts, application-specific content and associated MIME headers. The MIME headers consists of name/value pairs that can be used to identify and describe the content.

An AttachmentPart object must conform to certain standards.

1. It must conform to MIME [RFC2045] standards
2. It MUST contain content
3. The header portion MUST include the following header:
   - Content-Type
     This header identifies the type of data in the content of an AttachmentPart object and MUST conform to [RFC2045]. The following is an example of a Content-Type header:
     
     Content-Type: application/xml

     The following line of code, in which ap is an AttachmentPart object, sets the header shown in the previous example.
ap.setMimeHeader("Content-Type", "application/xml");

There are no restrictions on the content portion of an AttachmentPart object. The content may be anything from a simple plain text object to a complex XML document or image file.

An AttachmentPart object is created with the method SOAPMessage.createAttachmentPart. After setting its MIME headers, the AttachmentPart object is added to the message that created it with the method SOAPMessage.addAttachmentPart.

The following code fragment, in which m is a SOAPMessage object and contentString1 is a String, creates an instance of AttachmentPart, sets the AttachmentPart object with some content and header information, and adds the AttachmentPart object to the SOAPMessage object.

```java
AttachmentPart ap1 = m.createAttachmentPart();
ap1.setContent(contentString1, "text/plain");
m.addAttachmentPart(ap1);
```

The following code fragment creates and adds a second AttachmentPart instance to the same message. jpegData is a binary byte buffer representing the jpeg file.

```java
AttachmentPart ap2 = m.createAttachmentPart();
byte[] jpegData = ...;
ap2.setContent(new ByteArrayInputStream(jpegData), "image/jpeg");
m.addAttachmentPart(ap2);
```

The getContent method retrieves the contents and header from an AttachmentPart object. Depending on the DataContentHandler objects present, the returned Object can either be a typed Java object corresponding to the MIME type or an InputStream object that contains the content as bytes.

```java
String content1 = ap1.getContent();
java.io.InputStream content2 = ap2.getContent();
```

The method clearContent removes all the content from an
AttachmentPart object but does not affect its header information.

    ap1.clearContent();

---

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttachmentPart()</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract void</td>
<td>addMimeHeader(String name, String value)</td>
<td>Adds a MIME header with the specified name and value to this AttachmentPart object.</td>
</tr>
<tr>
<td>abstract void</td>
<td>clearContent()</td>
<td>Clears out the content of this AttachmentPart object.</td>
</tr>
<tr>
<td>abstract Iterator</td>
<td>getAllMimeHeaders()</td>
<td>Retrieves all the headers for this AttachmentPart object as an iterator over the MimeHeader objects.</td>
</tr>
<tr>
<td>abstract InputStream</td>
<td>getBase64Content()</td>
<td>Returns an InputStream which can be used to obtain the content of AttachmentPart as Base64 encoded character data, this method would base64 encode the raw bytes of the attachment and return.</td>
</tr>
<tr>
<td>abstract Object</td>
<td>getContent()</td>
<td>Gets the content of this AttachmentPart object as a Java object.</td>
</tr>
<tr>
<td>String</td>
<td>getContentId()</td>
<td>Gets the value of the MIME header whose name is &quot;Content-ID&quot;.</td>
</tr>
<tr>
<td>String</td>
<td>getContentLocation()</td>
<td>Gets the value of the MIME header whose name is &quot;Content-Location&quot;.</td>
</tr>
<tr>
<td>String</td>
<td>getContentType()</td>
<td>Gets the value of the MIME header whose name is &quot;Content-Type&quot;.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>getDataHandler()</code></td>
<td>Gets the DataHandler object for this AttachmentPart object.</td>
<td></td>
</tr>
<tr>
<td><code>getMatchingMimeHeaders(String[] names)</code></td>
<td>Retrieves all MimeHeader objects that match a name in the given array.</td>
<td></td>
</tr>
<tr>
<td><code>getMimeHeader(String name)</code></td>
<td>Gets all the values of the header identified by the given String.</td>
<td></td>
</tr>
<tr>
<td><code>getNonMatchingMimeHeaders(String[] names)</code></td>
<td>Retrieves all MimeHeader objects whose name does not match a name in the given array.</td>
<td></td>
</tr>
<tr>
<td><code>getRawContent()</code></td>
<td>Gets the content of this AttachmentPart object as an InputStream as if a call had been made to getContent and no DataContentHandler had been registered for the content-type of this AttachmentPart.</td>
<td></td>
</tr>
<tr>
<td><code>getRawContentBytes()</code></td>
<td>Gets the content of this AttachmentPart object as a byte[] array as if a call had been made to getContent and no DataContentHandler had been registered for the content-type of this AttachmentPart.</td>
<td></td>
</tr>
<tr>
<td><code>getSize()</code></td>
<td>Returns the number of bytes in this AttachmentPart object.</td>
<td></td>
</tr>
<tr>
<td><code>removeAllMimeHeaders()</code></td>
<td>Removes all the MIME header entries.</td>
<td></td>
</tr>
<tr>
<td><code>removeMimeHeader(String header)</code></td>
<td>Removes all MIME headers that match the given name.</td>
<td></td>
</tr>
<tr>
<td><code>setBase64Content(InputStream content, String contentType)</code></td>
<td>Sets the content of this attachment part from the Base64 source InputStream and sets the value of the Content-Type header to the value contained in contentType. This method would first decode the base64 input and write</td>
<td></td>
</tr>
</tbody>
</table>
the resulting raw bytes to the attachment.

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>setContent</strong> (Object object, String contentType)</td>
<td>Sets the content of this attachment part to that of the given object and sets the value of the Content-Type header to the given type.</td>
<td></td>
</tr>
<tr>
<td><strong>setContentType</strong> (String contentType)</td>
<td>Sets the MIME header whose name is &quot;Content-Type&quot; with the given value.</td>
<td></td>
</tr>
<tr>
<td><strong>setMimeType</strong> (String contentLocation)</td>
<td>Sets the MIME header whose name is &quot;Content-Location&quot; with the given value.</td>
<td></td>
</tr>
<tr>
<td><strong>setRawContent</strong> (InputStream content, String contentType)</td>
<td>Sets the content of this attachment part to that contained by the InputStream content and sets the value of the Content-Type header to the value contained in contentType.</td>
<td></td>
</tr>
<tr>
<td><strong>setRawContentBytes</strong> (byte[] content, int offset, int len, String contentType)</td>
<td>Sets the content of this attachment part to that contained by the byte[] array content and sets the value of the Content-Type header to the value contained in contentType.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detail

public AttachmentPart()
clearContent

public abstract void clearContent()

Clears out the content of this AttachmentPart object. The MIME header portion is left untouched.

abstract public Object getContent() throws SOAPException
Java AttachmentPart Java (1)
DataContentHandler (2) Content-Type

MIME "text/plain" "text/html"
"text/xml" DataContentHandler MIME Java
MIME DataContentHandler InputStream

SAAJ java.lang.String Content-Type
text/plain javax.xml.transform.stream.StreamSource
Content-Type text/xml java.awt.Image
Content-Type image/gif image/jpeg
DataContentHandler DataContentHandler
java.io.InputStream
return AttachmentPart Java
Throws SOAPException: AttachmentPart

getContent

public abstract Object getContent() throws SOAPException

Gets the content of this AttachmentPart object as a Java object. The type of the returned Java object depends on (1) the DataContentHandler object that is used to interpret the bytes and (2)
the Content-Type given in the header.

For the MIME content types "text/plain", "text/html" and "text/xml", the DataContentHandler object does the conversions to and from the Java types corresponding to the MIME types. For other MIME types, the DataContentHandler object can return an InputStream object that contains the content data as raw bytes.

A SAAJ-compliant implementation must, as a minimum, return a java.lang.String object corresponding to any content stream with a Content-Type value of text/plain, a javax.xml.transform.stream.StreamSource object corresponding to a content stream with a Content-Type value of text/xml, a java.awt.Image object corresponding to a content stream with a Content-Type value of image/gif or image/jpeg. For those content types that an installed DataContentHandler object does not understand, the DataContentHandler object is required to return a java.io.InputStream object with the raw bytes.

Returns:
a Java object with the content of this AttachmentPart object

Throws:
SOAPException - if there is no content set into this AttachmentPart object or if there was a data transformation error

abstract public java.io.InputStream getRawContent() throws SOAPException
InputStream AttachmentPart getContent
AttachmentPart content-type DataContentHandler

InputStream  Subsequent API  InputStream
#getRawContentBytes  API
return InputStream AttachmentPart
Throws SOAPException: AttachmentPart
since SAAJ 1.3
See also getRawContentBytes
getRawContent

public abstract InputStream getRawContent() throws SOAPException

Gets the content of this AttachmentPart object as an InputStream as if a call had been made to getContent and no DataContentHandler had been registered for the content-type of this AttachmentPart.

Note that reading from the returned InputStream would result in consuming the data in the stream. It is the responsibility of the caller to reset the InputStream appropriately before calling a subsequent API. If a copy of the raw attachment content is required then the getRawContentBytes() API should be used instead.

Returns:

an InputStream from which the raw data contained by the AttachmentPart can be accessed.

Throws:

SOAPException - if there is no content set into this AttachmentPart object or if there was a data transformation error.

Since:

SAAJ 1.3

See Also:

getRawContentBytes()

abstract public byte[] getRawContentBytes() throws SOAPException

byte[] AttachmentPart getContent
AttachmentPart content-type DataContentHandler
return AttachmentPart byte[]

Throws SOAPException: AttachmentPart

since SAAJ 1.3
getRawContentBytes

public abstract byte[] getRawContentBytes()
throws SOAPException

Gets the content of this AttachmentPart object as a byte[] array as if a call had been made to getContent and no DataContentHandler had been registered for the content-type of this AttachmentPart.

Returns:
   a byte[] array containing the raw data of the AttachmentPart.

Throws:
   SOAPException - if there is no content set into this AttachmentPart object or if there was a data transformation error.

Since:
   SAAJ 1.3

abstract public java.io.InputStream getBase64Content()
throws SOAPException

Base64

return InputStream Base64 AttachmentPart

Throws
   SOAPException: AttachmentPart

since
   SAAJ 1.3

getBase64Content

public abstract InputStream getBase64Content()
throws SOAPException

Returns an InputStream which can be used to obtain the content of AttachmentPart as Base64 encoded character data, this method would base64 encode the raw bytes of the attachment and return.

Returns:
   an InputStream from which the Base64 encoded AttachmentPart can be read.
abstract public void setContent(Object object, String contentType)

Sets the content of this attachment part to that of the given Object and sets the value of the Content-Type header to the given type. The type of the Object should correspond to the value given for the Content-Type. This depends on the particular set of DataContentHandler objects in use.

Parameters:
- object - the Java object that makes up the content for this attachment part
- contentType - the MIME string that specifies the type of the content

Throws:
- IllegalArgumentException - may be thrown if the contentType does not match the type of the content object, or if there was no DataContentHandler object for this content object
abstract public void setRawContent(InputStream content, String contentType) throws SOAPException

Sets the content of this attachment part to that contained by the InputStream content and sets the value of the Content-Type header to the value contained in contentType.

A subsequent call to getSize() may not be an exact measure of the content size.

Parameters:
- content - the raw data to add to the attachment part
- contentType - the value to set into the Content-Type header

Throws:
- SOAPException - if an there is an error in setting the content
- NullPointerException - if content is null

Since:
abstract public void setRawContentBytes(byte[] content, int offset, int len, String contentType) throws SOAPException

Sets the content of this attachment part to that contained by the byte[] array content and sets the value of the Content-Type header to the value contained in contentType.

Parameters:
- content - the raw data to add to the attachment part
- contentType - the value to set into the Content-Type header
- offset - the offset in the byte array of the content
- len - the number of bytes that form the content

Throws:
- SOAPException - if an there is an error in setting the content or content is null

Since:
- SAAJ 1.3
abstract public void setBase64Content(java.io.InputStream content, String contentType) throws SOAPException

Base64 InputStream Content-Type contentType base64

getSize()

content Base64
contentType Content-Type

Throws SOAPException:

Throws NullPointerException: content null

since SAAJ 1.3

setSize()

content Base64
contentType Content-Type

Sets the content of this attachment part from the Base64 source InputStream and sets the value of the Content-Type header to the value contained in contentType. This method would first decode the base64 input and write the resulting raw bytes to the attachment.

A subsequent call to getSize() may not be an exact measure of the content size.

Parameters:

content - the base64 encoded data to add to the attachment part
contentType - the value to set into the Content-Type header

Throws:

SOAPException - if an there is an error in setting the content

NullPointerException - if content is null

Since:

SAAJ 1.3
abstract public **DataHandler** getDataHandler() throws **SOAPException**

<table>
<thead>
<tr>
<th>AttachmentPart</th>
<th>DataHandler</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>AttachmentPart</td>
</tr>
<tr>
<td>Throws</td>
<td><strong>SOAPException</strong>: AttachmentPart</td>
</tr>
</tbody>
</table>

**getDataHandler**

public abstract **DataHandler** getDataHandler() throws **SOAPException**

Gets the DataHandler object for this AttachmentPart object.

**Returns:**
the DataHandler object associated with this AttachmentPart object

**Throws:**
**SOAPException** - if there is no data in this AttachmentPart object

---

abstract public void **setDataHandler**(**DataHandler** dataHandler)

<table>
<thead>
<tr>
<th>DataHandler</th>
<th>AttachmentPart</th>
</tr>
</thead>
<tbody>
<tr>
<td>setDataHandler</td>
<td>dataHandler</td>
</tr>
<tr>
<td>Throws</td>
<td><strong>IllegalArgumentException</strong>: DataHandler</td>
</tr>
</tbody>
</table>

**setDataHandler**

public abstract void **setDataHandler**(**DataHandler** dataHandler)

Sets the given DataHandler object as the data handler for this AttachmentPart object. Typically, on an incoming message, the data handler is automatically set. When a message is being created and populated with content, the **setDataHandler** method can be used to
get data from various data sources into the message.

Parameters:

dataHandler - the DataHandler object to be set

Throws:

IllegalArgumentException - if there was a problem with the specified DataHandler object

---

```java
public String getContentId()

"Content-ID" MIME

return "Content-ID" String null

See also

setContentId
```

getContentId

public String getContentId()

    Gets the value of the MIME header whose name is "Content-ID".

    Returns:

    a String giving the value of the "Content-ID" header or null if there is none

    See Also:

    `setContentId(java.lang.String)`

---

```java
public String getContentLocation()

"Content-Location" MIME

return "Content-Location" String null

getContentLocation

public String getContentLocation()

    Gets the value of the MIME header whose name is "Content-
```
Location".

Returns:
a String giving the value of the "Content-Location" header or
null if there is none

---

```java
public String getContentType()

"Content-Type" MIME

getContentType
```

```java
public String getContentType()

Gets the value of the MIME header whose name is "Content-Type".

Returns:
a String giving the value of the "Content-Type" header or null if
there is none
```

---

```java
public void setContentId(String contentId)

"Content-ID" MIME

setContentId
```

```java
public void setContentId(String contentId)

Sets the MIME header whose name is "Content-ID" with the given
value.

Parameters:
```
public void setContentLocation(String contentLocation)
"Content-Location" MIME

Sets the MIME header whose name is "Content-Location" with the given value.

Parameters:
contentLocation - a String giving the value of the "Content-Location" header

Throws:
IllegalArgumentException - if there was a problem with the specified content location

public void setContentType(String contentType)
"Content-Type" MIME

Sets the MIME header whose name is "Content-Type" with the given value.

Parameters:
contentType - a String giving the value of the "Content-Type" header

Throws:
IllegalArgumentException - if there was a problem with the specified content location
Sets the MIME header whose name is "Content-Type" with the given value.

**Parameters:**

- contentType - a String giving the value of the "Content-Type" header

**Throws:**

- IllegalArgumentException - if there was a problem with the specified content type

---

```java
abstract public void removeMimeHeader(String header)
```

**MIME**

- `header` - the string name of the MIME header/s to be removed

---

```java
public abstract void removeMimeHeader(String header)
```

Removes all MIME headers that match the given name.

**Parameters:**

- header - the string name of the MIME header/s to be removed

---

```java
abstract public void removeAllMimeHeaders()
```

**MIME**

---

```java
public abstract void removeAllMimeHeaders()
```

Removes all the MIME header entries.

---

```java
abstract public String[] getMimeHeader(String name)
```
public abstract String[] getMimeHeader(String name)

Gets all the values of the header identified by the given string.

Parameters:
name - the name of the header; example: "Content-Type"

Returns:
a String array giving the value for the specified header

See Also:
setMimeHeader(java.lang.String, java.lang.String)

abstract public void setMimeHeader(String name, String value)

Changes the first header entry that matches the given name to the given value, adding a new header if no existing header matches.
This method also removes all matching headers but the first.

Note that RFC822 headers can only contain US-ASCII characters.

**Parameters:**

- `name` - a `String` giving the name of the header for which to search
- `value` - a `String` giving the value to be set for the header whose name matches the given name

**Throws:**

- `IllegalArgumentException` - if there was a problem with the specified mime header name or value

---

```java
abstract public void addMimeHeader(String name, String value)
```

**MIME AttachmentPart**

**RFC822 US-ASCII**

```java
name  String
value String
```

**Throws** IllegalArgumentException: MIME

---

**addMimeHeader**

```java
public abstract void addMimeHeader(String name, String value)
```

Adds a MIME header with the specified name and value to this `AttachmentPart` object.

Note that RFC822 headers can contain only US-ASCII characters.

**Parameters:**

- `name` - a `String` giving the name of the header to be added
- `value` - a `String` giving the value of the header to be added
abstract public java.util.Iterator<E> getAllMimeHeaders()

MimeHeader AttachmentPart
return AttachmentPart MIME Iterator

getAllMimeHeaders

public abstract Iterator getAllMimeHeaders()

Retrieves all the headers for this AttachmentPart object as an iterator over the MimeHeader objects.

Returns:
   an Iterator object with all of the Mime headers for this AttachmentPart object

abstract public java.util.Iterator<E>
getMatchingMimeHeaders(String[] names)

MimeHeader
return MIME Iterator

getMatchingMimeHeaders

public abstract Iterator getMatchingMimeHeaders(String[] names)

Retrieves all MimeHeader objects that match a name in the given array.

Parameters:
   names - a String array with the name(s) of the MIME headers to

Throws:
   IllegalArgumentException - if there was a problem with the specified mime header name or value
Returns: all of the MIME headers that match one of the names in the given array as an Iterator object.

abstract public java.util.Iterator<E> getNonMatchingMimeHeaders(String[] names)

getNonMatchingMimeHeaders

public abstract Iterator getNonMatchingMimeHeaders(String[] names)

Retrieves all MimeHeader objects whose name does not match a name in the given array.

Parameters:
   names - a String array with the name(s) of the MIME headers not to be returned

Returns: all of the MIME headers in this AttachmentPart object except those that match one of the names in the given array. The nonmatching MIME headers are returned as an Iterator object.
PS:
public abstract class AttachmentUnmarshaller
extends Object

JAXB

API  JAXB 2.0  MIME  MTOM/XOP  WS-I AP 1.0
JAXB  MIME

XOP  #isXOPPackage()  id
getAttachment*(String cid) id cid

- XOP cid
- WS-I AP 1.0cid

See XML-binary Optimized Packaging, WS-I Attachments
also Profile Version 1.0, Describing Media Content of Binary Data in XML

since JAXB 2.0

This API enables an efficient cooperative processing of optimized binary data formats between a JAXB 2.0 implementation and MIME-based package processor (MTOM/XOP and WS-I AP 1.0). JAXB unmarshals the body of a package, delegating the understanding of the packaging format being used to a MIME-based package processor that implements this abstract class.

This abstract class identifies if a package requires XOP processing, isXOPPackage() and provides retrieval of binary content stored as attachments by content-id.
Identifying the content-id, cid, to pass to getAttachment*(String cid)

- For XOP processing, the infoset representation of the cid is described in step 2a in Section 3.2 Interpreting XOP Packages
- For WS-I AP 1.0, the cid is identified as an element or attribute of type ref:swaRef specified in Section 4.4 Referencing Attachments from the SOAP Envelope

Since:
JAXB 2.0

Author:
Marc Hadley, Kohsuke Kawaguchi, Joseph Fialli

See Also:
Unmarshaller.setAttachmentUnmarshaller, XML-binary Optimized Packaging, WS-I Attachments Profile Version 1.0, Describing Media Content of Binary Data in XML

---

## Constructor Summary

**AttachmentUnmarshaller()**

---

## Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAttachmentAsByteArray(String cid)</strong></td>
<td>abstract byte[] Retrieve the attachment identified by content-id, cid, as a byte[].</td>
</tr>
<tr>
<td><strong>getAttachmentAsDataHandler(String cid)</strong></td>
<td>abstract DataHandler Lookup MIME content by content-id, cid, and return as a DataHandler.</td>
</tr>
<tr>
<td><strong>isXOPPackage()</strong></td>
<td>boolean Read-only property that returns true if JAXB unmarshaller needs to perform XOP processing.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.\texttt{Object}:
\texttt{clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait}

\section*{Constructor Detail}

\begin{verbatim}
public AttachmentUnmarshaller()

AttachmentUnmarshaller

public AttachmentUnmarshaller()
\end{verbatim}

\section*{Method Detail}

abstract public \texttt{DataHandler} \texttt{getAttachmentAsDataHandler(String cid)}

\begin{tabular}{|c|c|c|}
\hline
id & cid & MIME & \texttt{DataHandler} \\
\hline
\end{tabular}

\begin{verbatim}
DataHandler.getContentType() instanceof DataHandler.getContent() \texttt{image/gif java.awt.Image} \\
image/jpeg \texttt{java.awt.Image} text/xml or application/xml \\
javax.xml.transform.Source
\end{verbatim}

\begin{tabular}{|c|c|c|}
\hline
\texttt{cid} & XML & \texttt{xs:anyURI} \\
\hline
\texttt{cid: URI URI} & \texttt{#isXOPPackage() ==true} & \texttt{RFC 2387} \\
\hline
\end{tabular}
getAttachmentAsDataHandler

public abstract DataHandler getAttachmentAsDataHandler(String cid)

Lookup MIME content by content-id, cid, and return as a DataHandler.

The returned DataHandler instance must be configured to meet the following required mapping constraint.

<table>
<thead>
<tr>
<th>MIME Type</th>
<th>Java Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataHandler.getContentType()</td>
<td>instanceof DataHandler.getContent()</td>
</tr>
<tr>
<td>image/gif</td>
<td>java.awt.Image</td>
</tr>
<tr>
<td>image/jpeg</td>
<td>java.awt.Image</td>
</tr>
<tr>
<td>text/xml or application/xml</td>
<td>javax.xml.transform.Source</td>
</tr>
</tbody>
</table>

Note that it is allowable to support additional mappings.

**Parameters:**

cid - It is expected to be a valid lexical form of the XML Schema xs:anyURI datatype. If isXOPPackage() == true, it must be a valid URI per the cid: URI scheme (see RFC 2387)

**Returns:**
a DataHandler that represents the MIME attachment.

**Throws:**

IllegalArgumentException - if the attachment for the given cid is not found.

abstract public byte[] getAttachmentAsByteArray(String cid)

byte[] id cid
getAttachmentAsByteArray

public abstract byte[] getAttachmentAsByteArray(String cid)

Retrieve the attachment identified by content-id, cid, as a byte[].

Parameters:
  cid - It is expected to be a valid lexical form of the XML Schema xs:anyURI datatype. If isXOPPackage() == true, it must be a valid URI per the cid: URI scheme (see RFC 2387)

Returns:
  byte[] representation of attachment identified by cid.

Throws:
  IllegalArgumentException - if the attachment for the given cid is not found.
Read-only property that returns true if JAXB unmarshaller needs to perform XOP processing.

This method returns true when the constraints specified in Identifying XOP Documents are met. This value must not change during the unmarshalling process.

**Returns:**
true when MIME context is a XOP Document.
public interface Attribute
extends XMLEvent

Implements: XMLEvent
Implemented by: Namespace

StartElement
  version 1.0
  See also javax.xml.stream.events.StartElement

An interface that contains information about an attribute. Attributes are reported as a set of events accessible from a StartElement. Other applications may report Attributes as first-order events, for example as the results of an XPath expression.

Version:
  1.0

Author:
  Copyright (c) 2003 by BEA Systems. All Rights Reserved.

See Also:
  javax.xml.stream.events.StartElement

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface</th>
</tr>
</thead>
</table>

javax.xml.stream.XMLStreamConstants

ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getDTDType()</td>
<td>Gets the type of this attribute, default is the String &quot;CDATA&quot;</td>
</tr>
<tr>
<td>QName getName()</td>
<td>Returns the QName for this attribute</td>
</tr>
<tr>
<td>String getValue()</td>
<td>Gets the normalized value of this attribute</td>
</tr>
<tr>
<td>boolean isSpecified()</td>
<td>A flag indicating whether this attribute was actually specified in the start-tag of its element, or was defaulted from the schema.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.stream.events.XMLEvent

asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public javax.xml.namespace.QName getName()

QName

getName
**QName getName()**

Returns the QName for this attribute

**public String getValue()**

**getValue**

**String getValue()**

Gets the normalized value of this attribute

**public String getDTDType()**

"CDATA"

```java
return "CDATA"
```

**getDTDType**

**String getDTDType()**

Gets the type of this attribute, default is the String "CDATA"

**Returns:**

the type as a String, default is "CDATA"

**public boolean isSpecified()**

```java
return true
```

**isSpecified**
boolean isSpecified()

A flag indicating whether this attribute was actually specified in the start-tag of its element, or was defaulted from the schema.

**Returns:**
returns true if this was specified in the start element
javax.persistence Annotation Type AttributeOverride

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface AttributeOverride

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

AttributeOverride Basic Id

AttributeOverride AttributeOverride

@MappedSuperclass
public class Employee {
    @Id protected Integer id;
    @Version protected Integer version;
    protected String address;
    public Integer getId() { ... }
    public void setId(Integer id) { ... }
    public String getAddress() { ... }
    public void setAddress(String address) { ... }
}

@Entity
@AttributeOverride(name="address", column=@Column(name="ADDR"))
public class PartTimeEmployee extends Employee {
    // address field mapping overridden to ADDR
    protected Float wage();
    public Float getHourlyWage() { ... }
    public void setHourlyWage(Float wage) { ... }
}

since Java Persistence 1.0

See javax.persistence.Embedded, javax.persistence.Embeddable, javax.persistence.MappedSuperclass

also
The `AttributeOverride` annotation is used to override the mapping of a Basic (whether explicit or default) property or field or Id property or field.

The `AttributeOverride` annotation may be applied to an entity that extends a mapped superclass or to an embedded field or property to override a basic mapping defined by the mapped superclass or embeddable class. If the `AttributeOverride` annotation is not specified, the column is mapped the same as in the original mapping.

Example:

```java
@MappedSuperclass
public class Employee {
    @Id protected Integer id;
    @Version protected Integer version;
    protected String address;
    public Integer getId() { ... }
    public void setId(Integer id) { ... }
    public String getAddress() { ... }
    public void setAddress(String address) { ... }
}

@Entity
@AttributeOverride(name="address", column=@Column(name="ADDR"))
public class PartTimeEmployee extends Employee {
    // address field mapping overridden to ADDR
    protected Float wage();
    public Float getHourlyWage() { ... }
    public void setHourlyWage(Float wage) { ... }
}
```

Since:
Java Persistence 1.0

See Also:
Embedded, Embeddable, MappedSuperclass

---

**Required Element Summary**
### Element Detail

**abstract public String name()**

**name**

```java
public abstract String name
```

(Required) The name of the property whose mapping is being overridden if property-based access is being used, or the name of the field if field-based access is being used.

---

**abstract public Column column()**

**column**

```java
public abstract Column column
```

(Required) The column that is being mapped to the persistent attribute. The mapping type will remain the same as is defined in the embeddable class or mapped superclass.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.persistence  
Annotation Type AttributeOverrides

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface AttributeOverrides

**Implements:** Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

@Embedded
@AttributeOverrides(
    @AttributeOverride(name="startDate", column=@Column("EMP_START")),
    @AttributeOverride(name="endDate", column=@Column("EMP_E")
}
public EmploymentPeriod getEmploymentPeriod() { ... }

**since** Java Persistence 1.0

Is used to override mappings of multiple properties or fields.

Example:
@Embedded
@AttributeOverrides(
    @AttributeOverride(name="startDate", column=@Column("EMP_START")),
    @AttributeOverride(name="endDate", column=@Column("EMP_E")
}
public EmploymentPeriod getEmploymentPeriod() { ... }

**Since:**
Java Persistence 1.0
Required Element Summary

<table>
<thead>
<tr>
<th>AttributeOverride[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One or more mapping override</td>
</tr>
</tbody>
</table>

Element Detail

abstract public **AttributeOverride[]** value()

value

public abstract **AttributeOverride[]** value

One or more mapping override

---

Overview  Package  Tree  Deprecated  Index  Help

PREV CLASS  NEXT CLASS
SUMMARY: REQUIRED | OPTIONAL
FRAMES  NO FRAMES
DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.webapp  Class AttributeTag

java.lang.Object
   ↓  javax.servlet.jsp.tagext.TagSupport
   ↓  javax.faces.webapp.AttributeTag

All Implemented Interfaces:
   Serializable, IterationTag, JspTag, Tag

Deprecated. The Faces implementation must now provide the implementation for this class.

Extends: TagSupport

String

deprecated  Faces

public class AttributeTag
extends TagSupport

Tag implementation that adds an attribute with a specified name and String value to the component whose tag it is nested inside, if the component does not already contain an attribute with the same name. This tag creates no output to the page currently being created.

See Also:
   Serialized Form

---

Field Summary

Fields inherited from class javax.servlet.jsp.tagext.TagSupport
id, pageContext
### Constructor Summary

**AttributeTag()**

*Deprecated.*

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>doEndTag()</strong></td>
<td><em>Deprecated.</em> Default processing of the end tag returning <code>EVAL_PAGE</code>.</td>
</tr>
<tr>
<td><strong>doStartTag()</strong></td>
<td><em>Deprecated.</em> Register the specified attribute name and value with the <code>UIComponent</code> instance associated with our most immediately surrounding <code>UIComponentTag</code> instance, if this <code>UIComponent</code> does not already have a value for the specified attribute name.</td>
</tr>
<tr>
<td><strong>release()</strong></td>
<td><em>Deprecated.</em> Release references to any acquired resources.</td>
</tr>
<tr>
<td><strong>setName(String name)</strong></td>
<td><em>Deprecated.</em> Set the attribute name.</td>
</tr>
<tr>
<td><strong>setValue(String value)</strong></td>
<td><em>Deprecated.</em> Set the attribute value.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.servlet.jsp.tagext.TagSupport`

- `doAfterBody`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`
Constructor Detail

public AttributeTag()

AttributeTag

public AttributeTag()

Deprecated.

Method Detail

public void setName(String name)

setName

public void setName(String name)

Deprecated.

Set the attribute name.

Parameters:

name - The new attribute name

public void setValue(String value)
setValue

```java
public void setValue(String value)
```

Deprecated.

Set the attribute value.

**Parameters:**

value - The new attribute value

---

doStartTag

```java
public int doStartTag() throws JspException
```

**UIComponentTag**  **UIComponent**  **UIComponent**

Throws

JspException: JSP

---

doStartTag

```java
public int doStartTag() throws JspException
```

Deprecated.

Register the specified attribute name and value with the **UIComponent** instance associated with our most immediately surrounding **UIComponentTag** instance, if this **UIComponent** does not already have a value for the specified attribute name.

**Specified by:**

doStartTag in interface Tag

**Overrides:**
**doStartTag** in class `TagSupport`

Returns: SKIP_BODY

Throws: `JspException` - if a JSP error occurs

See Also: `Tag.doStartTag()`

---

**public int doEndTag() throws `JspException`**

### doEndTag

**public int** `doEndTag()`

throws `JspException`

**Deprecated.**

**Description copied from class: `TagSupport`**

Default processing of the end tag returning EVAL_PAGE.

**Specified by:**

`doEndTag` in interface `Tag`

**Overrides:**

`doEndTag` in class `TagSupport`

Returns: EVAL_PAGE

**Throws:**

`JspException` - if an error occurs while processing this tag

See Also: `Tag.doEndTag()`

---

**public void release()**
public void release()

    Deprecated.

    Release references to any acquired resources.

    Specified by:
    release in interface Tag

    Overrides:
    release in class TagSupport

    See Also:
    Tag.release()
javax.xml.registry.infomodel  Interface AuditableEvent

All Superinterfaces: ExtensibleObject, RegistryObject

public interface AuditableEvent

extends RegistryObject

Implements: RegistryObject

AuditableEvent  RegistryObject AuditableEvent

RegistryObject RegistryObject CreateUpdate
Deprecate Delete RegistryObject AuditableEvent
AuditableEvent RegistryObject  Package Object
AuditableEvent

RegistryObject AuditableEvent Collection Object

See also  javax.xml.registry.infomodel.RegistryObject

AuditableEvent instances provide a long term record of events that effect a change of state in a RegistryObject. Such events are usually a result of a client initiated request. AuditableEvent instances are generated by the registry service to log such events.

Often such events effect a change in the life cycle of a RegistryObject. For example a client request could Create, Update, Deprecate or Delete a RegistryObject. No AuditableEvent is created for requests that do not alter the state of a RegistryObject. Specifically, read-only requests do not generate an AuditableEvent. No AuditableEvent is generated for a RegistryObject when it is classified, assigned to a Package or associated with another Object.

A RegistryObject is associated with an ordered Collection of
AuditableEvent instances that provide a complete audit trail for that Object.

**Author:**
Farrukh S. Najmi

**See Also:**
RegistryObject

### Field Summary

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_CREATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is created.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_DELETED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is deleted.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_DEPRECATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is deprecated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_UNDEPRECATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is undeprecated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_UPDATED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is updated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>EVENT_TYPE_VERSIONED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An event where a RegistryObject is versioned.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>int</th>
<th>getEventType()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the type of this event.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RegistryObject</th>
<th>getRegistryObject()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the RegistryObject associated with this AuditableEvent.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>getTimestamp()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the Timestamp for when this event occurred.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>User</th>
<th>getUser()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the User associated with this object.</td>
</tr>
</tbody>
</table>
Methods inherited from interface javax.xml.registry.infomodel.RegistryObject
addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternallink, addExternallinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternallink, removeExternallinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternallinks, setKey, setName, toXML

Methods inherited from interface javax.xml.registry.infomodel.ExtensibleObject
addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

Field Detail

EVENT_TYPE_CREATED
static final int EVENT_TYPE.Created

An event where a RegistryObject is created.

See Also: Constant Field Values

EVENT_TYPE_DELETED
static final int EVENT_TYPE_DELETED


An event where a RegistryObject is deleted.

See Also:
Constant Field Values

EVENT_TYPE_DEPRECATED

static final int EVENT_TYPE_DEPRECATED

An event where a RegistryObject is deprecated.

See Also:
Constant Field Values

EVENT_TYPE_UPDATED

static final int EVENT_TYPE_UPDATED

An event where a RegistryObject is updated.

See Also:
Constant Field Values

EVENT_TYPE_VERSIONED

static final int EVENT_TYPE_VERSIONED

An event where a RegistryObject is versioned.

See Also:
Constant Field Values
EVENT_TYPE_UNDEPRECATED

static final int EVENT_TYPE_UNDEPRECATED

An event where a RegistryObject is undeprecated.

See Also:
   Constant Field Values

 Method Detail

public User getUser() throws JAXRException

User

1

return AuditableEvent User null

Throws JAXRException: JAXR

label

requestor

supplierCardinality 1

directed

associates <{User}>
the User that sent the request that generated this this AuditableEvent. Must not be null

Throws:

`JAXRException` - If the JAXR provider encounters an internal error

```java
public java.sql.Timestamp getTimestamp() throws JAXRException
    Timestamp

1

    return
    Throws                      JAXRException: JAXR

getTimestamp

    Timestamp getTimestamp() throws JAXRException

    Gets the Timestamp for when this event occurred.

    Capability Level: 1

    Returns:
        the timestamp that records the time the event occurred
    Throws:
        JAXRException - If the JAXR provider encounters an internal error

public int getEventType() throws JAXRException

1
getEventType

int getEventType() throws JAXRException

Gets the type of this event.

Capability Level: 1

Returns: 
the type of this event

Throws:

JAXRException - If the JAXR provider encounters an internal error

See Also:
EVENT_TYPE_CREATED

getRegistryObject

public RegistryObject getRegistryObject() throws JAXRException

AuditableEvent RegistryObject

1

return RegistryObject

Throws JAXRException: JAXR

getRegistryObject

RegistryObject getRegistryObject() throws JAXRException
Gets the RegistryObject associated with this AuditableEvent.

**Capability Level: 1**

**Returns:**
the RegistryObject that was the focus of this event

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error
javax.mail Class AuthenticationFailedException

java.lang.Object
   ▼ java.lang.Throwable
      ◀ java.lang.Exception
         ◀ javax.mail.MessagingException
            ◀ javax.mail.AuthenticationFailedException

All Implemented Interfaces:
   Serializable

public class AuthenticationFailedException
extends MessagingException

Extends: Throwable > Exception > MessagingException

Store  Transport  connect

This exception is thrown when the connect method on a Store or Transport object fails due to an authentication failure (e.g., bad user name or password).

Author:
   Bill Shannon

See Also:
   Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthenticationFailedException() Constructor</td>
</tr>
<tr>
<td>AuthenticationFailedException(String message) Constructor</td>
</tr>
</tbody>
</table>


### Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class javax.mail.MessagingException</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCause, getNextException, setNextException, toString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

**public AuthenticationFailedException()**
Constructor

**AuthenticationFailedException**

**public AuthenticationFailedException()**
Constructor

**public AuthenticationFailedException(String message)**
Constructor

    message
public AuthenticationFailedException(String message)

Constructor

Parameters:
message - The detailed error message
Class Authenticator

public abstract class Authenticator
extends Object

getPasswordAuthentication (getRequestingXXX())

version 1.9, 05/04/07
java.net.Authenticator, getInstance(java.util.Properties, javax.mail.Authenticator),
getDefaultInstance(java.util.Properties, javax.mail.Authenticator),
requestPasswordAuthentication, javax.mail.PasswordAuthentication

The class Authenticator represents an object that knows how to obtain authentication for a network connection. Usually, it will do this by prompting the user for information.

Applications use this class by creating a subclass, and registering an instance of that subclass with the session when it is created. When authentication is required, the system will invoke a method on the subclass (like getPasswordAuthentication). The subclass's method can query about the authentication being requested with a number of inherited methods (getRequestingXXX()), and form an appropriate message for the user.
All methods that request authentication have a default implementation that fails.

**Version:**
1.9, 05/04/07

**Author:**
Bill Foote, Bill Shannon

**See Also:**

---

### Constructor Summary

**Authenticator()**

---

### Method Summary

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String</td>
<td>getDefaultUserName()</td>
<td></td>
</tr>
<tr>
<td>protected PasswordAuthentication</td>
<td>getPasswordAuthentication()</td>
<td>Called when password authentication is needed.</td>
</tr>
<tr>
<td>protected int</td>
<td>getRequestedPort()</td>
<td></td>
</tr>
<tr>
<td>protected String</td>
<td>getRequestedPrompt()</td>
<td></td>
</tr>
<tr>
<td>protected String</td>
<td>getRequestedProtocol()</td>
<td>Give the protocol that's requesting the connection.</td>
</tr>
<tr>
<td>protected InetAddress</td>
<td>getRequestedSite()</td>
<td></td>
</tr>
</tbody>
</table>
Constructor Detail

public Authenticator()

Authenticator

public Authenticator()

Method Detail

final protected java.net.InetAddress getRequestedSite()

    return InetAddress null

getRequestingSite

protected final InetAddress getRequestedSite()

    Returns:
    the InetAddress of the site requesting authorization, or null if it's not available.

final protected int getRequestedPort()

    return
protected final int getRequestingPort()

Returns:
the port for the requested connection

final protected String getRequestedProtocol()

returns

See also getProtocol

generatedProtocol

protected final String getRequestedProtocol()

Give the protocol that's requesting the connection. Often this will be based on a URLName.

Returns:
the protocol

See Also:
URLName.getProtocol()

generatedPrompt

final protected String getRequestedPrompt()

return

generatedUserName

protected final String getRequestedPrompt()

Returns:
the prompt string given by the requestor

final protected String getDefaultUserName()
getDefaultUserName

```java
protected final String getDefaultUserName()

Returns:
the default user name given by the requestor
```

getPasswordAuthentication

```java
protected PasswordAuthentication getPasswordAuthentication()
null

return PasswordAuthentication null
```

getPasswordAuthentication

```java
protected PasswordAuthentication getPasswordAuthentication()

Called when password authentication is needed. Subclasses should override the default implementation, which returns null.

Note that if this method uses a dialog to prompt the user for this information, the dialog needs to block until the user supplies the information. This method can not simply return after showing the dialog.

Returns:
The PasswordAuthentication collected from the user, or null if none is provided.
```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence  Annotation Type Basic

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Basic

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Basic  Basic Java
    String, java.math.BigInteger, java.math.BigDecimal, java.util.Date, java.util.Calendar,
    Byte[], char[], Character[]  Serializable

Basic  Java Persistence 1.0  en

The Basic annotation is the simplest type of mapping to a database column. The Basic annotation can be applied to a persistent property or instance variable of any of the following types: Java primitive types, wrappers of the primitive types, String, java.math.BigInteger, java.math.BigDecimal, java.util.Date, java.util.Calendar, java.sql.Date, java.sql.Time, java.sql.Timestamp, byte[], Byte[], char[], Character[], enums, and any other type that implements Serializable.

The use of the Basic annotation is optional for persistent fields and properties of these types.

Since:
    Java Persistence 1.0
### fetch

**Abstract public** `FetchType` **fetch()**

(Optional) Defines whether the value of the field or property should be lazily loaded or must be eagerly fetched. The **EAGER** strategy is a requirement on the persistence provider runtime that the value must be eagerly fetched. The **LAZY** strategy is a hint to the persistence provider runtime. If not specified, defaults to **EAGER**.

**Default:**

**EAGER**

### optional

**Abstract public** `boolean` **optional()**

**null**

(Optional) Defines whether the value of the field or property may be null. This is a hint and is disregarded for primitive types; it may be used in schema generation. If not specified, defaults to **true**.
Default:
   true

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.el Class BeanELResolver**

```java
public class BeanELResolver extends ELResolver
```

**Extends:** ELResolver  
**Inner classes:** BeanELResolver.BeanProperty, BeanELResolver.BeanProperties

JavaBean

```java
base base null base JavaBean
JavaBean
```

```java
#isReadOnly true#setValue PropertyNotWritableException
```

ELResolver CompositeELResolver ELResolver

**javadoc**

```java
since JSP 2.1
See also javax.el.CompositeELResolver, javax.el.ELResolver
```

Defines property resolution behavior on objects using the JavaBeans component architecture.

This resolver handles base objects of any type, as long as the base is not null. It accepts any object as a property, and coerces it to a string. That string is then used to find a JavaBeans compliant property on the base object. The value is accessed using JavaBeans getters and setters.
This resolver can be constructed in read-only mode, which means that `isReadOnly` will always return `true` and `setValue(javax.el.ELContext, javax.lang.Object, javax.lang.Object, javax.lang.Object)` will always throw `PropertyNotWritableException`.

ELResolvers are combined together using `CompositeELResolvers`, to define rich semantics for evaluating an expression. See the javadocs for `ELResolver` for details.

Because this resolver handles base objects of any type, it should be placed near the end of a composite resolver. Otherwise, it will claim to have resolved a property before any resolvers that come after it get a chance to test if they can do so as well.

**Since:**
- JSP 2.1

**See Also:**
- `CompositeELResolver`, `ELResolver`
BeanELResolver(boolean isReadOnly)

Creates a new BeanELResolver whose read-only status is determined by the given parameter.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCommonPropertyType</td>
<td>Class&lt;?&gt; getCommonPropertyType(ELContext context, Object base)</td>
<td>If the base object is not null, returns the most general type that this resolver accepts for the property argument.</td>
</tr>
<tr>
<td>getFeatureDescriptors</td>
<td>Iterator&lt;FeatureDescriptor&gt; getFeatureDescriptors(ELContext context, Object base)</td>
<td>If the base object is not null, returns an Iterator containing the set of JavaBeans properties available on the given object.</td>
</tr>
<tr>
<td>getType</td>
<td>Class&lt;?&gt; getType(ELContext context, Object base, Object property)</td>
<td>If the base object is not null, returns the most general acceptable type that can be set on this bean property.</td>
</tr>
<tr>
<td>getValue</td>
<td>Object getValue(ELContext context, Object base, Object property)</td>
<td>If the base object is not null, returns the current value of the given property on this bean.</td>
</tr>
<tr>
<td>isReadOnly</td>
<td>boolean isReadOnly(ELContext context, Object base, Object property)</td>
<td>If the base object is not null, returns whether a call to setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always fail.</td>
</tr>
<tr>
<td>setValue</td>
<td>void setValue(ELContext context, Object base, Object property, Object val)</td>
<td>If the base object is not null, attempts to set the value of the given property on this bean.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
close, equals, finalize, getClass, hashCode, notify, notifyAll,
Constructor Detail

public BeanELResolver()
/* BeanELResolver */

BeanELResolver

public BeanELResolver()

    Creates a new read/write BeanELResolver.

public BeanELResolver(boolean isReadOnly)

    BeanELResolver
        isReadOnly    Bean     true    false

BeanELResolver

public BeanELResolver(boolean isReadOnly)

    Creates a new BeanELResolver whose read-only status is determined by the given parameter.

    Parameters:
    isReadOnly - true if this resolver cannot modify beans; false otherwise.

Method Detail

public Class<T> getType(ELContext context, Object base, Object property)
**getType**

```java
public Class<?> getType(ELContext context,
                         Object base,
                         Object property)
```

If the base object is not `null`, returns the most general acceptable type that can be set on this bean property.

If the base is not `null`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

The provided property will first be coerced to a `String`. If there is a `BeanInfoProperty` for this property and there were no errors retrieving it, the `propertyType` of the `propertyDescriptor` is returned. Otherwise, a `PropertyNotFoundException` is thrown.

**Specified by:**

`getType` in class `ELResolver`
Parameters:
context - The context of this evaluation.
base - The bean to analyze.
property - The name of the property to analyze. Will be coerced to a String.

Returns:
If the propertyResolved property of ELContext was set to true, then the most general acceptable type; otherwise undefined.

Throws:
NullPointerException - if context is null
PropertyNotFoundException - if base is not null and the specified property does not exist or is not readable.
ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

public Object getValue(ELContext context, Object base, Object property)
    base null Bean
    base null ELContext propertyResolved true
    true

String JavaBean base
PropertyNotFoundException
getValue

```java
public Object getValue(ELContext context,
                         Object base,
                         Object property)
```

If the base object is not `null`, returns the current value of the given property on this bean.

If the base is not `null`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

The provided property name will first be coerced to a `String`. If the property is a readable property of the base object, as per the JavaBeans specification, then return the result of the getter call. If the getter throws an exception, it is propagated to the caller. If the property is not found or is not readable, a `PropertyNotFoundException` is thrown.

**Specified by:**
```
getValue in class ELResolver
```

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The bean on which to get the property.
- `property` - The name of the property to get. Will be coerced to a `String`.

**Returns:**
- If the `propertyResolved` property of `ELContext` was set to `true`, then the value of the given property. Otherwise, undefined.

**Throws:**
- `NullPointerException` - if `context` is `null`.
- `PropertyNotFoundException` - if `base` is not `null` and the specified property does not exist or is not readable.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.
public void setValue(ELContext context, Object base, Object property, Object val)

If the base object is not null, attempts to set the value of the given property on this bean.

If the base is not null, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller can safely assume no value was set.

If this resolver was constructed in read-only mode, this method will
always throw PropertyNotWritableException.

The provided property name will first be coerced to a String. If property is a writable property of base (as per the JavaBeans Specification), the setter method is called (passing value). If the property exists but does not have a setter, then a PropertyNotFoundException is thrown. If the property does not exist, a PropertyNotFoundException is thrown.

Specified by:
setValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - The bean on which to set the property.
property - The name of the property to set. Will be coerced to a String.
val - The value to be associated with the specified key.

Throws:
NullPointerException - if context is null.
PropertyNotFoundException - if base is not null and the specified property does not exist.
PropertyNotWritableException - if this resolver was constructed in read-only mode, or if there is no setter for the property.
ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.
**PropertyNotFoundException**

```
context base Bean
property String
return ELContext propertyResolved true setValue true false
```

**Throws**

- NullPointerException: context null
- PropertyNotFoundException: base null
- ELException: cause

**isReadOnly**

```java
public boolean isReadOnly(ELContext context,
                          Object base,
                          Object property)
```

If the base object is not null, returns whether a call to `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` will always fail.

If the base is not null, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller can safely assume no value was set.

If this resolver was constructed in read-only mode, this method will always return `true`.

The provided property name will first be coerced to a `String`. If property is a writable property of `base`, `false` is returned. If the property is found but is not writable, `true` is returned. If the property is not found, a `PropertyNotFoundException` is thrown.

**Specified by:**

- `isReadOnly` in class `ELResolver`

**Parameters:**

- `context` - The context of this evaluation.
base - The bean to analyze.
property - The name of the property to analyzed. Will be coerced to a String.

Returns:
If the propertyResolved property of ELContext was set to true, then true if calling the setValue method will always fail or false if it is possible that such a call may succeed; otherwise undefined.

Throws:
 NullPointerException - if context is null
 PropertyNotFoundException - if base is not null and the specified property does not exist.
 ELEexception - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```
public java.util.Iterator<E> getFeatureDescriptors(ELContext context, Object base)
base null JavaBean Iterator null

Iterator 0 java.beans.FeatureDescriptor info
bean BeanInfo.getPropertyDescriptor PropertyDescriptor FeatureDescriptor "type"
"resolvableAtDesignTime"

• ELEResolver#TYPE -
PropertyDescriptor.getPropertyType()
• ELEResolver#RESOLVABLE_AT_DESIGN_TIME - true

context base return 0 FeatureDescriptor Iterator Bean null base
null null
```
getFeatureDescriptors

public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext context, Object base)

If the base object is not null, returns an Iterator containing the set of JavaBeans properties available on the given object. Otherwise, returns null.

The Iterator returned must contain zero or more instances of FeatureDescriptor. Each info object contains information about a property in the bean, as obtained by calling the BeanInfo.getPropertyDescriptors method. The FeatureDescriptor is initialized using the same fields as are present in the PropertyDescriptor, with the additional required named attributes "type" and "resolvableAtDesignTime" set as follows:

- ELResolver.TYPE - The runtime type of the property, from PropertyDescriptor.getPropertyType().
- ELResolver.RESOLVABLE_AT_DESIGN_TIME - true.

Specified by:
getFeatureDescriptors in class ELResolver

Parameters:
context - The context of this evaluation.
base - The bean to analyze.

Returns:
An Iterator containing zero or more FeatureDescriptor objects, each representing a property on this bean, or null if the base object is null.

See Also:
FeatureDescriptor

---

public Class<T> getCommonPropertyType(ELContext context, Object base)

base null property null

base null Object.class
getCommonPropertyType

public Class<? extends Object> getCommonPropertyType(ELContext context, Object base)

If the base object is not null, returns the most general type that this resolver accepts for the property argument. Otherwise, returns null.

Assuming the base is not null, this method will always return Object.class. This is because any object is accepted as a key and is coerced into a string.

**Specified by:**
getCommonPropertyType in class ELResolver

**Parameters:**
- context - The context of this evaluation.
- base - The bean to analyze.

**Returns:**
- null if base is null; otherwise Object.class.
**javax.el**  
**Class** BeanELResolver.BeanProperties

**java.lang.Object**  
  \_ **javax.el.BeanELResolver.BeanProperties**

**Enclosing class:**  
BeanELResolver

---

protected static final class BeanELResolver.BeanProperties  
extends Object

**Contained within:**  
BeanELResolver

---

### Constructor Summary

| BeanELResolver.BeanProperties(\texttt{Class}\langle?\rangle \texttt{baseClass}) |

---

### Method Summary

| BeanELResolver.BeanProperty | getBeanProperty(\texttt{String} \texttt{property}) |

---

Methods inherited from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

---

### Constructor Detail
public BeanELResolver.BeanProperties(Class<?> baseClass)

Method Detail

public BeanELResolver.BeanProperty getBeanProperty(String property)

g.getBeanProperty

public BeanELResolver.BeanProperty getBeanProperty(String property)

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.el Class BeanELResolver.BeanProperty

java.lang.Object
   → javax.el.BeanELResolver.BeanProperty

Enclosing class:
   BeanELResolver

protected static final class BeanELResolver.BeanProperty
extends Object

Contained within: BeanELResolver

Constructor Summary

| BeanELResolver.BeanProperty(Class<?>) baseClass, PropertyDescriptor descriptor |

Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>getPropertyType()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>getReadMethod()</td>
</tr>
<tr>
<td>Method</td>
<td>getWriteMethod()</td>
</tr>
<tr>
<td>boolean</td>
<td>isReadOnly()</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
**Constructor Detail**

BeanELResolver.BeanProperty

```java
public BeanELResolver.BeanProperty(Class<? extends T> baseClass, PropertyDescriptor descriptor)
```

**Method Detail**

```java
public Class<T> getGenericType()
```

**getGenericType**

```java
public Class getGenericType()
```

```java
public boolean isReadOnly()
```

**isReadOnly**

```java
public boolean isReadOnly()
```

```java
public Method getReadMethod()
```

**getReadMethod**

```java
public Method getReadMethod()
```
public Method getWriteMethod()

getWriteMethod

public Method getWriteMethod()
javax.enterprise.deploy.spi.exceptions Class
BeanNotFoundException

java.lang.Object
 ^  java.lang.Throwable
   ^  java.lang.Exception
      ^  javax.enterprise.deploy.spi.exceptions.BeanNotFoundException

All Implemented Interfaces:
Serializable

public class BeanNotFoundException
extends Exception

Extends: Throwable > Exception

Bean Bean Bean

This exception is to report that the bean is not a child of the parent bean.

See Also:
Serialized Form

Constructor Summary

BeanNotFoundException(String s)

Creates an new BeanNotFoundException object.

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public BeanNotFoundException(String s)

BeanNotFoundException

public BeanNotFoundException(String s)

Creates an new BeanNotFoundException object.

Parameters:

s - a string indicating what was wrong with the target.
javax.faces.convert Class BigDecimalConverter

java.lang.Object    ^    javax.faces.convert.BigDecimalConverter

All Implemented Interfaces:
    Converter

public class BigDecimalConverter
extends Object
implements Converter

Implements: Converter

java.math.BigDecimal Converter

Converter implementation for java.math.BigDecimal values.

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>CONVERTER_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The standard converter id for this converter.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>DECIMAL_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the FacesMessage to be created if the conversion to BigDecimal fails.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>STRING_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the FacesMessage to be created if the conversion of the BigDecimal value to String fails.</td>
</tr>
</tbody>
</table>

Constructor Summary
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the <strong>Apply Request Values</strong> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the <strong>Render Response</strong> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
DECIMAL_ID

public static final String DECIMAL_ID

The message identifier of the FacesMessage to be created if the conversion to BigDecimal fails. The message format string for this message may optionally include the following placeholders:

- {0} replaced by the unconverted value.
- {1} replaced by an example value.
- {2} replaced by a String whose value is the label of the input component that produced this message.

See Also:
 Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the BigDecimal value to String fails. The message format string for this message may optionally include the following placeholders:

- {0} replaced by the unconverted value.
- {1} replaced by a String whose value is the label of the input component that produced this message.

See Also:
 Constant Field Values

| Constructor Detail |
public BigDecimalConverter()

BigDecimalConverter

public BigDecimalConverter()

### Method Detail

**public Object getAsObject(FacesContext context, UICOMPONENT component, String value)**

**Throws** ConverterException: NullPointerException

Throws NullPointerException: NullPointerException

context

component null

---

getAsObject

**public** **Object** **getAsObject**(FacesContext context, UICOMPONENT component, String value)**

**Description copied from interface: Converter**

Convert the specified string value, which is associated with the specified UICOMPONENT, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Specified by:** getAsObject in interface Converter

**Parameters:**

context - FacesContext for the request being processed

component - UICOMPONENT with which this model object value is associated

value - String value to be converted (may be null)

**Returns:**

null if the value to convert is null, otherwise the result of the
conversion

Throws:
  ConverterException - if conversion cannot be successfully performed
  NullPointerException - if context or component is null

public String getAsString(FacesContext context, UIComponent component, Object value)

Throws  ConverterException: NullPointerException
Throws  NullPointerException: NullPointerException

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
  getAsString in interface Converter

Parameters:
  context - FacesContext for the request being processed
  component - UIComponent with which this model object value is associated
  value - Model object value to be converted (may be null)

Returns:
  a zero-length String if value is null, otherwise the result of the conversion

Throws:
  ConverterException - if conversion cannot be successfully
NullPointerException - if context or component is null

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
public final class BigDecimalHolder

extends Object
implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigDecimal</td>
</tr>
<tr>
<td>value</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigDecimalHolder()</td>
</tr>
<tr>
<td>BigDecimalHolder(BigDecimal myBigDecimal)</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>
value

public BigDecimal value

Constructor Detail

public BigDecimalHolder()

BigDecimalHolder

public BigDecimalHolder()

public BigDecimalHolder(java.math.BigDecimal myBigDecimal)

BigDecimalHolder

public BigDecimalHolder(BigDecimal myBigDecimal)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.faces.convert Class BigIntegerConverter

java.lang.Object
   ▼ javax.faces.convert.BigIntegerConverter

All Implemented Interfaces:
   Converter

public class BigIntegerConverter
   extends Object
   implements Converter

Implements: Converter

java.math.BigInteger Converter

Converter implementation for java.math.BigInteger values.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String BIGINTEGER_ID</td>
</tr>
<tr>
<td>The message identifier of the FacesMessage to be created if the conversion to BigInteger fails.</td>
</tr>
<tr>
<td>static String CONVERTER_ID</td>
</tr>
<tr>
<td>The standard converter id for this converter.</td>
</tr>
<tr>
<td>static String STRING_ID</td>
</tr>
<tr>
<td>The message identifier of the FacesMessage to be created if the conversion of the BigInteger Value to String fails.</td>
</tr>
</tbody>
</table>

| Constructor Summary |
**BigIntegerConverter()**

# Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>getAsObject</strong>(FacesContext context, UIComponent component, String value)</td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td></td>
<td><strong>getAsString</strong>(FacesContext context, UIComponent component, Object value)</td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

# Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

# Field Detail

**CONVERTER_ID**

public static final String CONVERTER_ID

The standard converter id for this converter.

See Also:
Constant Field Values
BIGINTEGER_ID

public static final String BIGINTEGER_ID

The message identifier of the FacesMessage to be created if the conversion to BigInteger fails. The message format string for this message may optionally include the following placeholders:

- \{0\} replaced by the unconverted value.
- \{1\} replaced by an example value.
- \{2\} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the BigInteger value to String fails. The message format string for this message may optionally include the following placeholders:

- \{0\} replaced by the unconverted value.
- \{1\} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail
public BigIntegerConverter()

BigIntegerConverter

public BigIntegerConverter()

### Method Detail

**public Object getAsObject(FacesContext context, UIComponent component, String value)**

Throws ConverterException: NullPointerException

Throws NullpointerException: NullPointerException

getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

**Description copied from interface:** Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Specified by:**

getAsObject in interface Converter

**Parameters:**

- context - FacesContext for the request being processed
- component - UIComponent with which this model object value is associated
- value - String value to be converted (may be null)

**Returns:**

null if the value to convert is null, otherwise the result of the
Throws:  
ConverterException - if conversion cannot be successfully performed  
NullPointerException - if context or component is null

public String getAsString(FacesContext context, UIComponent component, Object value)

Throws  
ConverterException: NullPointerException

Throws  
NullPointerException: NullPointerException

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:  
getAsString in interface Converter

Parameters:  
context - FacesContext for the request being processed  
component - UIComponent with which this model object value is associated  
value - Model object value to be converted (may be null)

Returns:  
a zero-length String if value is null, otherwise the result of the conversion

Throws:  
ConverterException - if conversion cannot be successfully performed
performed

NullPointerException - if context or component is null
javax.xml.rpc.holders  **Class BigIntegerHolder**

**java.lang.Object**  
  └─javax.xml.rpc.holders.BigIntegerHolder

**All Implemented Interfaces:**  
  Holder

```java
public final class BigIntegerHolder  
extends Object  
implements Holder

Implements: Holder
```

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BigInteger</strong> value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BigIntegerHolder()</strong></td>
</tr>
<tr>
<td><strong>BigIntegerHolder(BigInteger myBigInteger)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| Methods inherited from class java.lang.Object  
  clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |
Field Detail

value

public BigInteger value

Constructor Detail

public BigIntegerHolder()

BigIntegerHolder

public BigIntegerHolder()

public BigIntegerHolder(java.math.BigInteger myBigInteger)

BigIntegerHolder

public BigIntegerHolder(BigInteger myBigInteger)
PS:
javax.xml.bind **Class Binder<XmlNode>**

**java.lang.Object**

```
javax.xml.bind.Binder<XmlNode>
```

---

public abstract class **Binder<XmlNode>** extends **Object**

**XML** **XML** **JAXB**

**XML** **XML** **JAXB**

`#getJAXBNode(Object)`

JAXB Binder  Binder
`#updateJAXB(Object)`

- **XML** **XML**
- **#unmarshal(Object)**  **XML** **JAXB**  **XML**  **JAXB**

- **XML / JAXB**
- **#updateXML(Object)**  **JAXB**  **XML**  **XML**  **PI**

Binder  `createBinder()`  `createBinder(Class)`

```
XmlNode  XML /Binder  org.w3c.dom.Node.class
XmlNode  Binder  XML
```

**since**  **JAXB 2.0**

Enable synchronization between XML infoset nodes and JAXB objects representing same XML document.

An instance of this class maintains the association between XML nodes
of an infoset preserving view and a JAXB representation of an XML document. Navigation between the two views is provided by the methods
`getXMLNode(Object)` and `getJAXBNode(Object)`.

Modifications can be made to either the infoset preserving view or the JAXB representation of the document while the other view remains unmodified. The binder is able to synchronize the changes made in the modified view back into the other view using the appropriate Binder update methods, `updateXML(Object, Object)` or `updateJAXB(Object)`.

A typical usage scenario is the following:

- load XML document into an XML infoset representation
- `unmarshal(Object)` XML infoset view to JAXB view. (Note to conserve resources, it is possible to only unmarshal a subtree of the XML infoset view to the JAXB view.)
- application access/updates JAXB view of XML document.
- `updateXML(Object)` synchronizes modifications to JAXB view back into the XML infoset view. Update operation preserves as much of original XML infoset as possible (i.e. comments, PI, ...)

A Binder instance is created using the factory method
`JAXBContext.createBinder()` or `JAXBContext.createBinder(Class)`.

The template parameter, `XmlNode`, is the root interface/class for the XML infoset preserving representation. A Binder implementation is required to minimally support an `XmlNode` value of `org.w3c.dom.Node.class`. A Binder implementation can support alternative XML infoset preserving representations.

**Since:**
JAXB 2.0

**Author:**
Kohsuke Kawaguchi (kohsuke.kawaguchi@sun.com) Joseph Fialli

---

**Constructor Summary**

`Binder()`
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getEventHandler()</code></td>
<td>Return the current event handler or the default event handler if one hasn't been set.</td>
</tr>
<tr>
<td><code>getJAXBNode(XmlNode xmlNode)</code></td>
<td>Gets the JAXB object associated with the given XML element.</td>
</tr>
<tr>
<td><code>getProperty(String name)</code></td>
<td>Get the particular property in the underlying implementation of Binder.</td>
</tr>
<tr>
<td><code>getSchema()</code></td>
<td>Gets the last Schema object (including null) set by the setSchema(Schema) method.</td>
</tr>
<tr>
<td><code>getXMLNode(Object jaxbObject)</code></td>
<td>Gets the XML element associated with the given JAXB object.</td>
</tr>
<tr>
<td><code>marshal(Object jaxbObject, XmlNode xmlNode)</code></td>
<td>Marshal a JAXB object tree to a new XML document.</td>
</tr>
<tr>
<td><code>setEventHandler(ValidationEventHandler handler)</code></td>
<td>Allow an application to register a ValidationEventHandler.</td>
</tr>
<tr>
<td><code>setProperty(String name, Object value)</code></td>
<td>Set the particular property in the underlying implementation of Binder.</td>
</tr>
<tr>
<td><code>setSchema(Schema schema)</code></td>
<td>Specifies whether marshal, unmarshal and update methods performs validation on their XML content.</td>
</tr>
<tr>
<td><code>unmarshal(XmlNode xmlNode)</code></td>
<td>Unmarshal XML infoset view to a JAXB object tree.</td>
</tr>
</tbody>
</table>
Unmarshal XML root element by provided declaredType to a JAXB object tree.

```java
unmarshal(XmlNode xmlNode, Class<T> declaredType)
```

Unmarshal XML infoset view to a JAXB object tree.

This method is similar to `Unmarshaller.unmarshal(Node)` with the
addition of maintaining the association between XML nodes and the produced JAXB objects, enabling future update operations, updateXML(Object, Object) or updateJAXB(Object).

When getSchema() is non-null, XmlNode and its descendants is validated during this operation.

This method throws UnmarshalException when the Binder's JAXBContext does not have a mapping for the XML element name or the type, specifiable via @xsi:type, of XmlNode to a JAXB mapped class. The method unmarshal(Object, Class) enables an application to specify the JAXB mapped class that the XmlNode should be mapped to.

**Parameters:**
XmlNode - the document/element to unmarshal XML data from.

**Returns:**
the newly created root object of the JAXB object tree.

**Throws:**
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Binder is unable to perform the XML to Java binding.
IllegalArgumentException - If the node parameter is null

```java
public abstract <T> JAXBElement<T> unmarshal(XmlNode xmlNode, Class<T> declaredType)
throws JAXBException
```

Unmarshal XML root element by provided declaredType to a JAXB object tree.

Implements Unmarshal by Declared Type
This method is similar to `Unmarshaller.unmarshal(Node, Class)` with the addition of maintaining the association between XML nodes and the produced JAXB objects, enabling future update operations, `updateXML(Object, Object)` or `updateJAXB(Object)`.

When `getSchema()` is non-null, `XmlNode` and its descendants is validated during this operation.

**Parameters:**
- `XmlNode` - the document/element to unmarshal XML data from.
- `declaredType` - appropriate JAXB mapped class to hold node's XML data.

**Returns:**
- `JAXB Element` representation of `node`

**Throws:**
- `JAXBException` - If any unexpected errors occur while unmarshalling
- `UnmarshallerException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the Binder is unable to perform the XML to Java binding.
- `IllegalArgumentException` - If any of the input parameters are null

**Since:**
JAXB2.0

---

`marshal`

```java
public abstract void marshal(Object jaxbObject,
                              XmlNode xmlNode)
    throws JAXBException
```

Marshal a JAXB object tree to a new XML document.

This method is similar to `Marshaller.marshal(Object, Node)` with the addition of maintaining the association between JAXB objects and the produced XML nodes, enabling future update operations such as `updateXML(Object, Object)` or `updateJAXB(Object)`. 
When `getSchema()` is non-null, the marshalled xml content is validated during this operation.

**Parameters:**
- `jaxbObject` - The content tree to be marshalled.
- `xmlNode` - The parameter must be a Node that accepts children.

**Throws:**
- `JAXBException` - If any unexpected problem occurs during the marshalling.
- `MarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Binder` is unable to marshal `jaxbObject` (or any object reachable from `jaxbObject`).
- `IllegalArgumentException` - If any of the method parameters are null

```java
abstract public Object getXMLNode(Object jaxbObject)
```

**abstract public Object getXMLNode(Object jaxbObject)**

**JAXB XML**

**XML JAXB bind update XML**

**JAXB JAXB XML**

```java
jaxbObject JAXB bind update
return JAXB Binder XML null
Throws IllegalArgumentException: jaxbObject null
```

**getXMLNode**

```java
public abstract XmlNode getXMLNode(Object jaxbObject)
```

**getXMLNode**

Gets the XML element associated with the given JAXB object.

Once a JAXB object tree is associated with an XML fragment, this method enables navigation between the two trees.
An association between an XML element and a JAXB object is established by the bind methods and the update methods. Note that this association is partial; not all XML elements have associated JAXB objects, and not all JAXB objects have associated XML elements.

**Parameters:**

jaxbObject - An instance that is reachable from a prior call to a bind or update method that returned a JAXB object tree.

**Returns:**

null if the specified JAXB object is not known to this Binder, or if it is not associated with an XML element.

**Throws:**

IllegalArgumentException - If the jaxbObject parameter is null

---

getJAXBNode

```java
public abstract Object getJAXBNode(XmlNode xmlNode)
```

Gets the JAXB object associated with the given XML element.

Once a JAXB object tree is associated with an XML fragment, this method enables navigation between the two trees.

An association between an XML element and a JAXB object is established by the unmarshal, marshal and update methods. Note that this association is partial; not all XML elements have associated JAXB objects, and not all JAXB objects have associated XML elements.

**Returns:**

null if the specified XML node is not known to this Binder, or if it is not associated with a JAXB object.

**Throws:**

IllegalArgumentException - If the node parameter is null
abstract public Object updateXML(Object jaxbObject)
throws JAXBException
JAXB XML

updateXML( jaxbObject, getXMLNode(jaxbObject));

Throws JAXBException: XML
Throws IllegalArgumentException: jaxbObject null

updateXML

public abstract XmlNode updateXML(Object jaxbObject)
throws JAXBException

Takes an JAXB object and updates its associated XML node and its descendants.

This is a convenience method of:

updateXML( jaxbObject, getXMLNode(jaxbObject));

Throws:
 JAXBException - If any unexpected problem occurs updating corresponding XML content.
IllegalArgumentException - If the jaxbObject parameter is null

updateXML

public abstract XmlNode updateXML(Object jaxbObject, XmlNode xmlNode)
throws JAXBException
Changes in JAXB object tree are updated in its associated XML parse tree.

This operation can be thought of as an "in-place" marshalling. The difference is that instead of creating a whole new XML tree, this operation updates an existing tree while trying to preserve the XML as much as possible.

For example, unknown elements/attributes in XML that were not bound to JAXB will be left untouched (whereas a marshalling operation would create a new tree that doesn't contain any of those.)

As a side-effect, this operation updates the association between XML nodes and JAXB objects.

**Parameters:**
- `jaxbObject` - root of potentially modified JAXB object tree
- `xmlNode` - root of update target XML parse tree

**Returns:**
Returns the updated XML node. Typically, this is the same node you passed in as `xmlNode`, but it maybe a different object, for example when the tag name of the object has changed.

**Throws:**
- `JAXBException` - If any unexpected problem occurs updating corresponding XML content.
- `IllegalArgumentException` - If any of the input parameters are null

updateJAXB

```java
public abstract Object updateJAXB(XmlNode xmlNode)
throws JAXBException
```

Takes an XML node and updates its associated JAXB object and its descendants.

This operation can be thought of as an "in-place" unmarshalling. The
difference is that instead of creating a whole new JAXB tree, this operation updates an existing tree, reusing as much JAXB objects as possible.

As a side-effect, this operation updates the association between XML nodes and JAXB objects.

**Returns:**
Returns the updated JAXB object. Typically, this is the same object that was returned from earlier `marshal(Object, Object)` or `updateJAXB(Object)` method invocation, but it maybe a different object, for example when the name of the XML element has changed.

**Throws:**
- `JAXBException` - If any unexpected problem occurs updating corresponding JAXB mapped content.
- `IllegalArgumentException` - If node parameter is null

### abstract public void setSchema(javax.xml.validation.Schema schema)

**marshalunmarshal update XML**

`schema` `null`

**See also**
`setSchema(Schema)`

### setSchema

**public abstract void** `setSchema(Schema schema)`

Specifies whether marshal, unmarshal and update methods performs validation on their XML content.

**Parameters:**
- `schema` - set to null to disable validation.

**See Also:**
`Unmarshaller.setSchema(Schema)`
abstract public javax.xml.validation.Schema getSchema()
    
    #setSchema(Schema)     Schema null
    return     Schema null

getSchema

public abstract Schema getSchema()

    Gets the last Schema object (including null) set by the
    setSchema(Schema) method.

    Returns:
    the Schema object for validation or null if not present

abstract public void setEventHandler(ValidationEventHandler handler) throws JAXBException

    Binder unmarshalmarshal update JAXB
    ValidationEventHandler

null Binder
    handler
    Throws JAXBException:

setEventHandler

public abstract void setEventHandler(ValidationEventHandler handler) throws JAXBException

    Allow an application to register a ValidationEventHandler.
The `ValidationEventHandler` will be called by the JAXB Provider if any validation errors are encountered during calls to any of the Binder unmarshal, marshal and update methods.

Calling this method with a null parameter will cause the Binder to revert back to the default default event handler.

**Parameters:**
- handler - the validation event handler

**Throws:**
- `JAXBException` - if an error was encountered while setting the event handler

---

**abstract public** `ValidationEventHandler` `getEventHandler()`
**throws** `JAXBException`

```java
return ValidationEventHandler
Throws JAXBException:
```

**getEventHandler**

**public abstract** `ValidationEventHandler` `getEventHandler()`
**throws** `JAXBException`

Return the current event handler or the default event handler if one hasn't been set.

**Returns:**
- the current `ValidationEventHandler` or the default event handler if it hasn't been set

**Throws:**
- `JAXBException` - if an error was encountered while getting the current event handler

---

**abstract public void** `setProperty(String name, Object value)`
unmarshal marshal PropertyException

Unmarshal Marshal

name value

Throws PropertyException:
Throws IllegalArgumentException: name null

setProperty

public abstract void setProperty(String name, Object value)
throws PropertyException

Set the particular property in the underlying implementation of Binder. This method can only be used to set one of the standard JAXB defined unmarshal/marshal properties or a provider specific property for binder, unmarshal or marshal. Attempting to set an undefined property will result in a PropertyException being thrown. See Supported Unmarshal Properties and Supported Marshal Properties.

Parameters:

name - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
value - the value of the property to be set

Throws:

PropertyException - when there is an error processing the given property or value
IllegalArgumentException - If the name parameter is null

abstract public Object getProperty(String name) throws PropertyException
getProperty

public abstract Object getProperty(String name)
  throws PropertyException

Get the particular property in the underlying implementation of
Binder. This method can only be used to get one of the standard
JAXB defined unmarshal/marshal properties or a provider specific
property for binder, unmarshal or marshal. Attempting to get an
undefined property will result in a PropertyException being thrown.
See Supported Unmarshal Properties and Supported Marshal
Properties.

Parameters:
  name - the name of the property to retrieve

Returns:
  the value of the requested property

Throws:
  PropertyException - when there is an error retrieving the given
  property or value property name
  IllegalArgumentException - If the name parameter is null
PS:
public interface Binding

Implemented by: HTTPBinding, SOAPBinding

Binding JAX-WS

since JAX-WS 2.0

The Binding interface is the base interface for JAX-WS protocol bindings.

Since:
JAX-WS 2.0

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getHandlerChain()</td>
<td>Gets a copy of the handler chain for a protocol binding instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setHandlerChain(List&lt;Handler&gt; chain)</td>
<td>Sets the handler chain for the protocol binding instance.</td>
</tr>
</tbody>
</table>

Method Detail

public java.util.List<E> getHandlerChain()

return java.util.List Handler
**getHandlerChain**

```java
List<Handler> getHandlerChain()
```

Gets a copy of the handler chain for a protocol binding instance. If the returned chain is modified a call to `setHandlerChain` is required to configure the binding instance with the new chain.

**Returns:**

java.util.List Handler chain

---

**setHandlerChain**

```java
void setHandlerChain(List<Handler> chain)
```

Sets the handler chain for the protocol binding instance.

**Parameters:**

- `chain` - A List of handler configuration entries

**Throws:**

- `WebServiceException` - On an error in the configuration of the handler chain
- `UnsupportedOperationException` - If this operation is not supported. This may be done to avoid any overriding of a pre-configured handler chain.
PS:
javax.xml.ws Interface BindingProvider

All Known Subinterfaces:
   Dispatch<T>

public interface BindingProvider

Implemented by: Dispatch

BindingProvider since JAX-WS 2.0
See also javax.xml.ws.Binding

The BindingProvider interface provides access to the protocol binding and associated context objects for request and response message processing.

Since: JAX-WS 2.0
See Also: Binding

Field Summary

| static String | ENDPOINT_ADDRESS_PROPERTY | Standard property: Target service endpoint address. |
| static String | PASSWORDPROPERTY | Standard property: Password for authentication. |
| static String | SESSION_MAINTAIN_PROPERTY | Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint. |
| static String | SOAPACTION_URI_PROPERTY | Standard property for SOAPAction. |
static String SOAPACTION_USE_PROPERTY
    Standard property for SOAPAction.

static String USERNAME_PROPERTY
    Standard property: User name for authentication.

## Method Summary

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binding</td>
<td>getBinding()</td>
<td>Get the Binding for this binding provider.</td>
</tr>
<tr>
<td>Map&lt;String, Object&gt;</td>
<td>getRequestContext()</td>
<td>Get the context that is used to initialize the message context for request messages.</td>
</tr>
<tr>
<td>Map&lt;String, Object&gt;</td>
<td>getResponseContext()</td>
<td>Get the context that resulted from processing a response message.</td>
</tr>
</tbody>
</table>

## Field Detail

**USERNAME_PROPERTY**

static final String USERNAME_PROPERTY

    Standard property: User name for authentication.

    Type: java.lang.String

    **See Also:**

    [Constant Field Values](#)
static final String PASSWORD_PROPERTY

Standard property: Password for authentication.

Type: java.lang.String

See Also:
Constant Field Values

ENDPOINT_ADDRESS_PROPERTY

static final String ENDPOINT_ADDRESS_PROPERTY

Standard property: Target service endpoint address. The URI scheme for the endpoint address specification must correspond to the protocol/transport binding for the binding in use.

Type: java.lang.String

See Also:
Constant Field Values

SESSION_MAINTAIN_PROPERTY

static final String SESSION_MAINTAIN_PROPERTY

Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint. If this property is set to true, the service client indicates that it wants the session to be maintained. If set to false, the session is not maintained. The default value for this property is false.

Type: java.lang.Boolean
SOAPACTION_USEPROPERTY

static final String SOAPACTION_USEPROPERTY

Standard property for SOAPAction. This boolean property indicates whether or not SOAPAction is to be used. The default value of this property is false indicating that the SOAPAction is not used.

Type: java.lang.Boolean

See Also:
Constant Field Values

SOAPACTION_URIPROPERTY

static final String SOAPACTION_URIPROPERTY

Standard property for SOAPAction. Indicates the SOAPAction URI if the javax.xml.ws.soap.http.soapaction.use property is set to true.

Type: java.lang.String

See Also:
Constant Field Values

Method Detail

public java.util.Map<K, V> getRequestContext()
getRequestContext

Map<String, Object> getRequestContext()

Get the context that is used to initialize the message context for request messages. Modifications to the request context do not affect the message context of either synchronous or asynchronous operations that have already been started.

Returns:
The context that is used in processing request messages.

public java.util.Map<K, V> getResponseContext()

Get the context that resulted from processing a response message. The returned context is for the most recently completed synchronous operation. Subsequent synchronous operation invocations overwrite the response context. Asynchronous operations return their response context via the Response interface.

Returns:
The context that resulted from processing the latest response messages.

public Binding getBinding()

Binding
getBinding

Binding getBinding()

Get the Binding for this binding provider.

**Returns:**

The Binding for this binding provider.
javax.xml.ws  Annotation Type BindingType

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented
public @interface BindingType

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented

BindingType  Web

**since**  JAX-WS 2.0

The BindingType annotation is used to specify the binding to use for a web service endpoint implementation class.

This annotation may be overridden programmatically or via deployment descriptors, depending on the platform in use.

**Since:**  
JAX-WS 2.0

---

**Optional Element Summary**

<table>
<thead>
<tr>
<th>String</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A binding identifier (a URI).</td>
</tr>
</tbody>
</table>

abstract public String value()  
(URI) SOAP 1.1 / HTTP
value

public abstract String value

A binding identifier (a URI). If not specified, the default is the SOAP 1.1 / HTTP binding.

See the SOAPBinding and HTTPBinding for the definition of the standard binding identifiers.

See Also:
- Binding, SOAPBinding.SOAP11HTTP_BINDING,
- SOAPBinding.SOAP12HTTP_BINDING, HTTPBinding.HTTP_BINDING

Default:

""
javax.servlet.jsp.tagext Class BodyContent

java.lang.Object
  └─ java.io.writer
      └─ javax.servlet.jsp.jspwriter
          └─ javax.servlet.jsp.tagext.bodycontent

All Implemented Interfaces:
   Closeable, Flushable, Appendable

public abstract class BodyContent
  extends JspWriter

Extends: java.io.writer > JspWriter

BodyContent  JspWriter

BodyContent

BodyContent  String

BodyContent  BodyContent  autoFlush  BodyContent
flush

BodyContent  PageContext  pushBody  popBody
BodyContent  JspWriter  BodyContent

  setBodyContent()  BodyContent  BodyTag  doEndTag()

An encapsulation of the evaluation of the body of an action so it is available to a tag handler. BodyContent is a subclass of JspWriter.

Note that the content of BodyContent is the result of evaluation, so it will not contain actions and the like, but the result of their invocation.

BodyContent has methods to convert its contents into a String, to read its
contents, and to clear the contents.

The buffer size of a BodyContent object is unbounded. A BodyContent object cannot be in autoFlush mode. It is not possible to invoke flush on a BodyContent object, as there is no backing stream.

Instances of BodyContent are created by invoking the pushBody and popBody methods of the PageContext class. A BodyContent is enclosed within another JspWriter (maybe another BodyContent object) following the structure of their associated actions.

A BodyContent is made available to a BodyTag through a setBodyContent() call. The tag handler can use the object until after the call to doEndTag().

---

**Field Summary**

Fields inherited from class javax.servlet.jsp.JspWriter

- autoFlush, bufferSize, DEFAULT_BUFFER, NO_BUFFER, UNBOUNDED_BUFFER

Fields inherited from class java.io.Writer

- lock

**Constructor Summary**

protected `BodyContent(JspWriter e)`

Protected constructor.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clearBody()</code></td>
<td>Clear the body without throwing any exceptions.</td>
</tr>
<tr>
<td><code>flush()</code></td>
<td>Redefined flush() so it is not legal.</td>
</tr>
<tr>
<td><code>getEnclosingWriter()</code></td>
<td></td>
</tr>
<tr>
<td><strong>JspWriter</strong></td>
<td>Get the enclosing JspWriter.</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>abstract Reader</td>
<td><strong>getReader()</strong></td>
</tr>
<tr>
<td>abstract String</td>
<td><strong>getString()</strong></td>
</tr>
<tr>
<td>abstract void</td>
<td><strong>writeOut(Writer out)</strong></td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.servlet.jsp.JspWriter**
- clear, clearBuffer, close, getBufferSize, getRemaining, isAutoFlush, newLine, print, print, print, print, print, print, print, print, print, println, println, println, println, println, println

**Methods inherited from class java.io.Writer**
- append, append, append, write, write, write, write, write

**Methods inherited from class java.lang.Object**
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

#### protected BodyContent(**JspWriter** e)

**e** JspWriter

**BodyContent**

#### protected BodyContent(**JspWriter** e)

Protected constructor. Unbounded buffer, no autoflushing.
Parameters:
  e - the enclosing JspWriter

Method Detail

public void flush() throws java.io.IOException

flush()

BodyContent

Throws java.io.IOException:

flush

public void flush() throws IOException

Redefined flush() so it is not legal.

It is not valid to flush a BodyContent because there is no backing stream behind it.

Specified by:
  flush in interface Flushable

Specified by:
  flush in class JspWriter

Throws:
  IllegalArgumentException - always thrown

public void clearBody()

clearBody
public void clearBody()

    Clear the body without throwing any exceptions.

abstract public java.io.Reader getReader()
Reader  BodyContent
    return  Reader  BodyContent

getReader

public abstract Reader getReader()

    Return the value of this BodyContent as a Reader.

    Returns:  
    the value of this BodyContent as a Reader

abstract public String getString()
String  BodyContent
    return  String  BodyContent

getString

public abstract String getString()

    Return the value of the BodyContent as a String.

    Returns:  
    the value of the BodyContent as a String

abstract public void writeOut(java.io.Writer out) throws java.io.IOException
BodyContent  Writer
writeOut

public abstract void writeOut(Writer out)
   throws IOException

Write the contents of this BodyContent into a Writer. Subclasses may optimize common invocation patterns.

Parameters:
   out - The writer into which to place the contents of this body evaluation

Throws:
   IOException - if an I/O error occurred while writing the contents of this BodyContent to the given Writer

public JspWriter getEnclosingWriter()
JspWriter

   return JspWriter

getEnclosingWriter

public JspWriter getEnclosingWriter()

   Get the enclosing JspWriter.

   Returns:
   the enclosing JspWriter passed at construction time
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail  **Class BodyPart**

```java
java.lang.Object
   \_ javax.mail.BodyPart
```

**All Implemented Interfaces:**
- Part

**Direct Known Subclasses:**
- MimeBodyPart

```java
public abstract class BodyPart
   extends Object
   implements Part

  Implements: Part
  Extended by: MimeBodyPart
  
  Multipart Part
  
  BodyPart Part ""
```

This class models a Part that is contained within a Multipart. This is an abstract class. Subclasses provide actual implementations.

BodyPart implements the Part interface. Thus, it contains a set of attributes and a "content".

**Author:**
  John Mani, Bill Shannon

---

**Field Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected</td>
<td>parent</td>
</tr>
</tbody>
</table>
The Multipart object containing this BodyPart, if known.

Fields inherited from interface javax.mail.Part
ATTACHMENT, INLINE

Constructor Summary

BodyPart()

Method Summary

Multipart
getParent() Return the containing Multipart object, or null if not known.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.mail.Part
addHeader, getAllHeaders, getContent, getContentType, getDataHandler, getDescription, getDisposition, getFileName, getHeader, getInputStream, getLineCount, getMatchingHeaders, getNonMatchingHeaders, getSize, isMimeType, removeHeader, setContent, setContentType, setDataHandler, setDescription, setDisposition, setFileName, setHeader, setText, writeTo

Field Detail

parent

protected Multipart parent
The Multipart object containing this BodyPart, if known.

Since:
JavaMail 1.1

Constructor Detail

public BodyPart()
PS:
public interface BodyTag

extends IterationTag

Implements: IterationTag
Implemented by: BodyTagSupport, UIComponentClassicTagBase

BodyTag IterationTag

bodyContent.getString String
bodyContent.writeOut JspWriter

BodyTag IterationTag doStartTag SKIP_BODY
EVAL_BODY_INCLUDE EVAL_BODY_BUFFERED

EVAL_BODY_INCLUDE IterationTag

EVAL_BODY_BUFFERED BodyContent JSP JSP
pageContext pushBody BodyContent out
PageContext popBody out

setter action

bodyContent BodyContent JSP setter
(setBodyContent) doStartTag() EVAL_BODY_BUFFERED
The BodyTag interface extends IterationTag by defining additional methods that let a tag handler manipulate the content of evaluating its body.

It is the responsibility of the tag handler to manipulate the body content. For example the tag handler may take the body content, convert it into a String using the bodyContent.getString method and then use it. Or the
A tag handler may take the body content and write it out into its enclosing JspWriter using the bodyContent.writeOut method.

A tag handler that implements BodyTag is treated as one that implements IterationTag, except that the doStartTag method can return SKIP_BODY, EVAL_BODY_INCLUDE or EVAL_BODY_BUFFERED.

If EVAL_BODY_INCLUDE is returned, then evaluation happens as in IterationTag.

If EVAL_BODY_BUFFERED is returned, then a BodyContent object will be created (by code generated by the JSP compiler) to capture the body evaluation. The code generated by the JSP compiler obtains the BodyContent object by calling the pushBody method of the current pageContext, which additionally has the effect of saving the previous out value. The page compiler returns this object by calling the popBody method of the PageContext class; the call also restores the value of out.

The interface provides one new property with a setter method and one new action method.

**Properties**

There is a new property: bodyContent, to contain the BodyContent object, where the JSP Page implementation object will place the evaluation (and reevaluation, if appropriate) of the body. The setter method (setBodyContent) will only be invoked if doStartTag() returns EVAL_BODY_BUFFERED and the corresponding action element does not have an empty body.

**Methods**

In addition to the setter method for the bodyContent property, there is a new action method: doInitBody(), which is invoked right after setBodyContent() and before the body evaluation. This method is only invoked if doStartTag() returns EVAL_BODY_BUFFERED.

**Lifecycle**

Lifecycle details are described by the transition diagram below.
Exceptions that are thrown during the computation of doStartTag(), setBodyContent(), doInitBody(), BODY, doAfterBody() interrupt the execution sequence and are propagated up the stack, unless the tag handler implements the TryCatchFinally interface; see that interface for details.

Empty and Non-Empty Action

If the TagLibraryDescriptor file indicates that the action must always have an empty element body, by an <body-content> entry of "empty", then the doStartTag() method must return SKIP_BODY. Otherwise, the doStartTag() method may return SKIP_BODY, EVAL_BODY_INCLUDE, or EVAL_BODY_BUFFERED.

Note that which methods are invoked after the doStartTag() depends on both the return value and on if the custom action element is empty or not in the JSP page, not how it's declared in the TLD.

If SKIP_BODY is returned the body is not evaluated, and doEndTag() is invoked.

If EVAL_BODY_INCLUDE is returned, and the custom action element is not empty, setBodyContent() is not invoked, doInitBody() is not invoked, the body is evaluated and "passed through" to the current out, doAfterBody() is invoked and then, after zero or more iterations, doEndTag() is invoked. If the custom action element is empty, only doStart() and doEndTag() are invoked.

If EVAL_BODY_BUFFERED is returned, and the custom action element is not empty, setBodyContent() is invoked, doInitBody() is invoked, the body is evaluated, doAfterBody() is invoked, and then, after zero or more iterations, doEndTag() is invoked. If the custom action element is empty, only doStart() and doEndTag() are invoked.

Field Summary
**EVAL_BODY_BUFFERED**
Request the creation of new buffer, a BodyContent on which to evaluate the body of this tag.

**EVAL_BODY_TAG**
*Deprecated.* As of Java JSP API 1.2, use BodyTag.EVAL_BODY_BUFFERED or IterationTag.EVAL_BODY_AGAIN.

**Fields inherited from interface** javax.servlet.jsp.tagext.IterationTag

- EVAL_BODY_AGAIN

**Fields inherited from interface** java.servlet.jsp.tagext.Tag

- EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

**Method Summary**

- **void** doInitBody()
  Prepare for evaluation of the body.

- **void** setBodyContent(BodyContent b)
  Set the bodyContent property.

**Methods inherited from interface** javax.servlet.jsp.tagext.IterationTag

- doAfterBody

**Methods inherited from interface** java.servlet.jsp.tagext.Tag

- doEndTag, doStartTag, getParent, release, setPageContext, setParent

**Field Detail**

**EVAL_BODY_TAG**
static final int EVAL_BODY_TAG

**Deprecated.** As of Java JSP API 1.2, use BodyTag.EVAL_BODY_BUFFERED or IterationTag.EVAL_BODY_AGAIN.
Deprecation constant that has the same value as EVAL_BODY_BUFFERED and EVAL_BODY_AGAIN. This name has been marked as deprecated to encourage the use of the two different terms, which are much more descriptive.

**See Also:**
[Constant Field Values](#)

---

EVAL_BODY_BUFFERED

static final int EVAL_BODY_BUFFERED

Request the creation of new buffer, a BodyContent on which to evaluate the body of this tag. Returned from doStartTag when it implements BodyTag. This is an illegal return value for doStartTag when the class does not implement BodyTag.

**See Also:**
[Constant Field Values](#)

---

**Method Detail**

def setBodyContent(BodyContent b)
def doInitBody()
def doStartTag() SKIP_BODY EVAL_BODY_INCLUDE

def setBodyContent out pageContext BodyContent
setBodyContent

```java
void setBodyContent(BodyContent b)
```

Set the bodyContent property. This method is invoked by the JSP page implementation object at most once per action invocation. This method will be invoked before doInitBody. This method will not be invoked for empty tags or for non-empty tags whose doStartTag() method returns SKIP_BODY or EVAL_BODY_INCLUDE.

When setBodyContent is invoked, the value of the implicit object out has already been changed in the pageContext object. The BodyContent object passed will have not data on it but may have been reused (and cleared) from some previous invocation.

The BodyContent object is available and with the appropriate content until after the invocation of the doEndTag method, at which case it may be reused.

**Parameters:**
- `b` - the BodyContent

**See Also:**
- `doInitBody()`, `IterationTag.doAfterBody()`
doInitBody

void doInitBody() throws JspException

Prepare for evaluation of the body. This method is invoked by the JSP page implementation object after setBodyContent and before the first time the body is to be evaluated. This method will not be invoked for empty tags or for non-empty tags whose doStartTag() method returns SKIP_BODY or EVAL_BODY_INCLUDE.

The JSP container will resynchronize the values of any AT_BEGIN and NESTED variables (defined by the associated TagExtraInfo or TLD) after the invocation of doInitBody().

**Throws:**

-JspException - if an error occurred while processing this tag

**See Also:**
IterationTag.doAfterBody()
javax.servlet.jsp.tagext  **Class BodyTagSupport**

**java.lang.Object**
  - javax.servlet.jsp.tagext.TagSupport
  - javax.servlet.jsp.tagext_BODY_TAG_SUPPORT

**All Implemented Interfaces:**
  - *Serializable*, *BodyTag*, *IterationTag*, *JspTag*, *Tag*

---

```java
public class BodyTagSupport
    extends TagSupport
    implements BodyTag

Extends: TagSupport
Implements: BodyTag

BodyTag

BodyTagSupport BodyTag bodyContent getter out
JspWriter

BodyTagSupport

---
```

A base class for defining tag handlers implementing BodyTag.

The BodyTagSupport class implements the BodyTag interface and adds additional convenience methods including getter methods for the bodyContent property and methods to get at the previous out JspWriter.

Many tag handlers will extend BodyTagSupport and only redefine a few methods.

**See Also:**
  - Serialized Form
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bodyContent</code></td>
<td>The current BodyContent for this BodyTag.</td>
</tr>
</tbody>
</table>

### Fields inherited from class `javax.servlet.jsp.tagext.TagSupport`

- `id`
- `pageContext`

### Fields inherited from interface `javax.servlet.jsp.tagext.BodyTag`

- `EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

### Fields inherited from interface `javax.servlet.jsp.tagext.IterationTag`

- `EVAL_BODY_AGAIN`

### Fields inherited from interface `javax.servlet.jsp.tagext.Tag`

- `EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

### Constructor Summary

#### `BodyTagSupport()`
Default constructor, all subclasses are required to only define a public constructor with the same signature, and to call the superclass constructor.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>doAfterBody()</code></td>
<td>After the body evaluation: do not reevaluate and continue with the page.</td>
</tr>
<tr>
<td><code>doEndTag()</code></td>
<td>Default processing of the end tag returning EVAL_PAGE.</td>
</tr>
<tr>
<td><code>doInitBody()</code></td>
<td>Prepare for evaluation of the body just before the first body evaluation: no action.</td>
</tr>
</tbody>
</table>
```java
int doStartTag()
    Default processing of the start tag returning EVAL_BODY_BUFFERED.

BodyContent getBodyContent()
    Get current bodyContent.

JspWriter getPreviousOut()
    Get surrounding out JspWriter.

void release()
    Release state.

void setBodyContent(BodyContent b)
    Prepare for evaluation of the body: stash the bodyContent away.

Methods inherited from class javax.servlet.jsp.tagext.TagSupport
findAncestorWithClass, getId, getParent, getValue, getValues, removeValue, setId, setPageContext, setParent, setValue

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Methods inherited from interface javax.servlet.jsp.tagext.Tag
getParent, setPageContext, setParent

Field Detail

bodyContent

protected BodyContent bodyContent
    The current BodyContent for this BodyTag.
```
**Constructor Detail**

public BodyTagSupport()

**BodyTagSupport**

public BodyTagSupport()

Default constructor, all subclasses are required to only define a public constructor with the same signature, and to call the superclass constructor. This constructor is called by the code generated by the JSP translator.

**Method Detail**

public int doStartTag() throws JspException
EPAL_BODY_BUFFERED

return EVAL_BODY_BUFFERED

Throws JspException:

See also doStartTag

doStartTag

public int doStartTag()

throws JspException

Default processing of the start tag returning EVAL_BODY_BUFFERED.

**Specified by:**

doStartTag in interface Tag

**Overrides:**
doStartTag in class TagSupport
Returns:
    EVAL_BODY_BUFFERED

Throws:
    JspException - if an error occurred while processing this tag

See Also:
    Tag.doStartTag()

public int doEndTag() throws JspException

EVAL_PAGE

return EVAL_PAGE

Throws JspException:

See also doEndTag

doEndTag

public int doEndTag() throws JspException

Default processing of the end tag returning EVAL_PAGE.

Specified by:
    doEndTag in interface Tag

Overrides:
    doEndTag in class TagSupport

Returns:
    EVAL_PAGE

Throws:
    JspException - if an error occurred while processing this tag

See Also:
    Tag.doEndTag()

public void setBodyContent(BodyContent b)

bodyContent

See also doAfterBody, doInitBody(), setBodyContent
**setBodyContent**

```java
public void setBodyContent(BodyContent b)
```

Prepare for evaluation of the body: stash the bodyContent away.

**Specified by:**
`setBodyContent` in interface `BodyTag`

**Parameters:**
- `b` - the BodyContent

**See Also:**
- `doAfterBody()`, `doInitBody()`, `BodyTag.setBodyContent(javax.servlet.jsp.tagext.BodyContent)`

---

**public void doInitBody() throws JspException**

**Throws**
- `JspException`

**See also**
- `setBodyContent`, `doAfterBody`, `doInitBody`

---

**doInitBody**

```java
public void doInitBody()
```

```java
throws JspException
```

Prepare for evaluation of the body just before the first body evaluation: no action.

**Specified by:**
`doInitBody` in interface `BodyTag`

**Throws:**
- `JspException` - if an error occurred while processing this tag

**See Also:**
- `setBodyContent(javax.servlet.jsp.tagext.BodyContent)`, `doAfterBody()`, `BodyTag.doInitBody()`
public int doAfterBody() throws JspException
bodyContent
return SKIP_BODY
Throws JspException:
doInitBody, doAfterBody
See also

---

doAfterBody

public int doAfterBody()
throws JspException

After the body evaluation: do not reevaluate and continue with the page. By default nothing is done with the bodyContent data (if any).

Specified by:
doAfterBody in interface IterationTag
Overrides:
doAfterBody in class TagSupport
Returns:
SKIP_BODY
Throws:
JspException - if an error occurred while processing this tag
See Also:
doInitBody(), IterationTag.doAfterBody()

---

public void release()

See also

---

release

public void release()

Release state.
Specified by: release in interface Tag
Overrides: release in class TagSupport
See Also: Tag.release()

public BodyContent getBodyContent()
bodyContent
    return

getBodyContent

public BodyContent getBodyContent()

    Get current bodyContent.

    Returns:
    the body content.

getPreviousOut

public JspWriter getPreviousOut()
out JspWriter
    return JspWriter bodyContent

getPreviousOut

public JspWriter getPreviousOut()

    Get surrounding out JspWriter.

    Returns:
    the enclosing JspWriter, from the bodyContent.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail.search Class BodyTerm

java.lang.Object
  └─javax.mail.search.SearchTerm
    └─javax.mail.search.StringTerm
     └─javax.mail.search.BodyTerm

All Implemented Interfaces:
  Serializable

public final class BodyTerm
extends StringTerm

Extends: searchTerm > StringTerm

Message Body MIME "text/*"

This class implements searches on a Message Body. All parts of the message that are of MIME type "text/*" are searched.

Author:
  Bill Shannon, John Mani

See Also:
  Serialized Form

Field Summary

| Fields inherited from class javax.mail.search.StringTerm |
| ignoreCase, pattern |

Constructor Summary

BodyTerm(String pattern)
## Constructor Detail

### public BodyTerm(String pattern)

Constructor

`pattern`  
String

### BodyTerm

public **BodyTerm**(String pattern)

Constructor

**Parameters:**

`pattern`  
The String to search for

## Method Detail

### Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th>equals(Object obj)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Equality comparison.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th>match(Message msg)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The match method.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.mail.search.StringTerm

getIgnoreCase, getPattern, hashCode, match

### Methods inherited from class java.lang.Object

clonen, finalize, getClass, notify, notifyAll, toString, wait, wait
public boolean match(Message msg)

    msg            Message
    return true    false

type

public boolean match(Message msg)

    The match method.

    Specified by:
    match in class SearchTerm

    Parameters:
    msg - The pattern search is applied on this Message's body

    Returns:
    true if the pattern is found; otherwise false

public boolean equals(Object obj)

equals

equality comparison.

    Overrides:
    equals in class StringTerm
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.convert Class BooleanConverter

java.lang.Object
   ^ javax.faces.convert.BooleanConverter

All Implemented Interfaces:
    Converter

public class BooleanConverter
extends Object
implements Converter

Implements: Converter

java.lang.Boolean boolean Converter

Converter implementation for java.lang.Boolean (and boolean primitive) values.

### Field Summary

<table>
<thead>
<tr>
<th>static String BOOLEAN_ID</th>
<th>The message identifier of the FacesMessage to be created if the conversion to Boolean fails.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String CONVERTER_ID</td>
<td>The standard converter id for this converter.</td>
</tr>
<tr>
<td>static String STRING_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion of the Boolean value to String fails.</td>
</tr>
</tbody>
</table>

### Constructor Summary
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAsObject()</code></td>
<td><code>Object getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified <code>UIComponent</code>, into a model data object that is appropriate for being stored during the <em>Apply Request Values</em> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>getAsString()</code></td>
<td><code>String getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified <code>UIComponent</code>, into a String that is suitable for being included in the response generated during the <em>Render Response</em> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

## Field Detail

**CONVERTER_ID**

public static final `String CONVERTER_ID`

The standard converter id for this converter.

**See Also:**

*Constant Field Values*
**BOOLEAN_ID**

public static final String BOOLEAN_ID

The message identifier of the FacesMessage to be created if the conversion to Boolean fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:

Constant Field Values

---

**STRING_ID**

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Boolean value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:

Constant Field Values

---

### Constructor Detail

public BooleanConverter()
BooleanConverter

class BooleanConverter {
    public BooleanConverter()

    Method Detail

    public Object getAsObject(FacesContext context, UIComponent component, String value)

    Description copied from interface: Converter

    Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

    Specified by:

        getAsObject in interface Converter

    Parameters:

        context - FacesContext for the request being processed
        component - UIComponent with which this model object value is associated
        value - String value to be converted (may be null)

    Returns:

        null if the value to convert is null, otherwise the result of the conversion
Throws:
- `ConverterException` - if conversion cannot be successfully performed
- `NullPointerException` - if context or component is null

```java
public String getAsString(FacesContext context, UIComponent component, Object value)
```

Throws `ConverterException`: `NullPointerException`

Throws `NullPointerException`

getAsString

```java
public String getAsString(FacesContext context, UIComponent component, Object value)
```

Description copied from interface: `Converter`

Convert the specified model object value, which is associated with the specified `UIComponent`, into a String that is suitable for being included in the response generated during the `Render Response` phase of the request processing lifecycle.

Specified by:
- `getAsString` in interface `Converter`

Parameters:
- context - `FacesContext` for the request being processed
- component - `UIComponent` with which this model object value is associated
- value - Model object value to be converted (may be `null`)

Returns:
- a zero-length String if value is `null`, otherwise the result of the conversion

Throws:
- `ConverterException` - if conversion cannot be successfully performed
NullPointerException - if context or component is null
javax.xml.rpc.holders  **Class BooleanHolder**

**java.lang.Object**
  └─javax.xml.rpc.holders.BooleanHolder

**All Implemented Interfaces:**
  - **Holder**

```java
public final class BooleanHolder
  extends Object
  implements Holder
```

**Implements:** **Holder**

---

**Field Summary**

<table>
<thead>
<tr>
<th>boolean</th>
<th>value</th>
</tr>
</thead>
</table>

**Constructor Summary**

- **BooleanHolder()**
- **BooleanHolder**(boolean myboolean)

**Method Summary**

**Methods inherited from class java.lang.Object:**
  - `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`
value

public boolean value

Constructor Detail

public BooleanHolder()

BooleanHolder

public BooleanHolder()

public BooleanHolder(boolean myboolean)

BooleanHolder

public BooleanHolder(boolean myboolean)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.rpc.holders  Class BooleanWrapperHolder

java.lang.Object
   └─javax.xml.rpc.holders.BooleanWrapperHolder

All Implemented Interfaces:
   Holder

public final class BooleanWrapperHolder
extends Object
implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>field</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>BooleanWrapperHolder()</td>
</tr>
<tr>
<td>BooleanWrapperHolder(Boolean myboolean)</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>
value

public Boolean value

Constructor Detail

public BooleanWrapperHolder()

BooleanWrapperHolder

public BooleanWrapperHolder()

public BooleanWrapperHolder(Boolean myboolean)

BooleanWrapperHolder

public BooleanWrapperHolder(Boolean myboolean)
PS:
public interface BootstrapContext

version 1.0

This provides a mechanism to pass a bootstrap context to a resource adapter instance when it is bootstrapped. That is, when (start(BootstrapContext)) method on the ResourceAdapter class is invoked. The bootstrap context contains references to useful facilities that could be used by the resource adapter instance.

Version:
1.0
Author:
Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Timer</strong></td>
</tr>
<tr>
<td>createTimer()</td>
</tr>
<tr>
<td>Creates a new java.util.Timer instance.</td>
</tr>
<tr>
<td><strong>WorkManager</strong></td>
</tr>
<tr>
<td>getWorkManager()</td>
</tr>
<tr>
<td>Provides a handle to a WorkManager instance.</td>
</tr>
<tr>
<td><strong>XATerminator</strong></td>
</tr>
<tr>
<td>getXATerminator()</td>
</tr>
<tr>
<td>Provides a handle to a XATerminator instance.</td>
</tr>
</tbody>
</table>

Method Detail

public WorkManager getWorkManager()
getWorkManager

`WorkManager getWorkManager()`

Provides a handle to a `WorkManager` instance. The `WorkManager` instance could be used by a resource adapter to do its work by submitting `Work` instances for execution.

Returns:

- a `WorkManager` instance.

getXATerminator

`XATerminator getXATerminator()`

Provides a handle to a `XATerminator` instance. The `XATerminator` instance could be used by a resource adapter to flow-in transaction completion and crash recovery calls from an EIS.

Returns:

- a `XATerminator` instance.

public `java.util.Timer createTimer()` throws `UnavailableException`

`java.util.Timer createTimer()`

Throws `UnavailableException`:
createTimer

```java
Timer createTimer()
  throws UnavailableException
```

Creates a new `java.util.Timer` instance. The `Timer` instance could be used to perform periodic `Work` executions or other tasks.

**Returns:**

- a new `Timer` instance.

** Throws:**

- `UnavailableException` - indicates that a `Timer` instance is not available. The request may be retried later.
javax.management.j2ee.statistics  

**Interface BoundaryStatistic**

**All Superinterfaces:**  
Statistic

**All Known Subinterfaces:**  
BoundedRangeStatistic

```java
public interface BoundaryStatistic
extends Statistic

Implements: Statistic
Implemented by: BoundedRangeStatistic
```

Specifies standard measurements of the upper and lower limits of the value of an attribute.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getLowerBound()</code></td>
<td>The lower limit of the value of this attribute. The upper limit of the value of this attribute.</td>
</tr>
<tr>
<td><code>getUpperBound()</code></td>
<td>The upper limit of the value of this attribute.</td>
</tr>
</tbody>
</table>

Methods inherited from interface  
javax.management.j2ee.statistics.Statistic

getDescription, getLastSampleTime, getName, getStartTime, getUnit
public long getUpperBound()

getUpperBound

long getUpperBound()

The upper limit of the value of this attribute.

public long getLowerBound()

getLowerBound

long getLowerBound()

The lower limit of the value of this attribute. The upper limit of the value of this attribute.
javax.management.j2ee.statistics Interface

BoundedRangeStatistic

All Superinterfaces:
   BoundaryStatistic, RangeStatistic, Statistic

public interface BoundedRangeStatistic
extends BoundaryStatistic, RangeStatistic

Implements: BoundaryStatistic, RangeStatistic

BoundedRangeStatistic RangeStatistic BoundaryStatistic

The BoundedRangeStatistic model aggregates the attributes of RangeStatistic and BoundaryStatistic and provides standard measurements of a range that has fixed limits.

Method Summary

Methods inherited from interface
javax.management.j2ee.statistics.BoundaryStatistic
getLowerBound, getUpperBound

Methods inherited from interface
javax.management.j2ee.statistics.Statistic
description, getLastSampleTime, getName, getStartTime, getUnit

Methods inherited from interface
javax.management.j2ee.statistics.RangeStatistic
current, getHighWaterMark, getLowWaterMark
Methods inherited from interface javax.management.j2ee.statistics.Statistic
getDescription, getLastSampleTime, getName, getStartTime, getUnit

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  SUMMARY: NESTED | FIELD | CONSTR | METHOD  FRAMES  NO FRAMES  DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL:</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.registry Interface BulkResponse

All Superinterfaces:
   JAXRResponse

public interface BulkResponse
extends JAXRResponse

Implements: JAXRResponse

API

BulkResponse getStatus
JAXRResponse.STATUS_WARNING JAXRException Collection
BulkResponse JAXRException

See also javax.xml.registry.QueryManager,
javax.xml.registry.LifeCycleManager

Contains the response of a method in the API that performs a bulk operation and returns a bulk response. Partial commits are allowed on a bulk operation.

In the event of a partial success where only a subset of objects were processed successfully, the getStatus method of the BulkResponse must return JAXRResponse.STATUS_WARNING. In this case, a Collection of JAXRException instances is included in the BulkResponse instance. The JAXRExceptions provide information on each error that prevented some objects in the request to not be processed successfully.

Author:
   Farrukh S. Najmi

See Also:
   QueryManager, LifeCycleManager
Field Summary

Fields inherited from interface javax.xml.registry.JAXRResponse
STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING

Method Summary

<table>
<thead>
<tr>
<th>Collection</th>
<th>getCollection()</th>
<th>Get the Collection of objects returned as a response of a bulk operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>getExceptions()</td>
<td>Get the Collection of RegistryException instances in case of partial commit.</td>
</tr>
<tr>
<td>boolean</td>
<td>isPartialResponse()</td>
<td>Determines whether the response is a partial response due to large result set.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.registry.JAXRResponse
getRequestId, getStatus, isAvailable

Method Detail

public java.util.Collection<E> getCollection() throws JAXRException
Collection

return RegistryObject CollectionCollection null
Throws JAXRException: JAXR
See also javax.xml.registry.infomodel.RegistryObject
getCollection

`Collection getCollection() throws JAXRException`

Get the Collection of objects returned as a response of a bulk operation. Caller thread will block here if result is not yet available.

**Capability Level:** 0

**Returns:**

- Collection of RegistryObject instances. The Collection may be empty but not null.

**Throws:**

- `JAXRException` - If the JAXR provider encounters an internal error

**See Also:**

- `RegistryObject`

---

`public java.util.Collection<E> getExceptions() throws JAXRException`

RegistryException Collection

0

```java
return RegistryException Collection RegistryException null
```

**Throws**

- `JAXRException`: JAXR

**See also**

- `javax.xml.registry.RegistryException`

---

getExceptions

`Collection getExceptions() throws JAXRException`

Get the Collection of RegistryException instances in case of partial commit. Caller thread will block here if result is not yet available.
Capability Level: 0

Returns:
Collection of RegistryException instances. Return null if result is available and there is no RegistryException.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
RegistryException

public boolean isPartialResponse() throws JAXRException

0

    return true false

Throws JAXRException: JAXR

isPartialResponse

boolean isPartialResponse() throws JAXRException

Determines whether the response is a partial response due to large result set.

Capability Level: 0

Returns:
true if the response is partial; false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
The BusinessLifeCycleManager interface, which is exposed by the Registry Service, implements the life cycle management functionality of the Registry as part of a business level API. Note that there is no authentication information provided, because the Connection interface keeps that state and context on behalf of the client.

Author:
Farrukh S. Najmi

See Also:
LifeCycleManager

### Field Summary

Fields inherited from interface `javax.xml.registry.LifeCycleManager`

ASSOCIATION, AUDITABLE_EVENT, CLASSIFICATION, CLASSIFICATION_SCHEME, CONCEPT, EMAIL_ADDRESS, EXTERNAL_IDENTIFIER, EXTERNAL_LINK, EXTRINSIC_OBJECT, INTERNATIONAL_STRING, KEY, LOCALIZED_STRING, ORGANIZATION, PERSON_NAME, POSTAL_ADDRESS, REGISTRY_ENTRY, REGISTRY_PACKAGE, SERVICE, SERVICE_BINDING, SLOT, SPECIFICATION_LINK,
<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong></td>
<td><code>confirmAssociation(Association assoc)</code>&lt;br&gt;Confirms this Association by the User associated with the caller.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteAssociations(Collection associationKeys)</code>&lt;br&gt;Deleted the Associations corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteClassificationSchemes(Collection schemeKeys)</code>&lt;br&gt;Deleted the ClassificationSchemes corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteConcepts(Collection conceptKeys)</code>&lt;br&gt;Deleted the Concepts corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteOrganizations(Collection organizationKeys)</code>&lt;br&gt;Deleted the organizations corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteServiceBindings(Collection bindingKeys)</code>&lt;br&gt;Deleted the ServiceBindings corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteServices(Collection serviceKeys)</code>&lt;br&gt;Deleted the services corresponding to the specified Keys.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveAssociations(Collection associations, boolean replace)</code>&lt;br&gt;Saves the specified Association instances.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveClassificationSchemes(Collection schemes)</code>&lt;br&gt;Saves the specified ClassificationScheme instances.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveConcepts(Collection concepts)</code>&lt;br&gt;Saves the specified Concepts.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveOrganizations(Collection organizations)</code>&lt;br&gt;Saves the specified Organizations.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveServiceBindings(Collection bindings)</code>&lt;br&gt;Saves the specified ServiceBindings.</td>
</tr>
</tbody>
</table>
BulkResponse saveServices(Collection services)
Saves the specified Services.

void unConfirmAssociation(Association assoc)
Undoes a previous confirmation of this Association by the User associated with the caller.

Methods inherited from interface javax.xml.registry.LifeCycleManager
createAssociation, createClassification, createClassification, createClassificationScheme, createClassificationScheme, createConcept, createConcept, createEmailAddress, createEmailAddress, createExternalIdentifier, createExternalIdentifier, createExternalLink, createExternalLink, createExtrinsicObject, createInternationalString, createInternationalString, createInternationalString, createKey, createLocalizedString, createLocalizedString, createObject, createOrganization, createOrganization, createPersonName, createPersonName, createPostalAddress, createRegistryPackage, createRegistryPackage, createService, createService, createServiceBinding, createSlot, createSlot, createSpecificationLink, createTelephoneNumber, createUser, deleteObjects, deleteObjects, deprecateObjects, getRegistryService, saveObjects, unDeprecateObjects

Method Detail

public BulkResponse saveOrganizations(java.util.Collection&lt;E&gt; organizations) throws JAXRException
Organization

SaveException

0
saveOrganizations

BulkResponse saveOrganizations(Collection organizations) throws JAXRException

Saves the specified Organizations.

If the object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

**Capability Level:** 0

**Parameters:**
- organizations - the Collection of Organization objects to be saved

**Returns:**
- a BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

---

public BulkResponse saveServices(java.util.Collection<E> services) throws JAXRException
Service
SaveException

0

services Service Collection
return BulkResponse Collection SaveException
Throws JAXRException: JAXR

saveServices

BulkResponse saveServices(Collection services) throws JAXRException

Saves the specified Services.

If the object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

Capability Level: 0

Parameters:
services - the Collection of Service objects to be saved

Returns:
a BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit

Throws:
JAXRException - if the JAXR provider encounters an internal error

public BulkResponse saveServiceBindings(java.util.Collection<E> bindings)
throws JAXRException

ServiceBinding

SaveException

0

bindings ServiceBinding Collection

return BulkResponse Collection SaveException

Throws JAXRException: JAXR

saveServiceBindings

BulkResponse saveServiceBindings(Collection bindings) throws JAXRException

Saves the specified ServiceBindings.

If the object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

Capability Level: 0

Parameters:

bindings - the Collection of ServiceBinding objects to be saved

Returns:

a BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit

Throws:

JAXRException - if the JAXR provider encounters an internal
public BulkResponse saveConcepts(java.util.Collection<E> concepts) throws JAXRException

Concept

SaveException

0

<table>
<thead>
<tr>
<th>concepts</th>
<th>Concept Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>BulkResponse Collection SaveException</td>
</tr>
<tr>
<td>Throws</td>
<td>JAXRException: JAXR</td>
</tr>
</tbody>
</table>

saveConcepts

BulkResponse saveConcepts(Collection concepts)

throws JAXRException

Saves the specified Concepts.

If the object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

Capability Level: 0

Parameters:

- concepts - the Collection of Concept objects to be saved

Returns:

- a BulkResponse containing the Collection of keys for those
objects that were saved successfully and any SaveException that was encountered in case of partial commit

Throws:
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public BulkResponse saveClassificationSchemes(java.util.Collection<? extends ClassificationScheme> schemes) throws JAXRException
```

### SaveException

0

```java
schemes ClassificationScheme Collection
return BulkResponse Collection SaveException
throws JAXRException: JAXR
```

### saveClassificationSchemes

```java
BulkResponse saveClassificationSchemes(Collection schemes) throws JAXRException
```

Saves the specified ClassificationScheme instances.

If the object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

**Capability Level: 0**
Parameters:
  schemes - the Collection of ClassificationScheme objects to be saved

Returns:
  a BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit

Throws:
  JAXRException - if the JAXR provider encounters an internal error

public BulkResponse saveAssociations(java.util.Collection<E> associations, boolean replace) throws JAXRException

SaveException

0

associations  Association  Collection
replace   true  false
return   BulkResponse  Collection  SaveException

Throws       JAXRException:  JAXR

saveAssociations

BulkResponse saveAssociations(Collection associations, boolean replace)
  throws JAXRException

Saves the specified Association instances.

If the object is not in the registry, it is created in the registry. If it
already exists in the registry and has been modified, its state is updated (replaced) in the registry.

Partial commits are allowed. Processing stops on first SaveException encountered.

**Capability Level: 0**

**Parameters:**
- `associations` - the Collection of Association objects to be saved
- `replace` - If set to true, the specified associations replace any existing associations owned by the caller. If set to false, the specified associations are saved, while any existing associations not being updated by this call are preserved.

**Returns:**
- A BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit.

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public BulkResponse deleteOrganizations(java.util.Collection<E> organizationKeys) throws JAXRException
```

**Key**

**DeleteException**

0

```java
organizationKeys Organization Collection
return BulkResponse Collection DeleteException
Throws JAXRException: JAXR
```
deleteOrganizations

BulkResponse deleteOrganizations(Collection organizationKeys) throws JAXRException

Deletes the organizations corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

Capability Level: 0

Parameters:
  organizationKeys - the Collection of keys for the Organization objects to be deleted

Returns:
  a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

Throws:
  JAXRException - if the JAXR provider encounters an internal error

public BulkResponse deleteServices(java.util.Collection<E> serviceKeys) throws JAXRException

Key

DeleteException

0

serviceKeys Service Collection
return BulkResponse Collection DeleteException
Throws JAXRException: JAXR
deleteServices

BulkResponse deleteServices(Collection serviceKeys) throws JAXRException

Deletes the services corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

Capability Level: 0

Parameters:
  serviceKeys - the Collection of keys for the Service objects to be deleted

Returns:
  a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

Throws:
  JAXRException - if the JAXR provider encounters an internal error

public BulkResponse deleteServiceBindings(java.util.Collection<E> bindingKeys) throws JAXRException

Key ServiceBinding

DeleteException

0

bindingKeys ServiceBinding Collection
return BulkResponse Collection DeleteException
Throws JAXRException: JAXR
deleteServiceBindings

BulkResponse deleteServiceBindings(Collection bindingKeys) throws JAXRException

Deletes the ServiceBindings corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

Capability Level: 0

Parameters:
  bindingKeys - the Collection of keys for the ServiceBinding objects to be deleted

Returns:
  a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

Throws:
  JAXRException - if the JAXR provider encounters an internal error

public BulkResponse deleteConcepts(java.util.Collection<E> conceptKeys) throws JAXRException
  Key Concept

  DeleteException

  0

  conceptKeys Concept Collection
  return BulkResponse Collection DeleteException
  Throws JAXRException: JAXR
**deleteConcepts**

BulkResponse deleteConcepts(Collection conceptKeys) throws JAXRException

Deletes the Concepts corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

**Capability Level: 0**

**Parameters:**
- conceptKeys - the Collection of keys for the Concept objects to be deleted

**Returns:**
- a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

---

public BulkResponse deleteClassificationSchemes(java.util.Collection<E> schemeKeys) throws JAXRException

**Key ClassificationScheme**

**DeleteException**

0

schemeKeys ClassificationScheme Collection

return BulkResponse Collection DeleteException

**Throws**

JAXRException: JAXR
deleteClassificationSchemes

BulkResponse deleteClassificationSchemes(Collection schemeKeys) throws JAXRException

Deletes the ClassificationSchemes corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

Capability Level: 0

Parameters:
schemeKeys - the Collection of keys for the ClassificationScheme objects to be deleted

Returns:
a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

Throws:
JAXRException - if the JAXR provider encounters an internal error

public BulkResponse deleteAssociations(java.util.Collection<E> associationKeys) throws JAXRException
Key Association

DeleteException

0

associationKeys Association Collection
return BulkResponse Collection DeleteException
Throws JAXRException: JAXR
deleteAssociations

BulkResponse deleteAssociations(Collection associationKeys) throws JAXRException

Deletes the Associations corresponding to the specified Keys.

Partial commits are allowed. Processing stops on first DeleteException encountered.

Capability Level: 0

Parameters:
associationKeys - the Collection of keys for the Association objects to be deleted

Returns:
a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

Throws:
JAXRException - if the JAXR provider encounters an internal error

public void confirmAssociation(Association assoc) throws JAXRException, InvalidRequestException

User Association Association User sourceObject targetObject

- Association
- Association isConfirmed true
- Association User

Association UDDI ebXML JAXR

0

assoc Association
void confirmAssociation(Association assoc)
throws JAXRException, 
InvalidRequestException

Confirms this Association by the User associated with the caller. The User must be the owner of the sourceObject or the targetObject in this association.

- If the Association is intramural, this method does nothing and returns.
- If the Association is extramural and already confirmed (isConfirmed returns true), this method does nothing and returns.
- If the Association is extramural and not already confirmed, this method confirms the association for the User associated with the caller.

The details of confirming an extramural Association are registry-specific. For UDDI and ebXML registries, the registry-specific details are described in the JAXR specification.

**Capability Level: 0**

**Parameters:**

assoc - the Association object to be confirmed

**Throws:**

JAXRException - if the JAXR provider encounters an internal error
InvalidRequestException - if the User is not the owner of the sourceObject or the targetObject
public void unConfirmAssociation(Association assoc) throws JAXRException, InvalidRequestException
User Association Association User
sourceObject targetObject

- Association
- Association isConfirmed true
- Association

Association UDDI ebXML JAXR

0

assoc Association

Throws JAXRException: JAXR
Throws InvalidRequestException: User sourceObject targetObject

unConfirmAssociation

void unConfirmAssociation(Association assoc) throws JAXRException, InvalidRequestException

Undoes a previous confirmation of this Association by the User associated with the caller. The User must be the owner of the sourceObject or the targetObject in this association.

- If the Association is intramural, this method does nothing and returns.
- If the Association is extramural and already confirmed (isConfirmed returns true), this method unconfirms it for the requestor.
- If the Association is extramural and not already confirmed, this method does nothing and returns.

The details of unconfirming an extramural Association are registry-
specific. For UDDI and ebXML registries, the registry-specific details are described in the JAXR specification.

**Capability Level: 0**

**Parameters:**
- `assoc` - the Association object to be unconfirmed

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error
- `InvalidRequestException` - if the User is not the owner of the sourceObject or the targetObject
javax.xml.registry Interface BusinessQueryManager

All Superinterfaces:  
 QueryManager

public interface BusinessQueryManager
extends QueryManager

Implements:  QueryManager

BusinessQueryManager  Registry Service

See also  javax.xml.registry.DeclarativeQueryManager,  
javax.xml.registry.FindQualifier,  
javax.xml.registry.BulkResponse

The BusinessQueryManager interface, which is exposed by the Registry Service, implements the business style query interface. It is also referred to as the focused query interface.

Author:  
Farrukh S. Najmi

See Also:  
DeclarativeQueryManager, FindQualifier, BulkResponse

Method Summary

<table>
<thead>
<tr>
<th>BulkResponse</th>
<th>findAssociations(Collection findQualifiers, String sourceObjectId, String targetObjectId, Collection associationTypes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Finds all Association objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>BulkResponse</th>
<th>findCallerAssociations(Collection findQualifiers, Boolean confirmedByCaller, Boolean confirmedByOther</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Collection associationTypes)</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassificationScheme</td>
<td><code>findClassificationSchemeByName</code></td>
<td>Finds a ClassificationScheme by name based on specified find qualifiers and name pattern.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findClassificationSchemes</code></td>
<td>Finds all ClassificationScheme objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
<tr>
<td>Concept</td>
<td><code>findConceptByPath</code></td>
<td>Finds a Concept based on the path specified.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findConcepts</code></td>
<td>Finds all Concept objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findOrganizations</code></td>
<td>Finds all Organization objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findRegistryPackages</code></td>
<td>Finds all RegistryPackage objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findServiceBindings</code></td>
<td>Finds all ServiceBinding objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>findServices</code></td>
<td>Finds all Service objects that match all of the criteria specified by the parameters of this call.</td>
</tr>
</tbody>
</table>
Methods inherited from interface javax.xml.registry.QueryManager
getRegistryObject, getRegistryObject, getRegistryObjects,
getRegistryObjects, getRegistryObjects, getRegistryObjects,
getRegistryService

Method Detail

public BulkResponse findAssociations(java.util.Collection<E> findQualifiers,
String sourceObjectId, String targetObjectId,
java.util.Collection<E> associationTypes) throws JAXRException

Association null AND Association

findQualifiers FindQualifier find CollectionFindQualifier
boolean (predicate logic)
sourceObjectId RegistryObject ID String RegistryObject
Association sourceObject null
targetObjectId RegistryObject ID String RegistryObject
Association targetObject null
associationTypes associationTypes Collection Collection OR
null
return BulkResponse Association Collection
Throws JAXRException: JAXR

findAssociations

BulkResponse findAssociations(Collection findQualifiers,
String sourceObjectId,
String targetObjectId,
Collection associationTypes)
throws JAXRException
Finds all Association objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters. If no parameters are specified, no Associations are returned.

**Capability Level: 0**

**Parameters:**
- `findQualifiers` - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
- `sourceObjectId` - a String that represents the id for a RegistryObject that must be the sourceObject of the Associations that match. This parameter is ignored if specified as null.
- `targetObjectId` - a String that represents the id for a RegistryObject that must be the targetObject of the Associations that match. This parameter is ignored if specified as null.
- `associationTypes` - a Collection of associationTypes. This is a logical OR operation across the collection. This parameter is ignored if specified as null.

**Returns:**
a BulkResponse containing a Collection of Associations

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public BulkResponse findCallerAssociations(java.util.Collection<E> findQualifiers, Boolean confirmedByCaller, Boolean confirmedByOtherParty, java.util.Collection<E> associationTypes) throws JAXRException
    Association null AND Association

Connection

0
```
findCallerAssociations

BulkResponse findCallerAssociations(Collection findQualifiers, Boolean confirmedByCaller, Boolean confirmedByOtherParty, Collection associationTypes) throws JAXRException

Finds all Association objects owned by the caller that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters. If no parameters are specified, no Associations are returned.

This call is sensitive to the identity of the caller and is a privileged operation that requires the caller to have provided its identity credentials to the Connection associated with this object.

Capability Level: 0

Parameters:

findQualifiers - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.

confirmedByCaller - If true, provider must include Associations involving the caller that have been confirmed by the caller. If
false, provider must include Associations involving the caller that have not been confirmed by the caller. This parameter is ignored if specified as null.

confirmedByOtherParty - If true, provider must include Associations involving the caller that have been confirmed by the other party. If false, provider must include Associations involving the caller that have not been confirmed by the other party. This parameter is ignored if specified as null.

associationTypes - a Collection of associationTypes. This is a logical OR operation across the collection. This parameter is ignored if specified as null.

**Returns:**
a BulkResponse containing a Collection of Associations

**Throws:**

JAXRException - if the JAXR provider encounters an internal error

```java
public BulkResponse findOrganizations(java.util.Collection<E> findQualifiers,
java.util.Collection<E> namePatterns,
java.util.Collection<E> classifications,
java.util.Collection<E> specifications,
java.util.Collection<E> externalIdentifiers,
java.util.Collection<E> externalLinks) throws JAXRException

Organization null AND

0

<table>
<thead>
<tr>
<th>findQualifiers</th>
<th>FindQualifier find CollectionFindQualifier</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean (predicate logic)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>namePatterns</th>
<th>LocalizedString String localizedString Collection String</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL-92 LIKE findQualifier OR Classification Collection UDDI</td>
<td></td>
</tr>
<tr>
<td>catgegoryBag findQualifier AND</td>
<td></td>
</tr>
</tbody>
</table>
findOrganizations

BulkResponse findOrganizations(Collection findQualifiers,
Collection namePatterns,
Collection classifications,
Collection specifications,
Collection externalIdentifiers,
Collection externalLinks)
throws JAXRException

Finds all Organization objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters.

Capability Level: 0

Parameters:

findQualifiers - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
namePatterns - a Collection that may consist of either String or...
LocalizedString objects. Each String or value within a LocalizedString is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification. Unless otherwise specified in findQualifiers, this is a logical OR, and a match on any name qualifies as a match for this criterion.

classifications - a Collection of Classification objects that classify the object. It is analogous to a catgegoryBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createClassification method to create a transient Classification for use in this Collection.

specifications - a Collection of RegistryObjects that represent (proxy) a technical specification. It is analogous to a tModelBag in the UDDI specification. In the case of a UDDI provider, the RegistryObject is a specification Concept. In the case of an ebXML provider, the RegistryObject is likely to be an ExtrinsicObject. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Specifications qualifies as a match for this criterion.

externalIdentifiers - a Collection of ExternalIdentifier objects that provide an external identifier for the object using an identification scheme such as DUNS. It is analogous to an identifierBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalIdentifiers qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createExternalIdentifier method to create a transient ExternalIdentifier for use in this Collection.

externalLinks - a Collection of ExternalLink objects that link the object to content outside the registry. It is analogous to an overviewDoc in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalLinks qualifies as a match for this criterion.

Returns:
- a BulkResponse containing a Collection of Organizations

Throws:
- JAXRException - if the JAXR provider encounters an internal error
```java
public BulkResponse findServices(Key orgKey,
java.util.Collection<E> findQualifiers,
java.util.Collection<E> namePatterns,
java.util.Collection<E> classifications,
java.util.Collection<E> specifications) throws JAXRException

Service null AND

<table>
<thead>
<tr>
<th>orgKey</th>
<th>Organization Key UDDI</th>
</tr>
</thead>
<tbody>
<tr>
<td>findQualifiers</td>
<td>FindQualifier find CollectionFindQualifier boolean (predicate logic) String LocalizedString Collection String</td>
</tr>
<tr>
<td>namePatterns</td>
<td>LocalizedString SQL-92 LIKE findQualifier OR Classification Collection UDDI catgegoryBag findQualifier AND</td>
</tr>
<tr>
<td>classifications</td>
<td>Classification LifeCycleManager.createClassification Classification Collection</td>
</tr>
<tr>
<td>specifications</td>
<td>RegistryObject Collection UDDI tModelBag UDDI RegistryObject Concept ebXML RegistryObject ExtrinsicObject findQualifier AND Specification</td>
</tr>
</tbody>
</table>

return BulkResponse Service Collection

Throws JAXRException: JAXR
```

**findServices**

```java
BulkResponse findServices(Key orgKey,
Collection findQualifiers,
Collection namePatterns,
Collection classifications,
Collection specifications)
throws JAXRException
```
Finds all Service objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters.

**Capability Level: 0**

**Parameters:**

- `orgKey`: Key identifying an Organization. Required for UDDI providers.
- `findQualifiers`: a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
- `namePatterns`: a Collection that may consist of either String or LocalizedString objects. Each String or value within a LocalizedString is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification. Unless otherwise specified in `findQualifiers`, this is a logical OR, and a match on any name qualifies as a match for this criterion.
- `classifications`: a Collection of Classification objects that classify the object. It is analogous to a categoryBag in the UDDI specification. Unless otherwise specified in `findQualifiers`, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the `LifeCycleManager.createClassification` method to create a transient Classification for use in this Collection.
- `specifications`: a Collection of RegistryObjects that represent (proxy) a technical specification. It is analogous to a tModelBag in the UDDI specification. In the case of a UDDI provider, the RegistryObject is a specification Concept. In the case of an ebXML provider, the RegistryObject is likely to be an ExtrinsicObject. Unless otherwise specified in `findQualifiers`, this is a logical AND, and a match on all specified Specifications qualifies as a match for this criterion.

**Returns:**

- a BulkResponse containing a Collection of Services

**Throws:**

- `JAXRException` - if the JAXR provider encounters an internal error
public BulkResponse findServiceBindings(Key serviceKey, java.util.Collection<E> findQualifiers, java.util.Collection<E> classifications, java.util.Collection<E> specifications) throws JAXRException

ServiceBinding null AND

0

serviceKey Service Key UDDI

findQualifiers FindQualifier find CollectionFindQualifier

boolean (predicate logic)

Classification Collection UDDI
categoryBag findQualifier AND
classifications Classification

LifeCycleManager.createClassification Classification

Collection

RegistryObject Collection UDDI
tModelBag UDDI RegistryObject Concept
ebXML RegistryObject ExtrinsicObject

findQualifier AND Specification

specifications RegistryObject Concept

return BulkResponse ServiceBinding Collection

Throws JAXRException: JAXR

findServiceBindings

BulkResponse findServiceBindings(Key serviceKey, Collection findQualifiers, Collection classifications, Collection specifications) throws JAXRException

Finds all ServiceBinding objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters.

Capability Level: 0
Parameters:

serviceKey - Key identifying a Service. Required for UDDI providers.
findQualifiers - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
classifications - a Collection of Classification objects that classify the object. It is analogous to a categorogyBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the LifecycleManager.createClassification method to create a transient Classification for use in this Collection.
specifications - a Collection of RegistryObjects that represent (proxy) a technical specification. It is analogous to a tModelBag in the UDDI specification. In the case of a UDDI provider, the RegistryObject is a specification Concept. In the case of an ebXML provider, the RegistryObject is likely to be an ExtrinsicObject. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Specifications qualifies as a match for this criterion.

Returns:

a BulkResponse containing a Collection of ServiceBindings

Throws:

JAXRException - if the JAXR provider encounters an internal error

```java
public BulkResponse findClassificationSchemes(java.util.Collection<E> findQualifiers, java.util.Collection<E> namePatterns, java.util.Collection<E> classifications, java.util.Collection<E> externalLinks) throws JAXRException
ClassificationScheme null AND 0
```
findQualifiers - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.

namePatterns - a Collection that may consist of either String or LocalizedString objects. Each String or value within a LocalizedString is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification. Unless otherwise specified in findQualifiers, this is a logical OR, and a match on any name qualifies as a match for this criterion.
classifications - a Collection of Classification objects that classify the object. It is analogous to a categoryBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createClassification method to create a transient Classification for use in this Collection.

externalLinks - a Collection of ExternalLink objects that link the object to content outside the registry. It is analogous to an overviewDoc in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalLinks qualifies as a match for this criterion.

Returns:
a BulkResponse containing a Collection of ClassificationSchemes

Throws:
JAXRException - if the JAXR provider encounters an internal error

public ClassificationScheme findClassificationSchemeByName(java.util.Collection<E> findQualifiers, String namePattern) throws JAXRException

ClassificationScheme

findQualifiers    FindQualifier find CollectionFindQualifier boolean (predicate logic)
namePattern       String SQL-92 LIKE
return            namePattern ClassificationScheme null

Throws
JAXRException: JAXR
InvalidRequestException

findClassificationSchemeByName
Finds a ClassificationScheme by name based on the specified find qualifiers and name pattern.

**Capability Level:** 0

**Parameters:**
- `findQualifiers` - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
- `namePattern` - a String that is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification

**Returns:**
the ClassificationScheme matching the namePattern, or null if no match is found.

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error. Throws an InvalidRequestException if multiple matches are found.

```java
```

Concept null AND

0

`findQualifiers` FindQualifier find CollectionFindQualifier
boolean (predicate logic)
String LocalizedString Collection String
findConcepts

BulkResponse findConcepts(Collection findQualifiers, Collection namePatterns, Collection classifications, Collection externalIdentifiers, Collection externalLinks) throws JAXRException

Finds all Concept objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters.

Capability Level: 0

Parameters:

findQualifiers - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.

namePatterns - a Collection that may consist of either String or LocalizedString objects. Each String or value within a
LocalizedString is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification. Unless otherwise specified in findQualifiers, this is a logical OR, and a match on any name qualifies as a match for this criterion.

classifications - a Collection of Classification objects that classify the object. It is analogous to a categoryBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createClassification method to create a transient Classification for use in this Collection.

externalIdentifiers - a Collection of ExternalIdentifier objects that provide an external identifier for the object using an identification scheme such as DUNS. It is analogous to an identifierBag in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalIdentifiers qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createExternalIdentifier method to create a transient ExternalIdentifier for use in this Collection.

externalLinks - a Collection of ExternalLink objects that link the object to content outside the registry. It is analogous to an overviewDoc in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalLinks qualifies as a match for this criterion.

Returns:
a BulkResponse containing a Collection of Concepts

Throws:
JAXRException - if the JAXR provider encounters an internal error

public Concept findConceptByPath(String path) throws JAXRException

Concept Concept Concept

0
findConceptByPath

```java
Concept findConceptByPath(String path)
throws JAXRException
```

Finds a Concept based on the path specified. If the specified path matches more than one Concept, the one that is most general (higher in the concept hierarchy) is returned.

**Capability Level:** 0

**Parameters:**
- `path` - a canonical path expression, as defined in the JAXR specification, that identifies the Concept.

**Returns:**
- the Concept found, or null if no match is found

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

---

```java
public BulkResponse findRegistryPackages(java.util.Collection<E> findQualifiers,
java.util.Collection<E> namePatterns,
java.util.Collection<E> classifications,
java.util.Collection<E> externalLinks) throws JAXRException
```

RegistryPackage null AND

1

```
findQualifiers FindQualifier find CollectionFindQualifier
boolean (predicate logic)
```
**findRegistryPackages**

*BulkResponse* `findRegistryPackages(Collection findQualifiers, Collection namePatterns, Collection classifications, Collection externalLinks) throws JAXRException*

Finds all RegistryPackage objects that match all of the criteria specified by the parameters of this call. This is a logical AND operation between all non-null parameters.

**Capability Level: 1**

**Parameters:**

- `findQualifiers` - a Collection of find qualifiers as defined by the FindQualifier interface, which specifies qualifiers that affect string matching, sorting, boolean predicate logic, and the like.
- `namePatterns` - a Collection that may consist of either String or LocalizedString objects. Each String or value within a LocalizedString is a partial or full name pattern with wildcard searching as specified by the SQL-92 LIKE specification. Unless otherwise specified in findQualifiers, this is a logical OR, and a match on any name qualifies as a match for this criterion.
- `classifications` - a Collection of Classification objects that classify the object. It is analogous to a categoryBag in the
UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified Classifications qualifies as a match for this criterion. The programmer may use the LifeCycleManager.createClassification method to create a transient Classification for use in this Collection.

`externalLinks` - a Collection of ExternalLink objects that link the object to content outside the registry. It is analogous to an overviewDoc in the UDDI specification. Unless otherwise specified in findQualifiers, this is a logical AND, and a match on all specified ExternalLinks qualifies as a match for this criterion.

**Returns:**
a BulkResponse containing a Collection of RegistryPackages

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error
public class ByteArrayDataSource
extends Object
implements DataSource

Implements: DataSource

byte DataSource byte InputStream String byte

since JavaMail 1.4

A DataSource backed by a byte array. The byte array may be passed in directly, or may be initialized from an InputStream or a String.

Since:
JavaMail 1.4

Author:
John Mani, Bill Shannon, Max Spivak

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ByteArrayDataSource</strong></td>
<td></td>
</tr>
<tr>
<td><code>(byte[] data, String type)</code></td>
<td>Create a ByteArrayDataSource with data from the specified byte array and with the specified MIME type.</td>
</tr>
<tr>
<td><strong>ByteArrayDataSource</strong></td>
<td></td>
</tr>
<tr>
<td><code>(InputStream is, String type)</code></td>
<td>Create a ByteArrayDataSource with data from the specified InputStream and with the specified MIME type.</td>
</tr>
<tr>
<td><strong>ByteArrayDataSource</strong></td>
<td></td>
</tr>
<tr>
<td><code>(String data, String type)</code></td>
<td>Create a ByteArrayDataSource with data from the specified String and with the specified MIME type.</td>
</tr>
</tbody>
</table>
Create a ByteArrayDataSource with data from the specified String and with the specified MIME type.

## Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>String getContentType()</code></td>
<td>Get the MIME content type of the data.</td>
</tr>
<tr>
<td><code>InputStream</code></td>
<td><code>getInputStream()</code></td>
<td>Return an InputStream for the data.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getName()</code></td>
<td>Get the name of the data.</td>
</tr>
<tr>
<td><code>OutputStream</code></td>
<td><code>getOutputStream()</code></td>
<td>Return an OutputStream for the data.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setName(String name)</code></td>
<td>Set the name of the data.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`.

## Constructor Detail

```java
public ByteArrayDataSource(java.io.InputStream is, String type) throws java.io.IOException
{
    this(is, null, type);
}

public ByteArrayDataSource(InputStream is, String name, String type)
{
    this(is, name, type);
}

public ByteArrayDataSource(InputStream is, String name)
{
    this(is, name, null);
}

public ByteArrayDataSource(String name)
{
    this(null, name, null);
}
```

ByteArrayDataSource

```java
public ByteArrayDataSource(InputStream is, String name, String type)
{
    super(is, null, type);
    super.setName(name);
}
```

Throws java.io.IOException:
Create a ByteArrayInputStream with data from the specified InputStream and with the specified MIME type. The InputStream is read completely and the data is stored in a byte array.

**Parameters:**
- is - the InputStream
- type - the MIME type

**Throws:**
- `IOException` - errors reading the stream

```java
public ByteArrayInputStream(byte[] data, String type) throws IOException
```

**ByteArrayDataSource**

Create a ByteArrayInputStream with data from the specified byte array and with the specified MIME type.

**Parameters:**
- data - the data
- type - the MIME type

```java
public ByteArrayInputStream(data, type) throws java.io.IOException
```

```java
public ByteArrayInputStream(String data, String type) throws java.io.IOException
```
ByteArrayDataSource

public ByteArrayDataSource(String data, String type) throws IOException

Create a ByteArrayDataSource with data from the specified String and with the specified MIME type. The MIME type should include a charset parameter specifying the charset to be used for the string. If the parameter is not included, the default charset is used.

Parameters:
- data - the String
- type - the MIME type

Throws:
- IOException - errors reading the String

Method Detail

public java.io.InputStream getInputStream() throws java.io.IOException

InputStream

return InputStream

throws IOException:

getInputStream

public InputStream getInputStream() throws IOException

Return an InputStream for the data. Note that a new stream is returned each time this method is called.
public java.io.OutputStream getOutputStream() throws java.io.IOException

OutputStream

Throws java.io.IOException:

getOutputStream

public OutputStream getOutputStream() throws IOException

Return an OutputStream for the data. Writing the data is not supported; an IOException is always thrown.

Specified by:

getOutputStream in interface DataSource

Returns:

an OutputStream

Throws:

IOException - always

public String getContentType()

MIME

return MIME

gcontentType

public String getContentType()
Get the MIME content type of the data.

**Specified by:**

`getContentType` in interface `DataSource`

**Returns:**

the MIME type

---

```java
public String getName()
{
    return
}
```

**getName**

```java
public String getName()
{
    return
}
```

Get the name of the data. By default, an empty string (""") is returned.

**Specified by:**

`getName` in interface `DataSource`

**Returns:**

the name of this data

---

```java
public void setName(String name)
{
    name
}
```

**setName**

```java
public void setName(String name)
{
    name
}
```

Set the name of the data.

**Parameters:**

`name` - the name of this data
javax.xml.rpc.holders  **Class ByteArrayHolder**

**java.lang.Object**
  | *javax.xml.rpc.holders.ByteArrayHolder*

**All Implemented Interfaces:**
  | **Holder**

---

```
public final class ByteArrayHolder extends Object implements Holder
```

**Implements:** **Holder**

---

### Field Summary

| byte[] value |

### Constructor Summary

| ByteArrayHolder() |
| ByteArrayHolder(byte[] mybyteArray) |

### Method Summary

**Methods inherited from class java.lang.Object**
**clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait**
value

public byte[] value

**Constructor Detail**

public ByteArrayHolder()

ByteArrayHolder

public ByteArrayHolder()

public ByteArrayHolder(byte[] mybyteArray)

ByteArrayHolder

public ByteArrayHolder(byte[] mybyteArray)
PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.faces.convert Class ByteConverter

java.lang.Object
   - javax.faces.convert.ByteConverter

All Implemented Interfaces:
   Converter

public class ByteConverter

extends Object
implements Converter

Implements: Converter

java.lang.Byte byte                  Converter

Converter implementation for java.lang.Byte (and byte primitive) values.

---

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>BYTE_ID</td>
</tr>
<tr>
<td>static String</td>
<td>CONVERTER_ID</td>
</tr>
<tr>
<td>static String</td>
<td>STRING_ID</td>
</tr>
</tbody>
</table>

- BYTE_ID: The message identifier of the FacesMessage to be created if the conversion to Byte fails.
- CONVERTER_ID: The standard converter id for this converter.
- STRING_ID: The message identifier of the FacesMessage to be created if the conversion of the Byte value to String fails.

Constructor Summary

ByteConverter()
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAsObject</strong></td>
<td>Convert the specified string value, which is associated with the specified <strong>UIComponent</strong>, into a model data object that is appropriate for being stored during the <em>Apply Request Values</em> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><strong>getAsString</strong></td>
<td>Convert the specified model object value, which is associated with the specified <strong>UIComponent</strong>, into a String that is suitable for being included in the response generated during the <em>Render Response</em> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.**Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

**CONVERTER_ID**

```java
public static final String CONVERTER_ID
```

The standard converter id for this converter.

**See Also:**
- Constant Field Values
BYTE_ID

public static final String BYTE_ID

The message identifier of the FacesMessage to be created if the conversion to Byte fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

---

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Byte value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail

public ByteConverter()
ByteConverter

```java
public ByteConverter()
```

## Method Detail

```java
public Object getAsObject(FacesContext context, UIComponent component, String value)
```

Throws: 
- `ConverterException`: NullPointerException
- NullPointerException: NullPointerException

### getAsObject

```java
public Object getAsObject(FacesContext context, UIComponent component, String value)
```

**Description copied from interface: Converter**

Convert the specified string value, which is associated with the specified `UIComponent`, into a model data object that is appropriate for being stored during the *Apply Request Values* phase of the request processing lifecycle.

**Specified by:**
- `getAsObject` in interface `Converter`

**Parameters:**
- `context` - `FacesContext` for the request being processed
- `component` - `UIComponent` with which this model object value is associated
- `value` - String value to be converted (may be `null`)

**Returns:**
- `null` if the value to convert is `null`, otherwise the result of the conversion

**Throws:**
- `ConverterException` - if conversion cannot be successfully
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws ConverterException: NullPointerException

Throws NullPointerException: NullPointerException context null component null

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.holders  Class ByteHolder

java.lang.Object
   | javax.xml.rpc.holders.ByteHolder

All Implemented Interfaces:
   Holder

public final class ByteHolder

extends Object
implements Holder

Implements: Holder

Field Summary

| byte  | value |

Constructor Summary

ByteHolder()

ByteHolder(byte mybyte)

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
value

public byte value

Constructor Detail

public ByteHolder()

ByteHolder

public ByteHolder()

public ByteHolder(byte mybyte)

ByteHolder

public ByteHolder(byte mybyte)
PS:
javax.jms Interface BytesMessage

All Superinterfaces:
   Message

public interface BytesMessage extends Message

Implements: Message

BytesMessage Message

BytesMessage java.io.DataInputStream java.io.DataOutputStream

JMS API

BytesMessage.writeInt(6)
BytesMessage.writeObject(new Integer(6))

clearBody reset reset
clearBody

MessageNotReadableException

MessageNotWriteableException

version 1.1 April 2, 2002

A `BytesMessage` object is used to send a message containing a stream of uninterpreted bytes. It inherits from the `Message` interface and adds a bytes message body. The receiver of the message supplies the interpretation of the bytes.

The `BytesMessage` methods are based largely on those found in `java.io.DataInputStream` and `java.io.DataOutputStream`.

This message type is for client encoding of existing message formats. If possible, one of the other self-defining message types should be used instead.

Although the JMS API allows the use of message properties with byte messages, they are typically not used, since the inclusion of properties may affect the format.

The primitive types can be written explicitly using methods for each type. They may also be written generically as objects. For instance, a call to `BytesMessage.writeInt(6)` is equivalent to `BytesMessage.writeObject(new Integer(6))`. Both forms are provided, because the explicit form is convenient for static programming, and the object form is needed when types are not known at compile time.

When the message is first created, and when `clearBody` is called, the body of the message is in write-only mode. After the first call to `reset` has been made, the message body is in read-only mode. After a message has been sent, the client that sent it can retain and modify it without affecting the message that has been sent. The same message object can be sent multiple times. When a message has been received, the provider has called `reset` so that the message body is in read-only mode for the client.

If `clearBody` is called on a message in read-only mode, the message body is cleared and the message is in write-only mode.

If a client attempts to read a message in write-only mode, a `MessageNotReadableException` is thrown.

If a client attempts to write a message in read-only mode, a `MessageNotWritableException` is thrown.
### Field Summary

Fields inherited from interface javax.jms.Message

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_DELIVERY_MODE</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_PRIORITY</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_TIME_TO_LIVE</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>getBodyLength()</td>
<td>Gets the number of bytes of the message body when the message is in read-only mode.</td>
</tr>
<tr>
<td>boolean</td>
<td>readBoolean()</td>
<td>Reads a boolean from the bytes message stream.</td>
</tr>
<tr>
<td>byte</td>
<td>readByte()</td>
<td>Reads a signed 8-bit value from the bytes message stream.</td>
</tr>
<tr>
<td>int</td>
<td>readBytes(byte[] value)</td>
<td>Reads a byte array from the bytes message stream.</td>
</tr>
<tr>
<td>int</td>
<td>readBytes(byte[] value, int length)</td>
<td>Reads a portion of the bytes message stream.</td>
</tr>
<tr>
<td>char</td>
<td>readChar()</td>
<td>Reads a Unicode character value from the bytes message stream.</td>
</tr>
<tr>
<td>double</td>
<td>readDouble()</td>
<td>Reads a double from the bytes message stream.</td>
</tr>
<tr>
<td>float</td>
<td>readFloat()</td>
<td>Reads a float from the bytes message stream.</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>readInt()</code></td>
<td>Reads a signed 32-bit integer from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>readLong()</code></td>
<td>Reads a signed 64-bit integer from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>readShort()</code></td>
<td>Reads a signed 16-bit number from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>readUnsignedByte()</code></td>
<td>Reads an unsigned 8-bit number from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>readUnsignedShort()</code></td>
<td>Reads an unsigned 16-bit number from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>readUTF()</code></td>
<td>Reads a string that has been encoded using a modified UTF-8 format from the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>reset()</code></td>
<td>Puts the message body in read-only mode and repositions the stream of bytes to the beginning.</td>
<td></td>
</tr>
<tr>
<td><code>writeBoolean(boolean value)</code></td>
<td>Writes a boolean to the bytes message stream as a 1-byte value.</td>
<td></td>
</tr>
<tr>
<td><code>writeByte(byte value)</code></td>
<td>Writes a byte to the bytes message stream as a 1-byte value.</td>
<td></td>
</tr>
<tr>
<td><code>writeBytes(byte[] value)</code></td>
<td>Writes a byte array to the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>writeBytes(byte[] value, int offset, int length)</code></td>
<td>Writes a portion of a byte array to the bytes message stream.</td>
<td></td>
</tr>
<tr>
<td><code>writeChar(char value)</code></td>
<td>Writes a char to the bytes message stream as a 2-byte value, high byte first.</td>
<td></td>
</tr>
<tr>
<td><code>writeDouble(double value)</code></td>
<td>Writes a double to the bytes message stream.</td>
<td></td>
</tr>
</tbody>
</table>
Converts the double argument to a long using the doubleToLongBits method in class Double, and then writes that long value to the bytes message stream as an 8-byte quantity, high byte first.

writeFloat(float value)
Converts the float argument to an int using the floatToIntBits method in class Float, and then writes that int value to the bytes message stream as a 4-byte quantity, high byte first.

writeInt(int value)
Writes an int to the bytes message stream as four bytes, high byte first.

writeLong(long value)
Writes a long to the bytes message stream as eight bytes, high byte first.

writeObject(Object value)
Writes an object to the bytes message stream.

writeShort(short value)
Writes a short to the bytes message stream as two bytes, high byte first.

writeUTF(String value)
Writes a string to the bytes message stream using UTF-8 encoding in a machine-independent manner.

Methods inherited from interface javax.jms.Message
Method Detail

public long getBodyLength() throws JMSException
byte

    return

Throws    JMSException: JMS
Throws    MessageNotReadableException:
since 1.1

getBodyLength

long getBodyLength() throws JMSException

Gets the number of bytes of the message body when the message is in read-only mode. The value returned can be used to allocate a byte array. The value returned is the entire length of the message body, regardless of where the pointer for reading the message is currently located.

Returns: number of bytes in the message

Throws: JMSException - if the JMS provider fails to read the message due to some internal error.
MessageNotReadableException - if the message is in write-only mode.

Since: 1.1

public boolean readBoolean() throws JMSException
boolean

    return boolean

Throws: JMSException - if the JMS provider fails to read the message due to some internal error.
throws `JMSException`: JMS

Throws `MessageEOFException`:  

Throws `MessageNotReadableException`:  

---

### readBoolean

```java
boolean readBoolean() throws JMSException
```

Reads a boolean from the bytes message stream.

**Returns:**

the boolean value read

**Throws:**

`JMSException` - if the JMS provider fails to read the message due to some internal error.
`MessageEOFException` - if unexpected end of bytes stream has been reached.
`MessageNotReadableException` - if the message is in write-only mode.

---

### public byte readByte() throws JMSException

```java
byte readByte() throws JMSException
```

Reads a signed 8-bit value from the bytes message stream.

---

### readByte

```java
byte readByte() throws JMSException
```

Reads a signed 8-bit value from the bytes message stream.
Returns:
the next byte from the bytes message stream as a signed 8-bit byte

Throws:
- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of bytes stream has been reached.
- `MessageNotReadableException` - if the message is in write-only mode.

---

public int readUnsignedByte() throws `JMSException`

```
return 8
```

Throws
- `JMSException`: JMS
- `MessageEOFException`
- `MessageNotReadableException`:

---

`readUnsignedByte`

```
int readUnsignedByte() throws JMSException
```

Reads an unsigned 8-bit number from the bytes message stream.

Returns:
the next byte from the bytes message stream, interpreted as an unsigned 8-bit number

Throws:
- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of bytes stream has been reached.
- `MessageNotReadableException` - if the message is in write-only mode.
public short readShort() throws JMSException

return 16

Throws JMSException: JMS
Throws MessageEOFException:
Throws MessageNotReadableException:

readShort

short readShort() throws JMSException

Reads a signed 16-bit number from the bytes message stream.

Returns:
the next two bytes from the bytes message stream, interpreted as a signed 16-bit number

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of bytes stream has been reached.
MessageNotReadableException - if the message is in write-only mode.

public int readUnsignedShort() throws JMSException

return 16

Throws JMSException: JMS
Throws MessageEOFException:
Throws MessageNotReadableException:

readUnsignedShort
int readUnsignedShort() throws JMSException

Reads an unsigned 16-bit number from the bytes message stream.

**Returns:**
the next two bytes from the bytes message stream, interpreted as an unsigned 16-bit integer

**Throws:**

- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of bytes stream has been reached.
- `MessageNotReadableException` - if the message is in write-only mode.

---

public char readChar() throws JMSException

Unicode

```java
    return Unicode
```

**Throws:**

- `JMSException`: JMS
- `MessageEOFException`
- `MessageNotReadableException`

---

char readChar() throws JMSException

Reads a Unicode character value from the bytes message stream.

**Returns:**
the next two bytes from the bytes message stream as a Unicode character

**Throws:**

- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of bytes stream has
been reached.
MessageNotReadableException - if the message is in write-only mode.

```java
public int readInt() throws JMSException
```

Return int

Throws JMSException: JMS

Throws MessageEOFException:

Throws MessageNotReadableException:

readInt

```java
int readInt() throws JMSException
```

Reads a signed 32-bit integer from the bytes message stream.

Returns:
the next four bytes from the bytes message stream, interpreted as an int

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of bytes stream has been reached.
MessageNotReadableException - if the message is in write-only mode.

```java
public long readLong() throws JMSException
```

Return long

Throws JMSException: JMS

Throws MessageEOFException:

Throws MessageNotReadableException:
**readLong**

```java
long readLong() throws JMSException

Returns:
the next eight bytes from the bytes message stream, interpreted as a long
```

**Throws:**
- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of bytes stream has been reached.
- `MessageNotReadableException` - if the message is in write-only mode.

---

**public float readFloat() throws JMSException**

```java
float readFloat() throws JMSException

Returns:
the next four bytes from the bytes message stream, interpreted as a float
```

**Throws:**
- `JMSException`: JMS
- `MessageEOFException`
- `MessageNotReadableException`
Throws:

- **JMSException** - if the JMS provider fails to read the message due to some internal error.
- **MessageEOFException** - if unexpected end of bytes stream has been reached.
- **MessageNotReadableException** - if the message is in write-only mode.

public double readDouble() throws **JMSException**

double

return double

Throws

- **JMSException**: JMS
- **MessageEOFException**: 
- **MessageNotReadableException**: 

readDouble

double readDouble() throws **JMSException**

Reads a double from the bytes message stream.

Returns:

the next eight bytes from the bytes message stream, interpreted as a double

Throws:

- **JMSException** - if the JMS provider fails to read the message due to some internal error.
- **MessageEOFException** - if unexpected end of bytes stream has been reached.
- **MessageNotReadableException** - if the message is in write-only mode.

public String readUTF() throws **JMSException**

UTF-8
return Unicode
Throws JMSException: JMS
Throws MessageEOFException:
Throws MessageNotReadableException:

readUTF

String readUTF()
throws JMSException

Reads a string that has been encoded using a modified UTF-8 format from the bytes message stream.

For more information on the UTF-8 format, see "File System Safe UCS Transformation Format (FSS_UTF)", X/Open Preliminary Specification, X/Open Company Ltd., Document Number: P316. This information also appears in ISO/IEC 10646, Annex P.

Returns:
a Unicode string from the bytes message stream

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of bytes stream has been reached.
MessageNotReadableException - if the message is in write-only mode.

public int readBytes(byte[] value) throws JMSException
byte
value

value -1

value
return -1

Throws JMSException: JMS

Throws MessageNotReadableException:

readBytes

int readBytes(byte[] value)
throws JMSException

Reads a byte array from the bytes message stream.

If the length of array value is less than the number of bytes remaining to be read from the stream, the array should be filled. A subsequent call reads the next increment, and so on.

If the number of bytes remaining in the stream is less than the length of array value, the bytes should be read into the array. The return value of the total number of bytes read will be less than the length of the array, indicating that there are no more bytes left to be read from the stream. The next read of the stream returns -1.

Parameters: value - the buffer into which the data is read

Returns: the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached

Throws: JMSException - if the JMS provider fails to read the message due to some internal error.

MessageNotReadableException - if the message is in write-only mode.
public int readBytes(byte[] value, int length) throws JMSException

value

value -1

length

length

value

Throws JMSException: JMS

Throws MessageNotReadableException:

readBytes

int readBytes(byte[] value,

int length)

throws JMSException

Reads a portion of the bytes message stream.

If the length of array value is less than the number of bytes remaining to be read from the stream, the array should be filled. A subsequent call reads the next increment, and so on.

If the number of bytes remaining in the stream is less than the length of array value, the bytes should be read into the array. The return value of the total number of bytes read will be less than the length of the array, indicating that there are no more bytes left to be read from the stream. The next read of the stream returns -1.

If length is negative, or length is greater than the length of the array value, then an IndexOutOfBoundsException is thrown. No bytes will be
read from the stream for this exception case.

Parameters:
value - the buffer into which the data is read
length - the number of bytes to read; must be less than or equal to value.length

Returns:
the total number of bytes read into the buffer, or -1 if there is no more data because the end of the stream has been reached

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageNotReadableException - if the message is in write-only mode.

---

public void writeBoolean(boolean value) throws JMSException

<table>
<thead>
<tr>
<th>boolean 1-byte</th>
<th>true (byte)1</th>
<th>false (byte)0</th>
</tr>
</thead>
</table>

Parameters:
value - the boolean value to be written

Throws: JMSException: JMS
MessageNotWriteableException: JMS

writeBoolean

void writeBoolean(boolean value)
throws JMSException

Writes a boolean to the bytes message stream as a 1-byte value. The value true is written as the value (byte)1; the value false is written as the value (byte)0.

Parameters:
value - the boolean value to be written

Throws: JMSException - if the JMS provider fails to write the message due
to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeByte(byte value) throws JMSException
byte value
Throws JMSException: JMS
Throws MessageNotWriteableException:

writeByte

void writeByte(byte value)
throws JMSException

Writes a byte to the bytes message stream as a 1-byte value.

Parameters:
value - the byte value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeShort(short value) throws JMSException
short value
Throws JMSException: JMS
Throws MessageNotWriteableException:

writeShort

void writeShort(short value)
Throws: \texttt{JMSException} - if the JMS provider fails to write the message due to some internal error.
\texttt{MessageNotWriteableException} - if the message is in read-only mode.

\textbf{Parameters:}
\begin{itemize}
  \item \texttt{value} - the short to be written
\end{itemize}

\textbf{Throws:}
\begin{itemize}
  \item \texttt{JMSException} - if the JMS provider fails to write the message due to some internal error.
  \item \texttt{MessageNotWriteableException} - if the message is in read-only mode.
\end{itemize}

\begin{verbatim}
public void writeChar(char value) throws JMSException
char 2-byte
value char
Throws JMSException: JMS
Throws MessageNotWriteableException:
\end{verbatim}

\textbf{writeChar}

void \texttt{writeChar(char value)}
throws \texttt{JMSException}

Writes a \texttt{char} to the bytes message stream as a 2-byte value, high byte first.

\textbf{Parameters:}
\begin{itemize}
  \item \texttt{value} - the \texttt{char} value to be written
\end{itemize}

\textbf{Throws:}
\begin{itemize}
  \item \texttt{JMSException} - if the JMS provider fails to write the message due to some internal error.
  \item \texttt{MessageNotWriteableException} - if the message is in read-only mode.
\end{itemize}

\begin{verbatim}
public void writeInt(int value) throws JMSException
\end{verbatim}
writeInt

`void writeInt(int value)`

Throws `JMSException`: JMS

**Parameter:**
- `value` - the int to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

writeLong

`void writeLong(long value)`

Throws `JMSException`: JMS

**Parameter:**
- `value` - the long to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

Writes a long to the bytes message stream as eight bytes, high byte first.
Parameters:

value - the long to be written

Throws:

- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```
public void writeFloat(float value) throws JMSException
```

Parameters:

value - the float value to be written

Throws:

- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```
public void writeDouble(double value) throws JMSException
```

Converts the `double` argument to an `int` using the `doubleToLongBits` method in class `Double`, and then writes that `int` value to the bytes message stream as a 8-byte quantity, high byte first.
byte

void writeDouble(double value)
throws JMSException

Converts the double argument to a long using the doubleToLongBits method in class double, and then writes that long value to the bytes message stream as an 8-byte quantity, high byte first.

Parameters:
value - the double value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeUTF(String value) throws JMSException

UTF-8

UTF-8 "File System Safe UCS Transformation Format (FSS_UTF)"X/Open Preliminary Specification
X/Open Company Ltd.P316 ISO/IEC 10646Annex P

void writeUTF(String value)
throws JMSException

Throws:
JMSException - JMS
MessageNotWriteableException -
**writeUTF**

```java
void writeUTF(String value)
throws JMSException
```

Writes a string to the bytes message stream using UTF-8 encoding in a machine-independent manner.

For more information on the UTF-8 format, see "File System Safe UCS Transformation Format (FSS_UTF)", X/Open Preliminary Specification, X/Open Company Ltd., Document Number: P316. This information also appears in ISO/IEC 10646, Annex P.

**Parameters:**
- `value` - the String value to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

---

**public void writeBytes(byte[] value) throws JMSException**

```java
byte value
```

Writes a byte array to the bytes message stream.

**Parameters:**
- `value` - the byte array to be written

**Throws:**
- `JMSException` - JMS
- `MessageNotWriteableException` -
`JMSException` - if the JMS provider fails to write the message due to some internal error.
`MessageNotWriteableException` - if the message is in read-only mode.

```java
public void writeBytes(byte[] value, int offset, int length) throws JMSException;
```

**Parameters:**
- `value` - the byte array value to be written
- `offset` - the initial offset within the byte array
- `length` - the number of bytes to use

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```java
public void writeObject(Object value) throws JMSException
```

`writeBytes` method writes a portion of a byte array to the bytes message stream.

**Parameters:**
- `value` - the byte array value to be written
- `offset` - the initial offset within the byte array
- `length` - the number of bytes to use

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
**writeObject**

```java
void writeObject(Object value)

throws JMSException
```

Writes an object to the bytes message stream.

This method works only for the objectified primitive object types (Integer, Double, Long ...), String objects, and byte arrays.

**Parameters:**

- `value` - the object in the Java programming language ("Java object") to be written; it must not be null

**Throws:**

- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageFormatException` - if the object is of an invalid type.
- `MessageNotWriteableException` - if the message is in read-only mode.
- `NullPointerException` - if the parameter `value` is null.

---

**public void reset() throws JMSException**

```java
void reset()

throws JMSException
```

**Throws:**

- `JMSException` - JMS
- `MessageFormatException`:
void reset() throws JMSException

Puts the message body in read-only mode and repositions the stream of bytes to the beginning.

**Throws:**
- `JMSException` - if the JMS provider fails to reset the message due to some internal error.
- `MessageFormatException` - if the message has an invalid format.
public final class ByteWrapperHolder extends Object implements Holder

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte</td>
<td>value</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteWrapperHolder()</td>
</tr>
<tr>
<td>ByteWrapperHolder(Byte mybyte)</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object: clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>
value

public Byte value

Constructor Detail

public ByteWrapperHolder()

ByteWrapperHolder

public ByteWrapperHolder()

public ByteWrapperHolder(Byte mybyte)

ByteWrapperHolder

public ByteWrapperHolder(Byte mybyte)
PS:
javax.xml.rpc.holders  Class CalendarHolder

java.lang.Object
  ↓ javax.xml.rpc.holders.CalendarHolder

All Implemented Interfaces:
  Holder

public final class CalendarHolder
  extends Object
  implements Holder

Implements: Holder

Field Summary

| Calendar | value |

Constructor Summary

CalendarHolder()

CalendarHolder(Calendar myCalendar)

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Field Detail

value

public Calendar value

Constructor Detail

public CalendarHolder()

CalendarHolder

public CalendarHolder()

public CalendarHolder(java.util.Calendar myCalendar)

CalendarHolder

public CalendarHolder(Calendar myCalendar)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.rpc Interface Call

public interface Call

javax.xml.rpc.Call  javax.xml.rpc.Service  Call

Call  Call

version  1.0

The javax.xml.rpc.Call interface provides support for the dynamic invocation of a service endpoint. The javax.xml.rpc.Service interface acts as a factory for the creation of call instances.

Once a Call instance is created, various setter and getter methods may be used to configure this Call instance.

Version:  1.0
Author:  Rahul Sharma

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>ENCODINGSTYLE_URI_PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard property for encoding Style: Encoding style specified as a namespace URI.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>OPERATION_STYLE_PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard property for operation style.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>PASSWORD_PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard property: Password for authentication Type: java.lang.String</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>SESSION_MAINTAIN_PROPERTY</th>
</tr>
</thead>
</table>
**static String**

Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint.

**static String SOAPACTION_URI_PROPERTY**

Standard property for SOAPAction.

**static String SOAPACTION_USE_PROPERTY**

Standard property for SOAPAction.

**static String USERNAME_PROPERTY**

Standard property: User name for authentication

Type: java.lang.String

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addParameter(String paramName, QName xmlType, Class javaType, ParameterMode parameterMode)</td>
<td>Adds a parameter type and mode for a specific operation.</td>
</tr>
<tr>
<td>QName getOperationName()</td>
<td>Gets the name of the operation to be invoked using this Call instance.</td>
</tr>
<tr>
<td>Map getOutputParams()</td>
<td>Returns a Map of {name, value} for the output parameters of the last invoked operation.</td>
</tr>
<tr>
<td>List getOutputValues()</td>
<td>Returns a List values for the output parameters of the last invoked operation.</td>
</tr>
<tr>
<td>QName getParameterTypeByName(String paramName)</td>
<td>Gets the XML type of a parameter by name</td>
</tr>
<tr>
<td>QName getPortTypeName()</td>
<td>Gets the qualified name of the port type.</td>
</tr>
<tr>
<td>Object getProperty(String name)</td>
<td>Gets the value of a named property.</td>
</tr>
<tr>
<td>Iterator getPropertyNames()</td>
<td>Gets the names of configurable properties supported by</td>
</tr>
</tbody>
</table>
### Call Object

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>QName getReturnType()</code></td>
<td>Gets the return type for a specific operation</td>
</tr>
<tr>
<td><code>String getTargetEndpointAddress()</code></td>
<td>Gets the address of a target service endpoint.</td>
</tr>
<tr>
<td><code>Object invoke(Object[] inputParams)</code></td>
<td>Invokes a specific operation using a synchronous request-response interaction mode.</td>
</tr>
<tr>
<td><code>Object invoke(QName operationName, Object[] inputParams)</code></td>
<td>Invokes a specific operation using a synchronous request-response interaction mode.</td>
</tr>
<tr>
<td><code>void invokeOneWay(Object[] inputParams)</code></td>
<td>Invokes a remote method using the one-way interaction mode.</td>
</tr>
<tr>
<td><code>boolean isParameterAndReturnSpecRequired(QName operationName)</code></td>
<td>Indicates whether addParameter and setReturn methods are to be invoked to specify the parameter and return type specification for a specific operation.</td>
</tr>
<tr>
<td><code>void removeAllParameters()</code></td>
<td>Removes all specified parameters from this call instance.</td>
</tr>
<tr>
<td><code>void removeProperty(String name)</code></td>
<td>Removes a named property.</td>
</tr>
<tr>
<td><code>void setOperationName(QName operationName)</code></td>
<td>Sets the name of the operation to be invoked using this call instance.</td>
</tr>
<tr>
<td><code>void setPortTypeName(QName portType)</code></td>
<td>Sets the qualified name of the port type.</td>
</tr>
<tr>
<td><code>void setProperty(String name, Object value)</code></td>
<td>Sets the value for a named property.</td>
</tr>
<tr>
<td><code>void setReturnType(QName xmlType)</code></td>
<td>Sets the return type for a specific operation.</td>
</tr>
<tr>
<td><code>void setReturnType(QName xmlType, Class javaType)</code></td>
<td>Sets the return type for a specific operation.</td>
</tr>
<tr>
<td><code>void setTargetEndpointAddress(String address)</code></td>
<td>Sets the address of the target service endpoint.</td>
</tr>
</tbody>
</table>
USERNAMEPROPERTY

static final String USERNAMEPROPERTY

Standard property: User name for authentication

Type: java.lang.String

See Also:
Constant Field Values

PASSWORDPROPERTY

static final String PASSWORDPROPERTY

Standard property: Password for authentication

Type: java.lang.String

See Also:
Constant Field Values

OPERATIONSTYLEPROPERTY

static final String OPERATIONSTYLEPROPERTY

Standard property for operation style. This property is set to "rpc" if
the operation style is rpc; "document" if the operation style is document.

Type: java.lang.String

See Also: Constant Field Values

-------------------------------

SOAPACTION_USE_PROPERTY

static final String SOAPACTION_USE_PROPERTY

Standard property for SOAPAction. This boolean property indicates whether or not SOAPAction is to be used. The default value of this property is false indicating that the SOAPAction is not used.

Type: java.lang.Boolean

See Also: Constant Field Values

-------------------------------

SOAPACTION_URIPROPERTY

static final String SOAPACTION_URIPROPERTY

Standard property for SOAPAction. Indicates the SOAPAction URI if the javax.xml.rpc.soap.http.soapaction.use property is set to true.

Type: java.lang.String

See Also: Constant Field Values
ENCODINGSTYLE_URI_PROPERTY

static final String ENCODINGSTYLE_URI_PROPERTY

Standard property for encoding Style: Encoding style specified as a namespace URI. The default value is the SOAP 1.1 encoding http://schemas.xmlsoap.org/soap/encoding/

Type: java.lang.String

See Also:
Constant Field Values

---

SESSION_MAINTAIN_PROPERTY

static final String SESSION_MAINTAIN_PROPERTY

Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint. If this property is set to true, the service client indicates that it wants the session to be maintained. If set to false, the session is not maintained. The default value for this property is false.

Type: java.lang.Boolean

See Also:
Constant Field Values

---

Method Detail

public boolean isParameterAndReturnSpecRequired(javax.xml.namespace.QName operationName)
addParameter  setReturnType
operationName
return  Call  addParameter  setReturnType  true  false
Threws  IllegalArgumentException:

isParameterAndReturnSpecRequired

boolean isParameterAndReturnSpecRequired(QName operationName)

Indicates whether addParameter and setReturnType methods are to be invoked to specify the parameter and return type specification for a specific operation.

Parameters:
operationName - Qualified name of the operation

Returns:
Returns true if the Call implementation class requires addParameter and setReturnType to be invoked in the client code for the specified operation. This method returns false otherwise.

Throws:
IllegalArgumentException - If invalid operation name is specified

public void addParameter(String paramName, javax.xml.namespace.QName xmlType, ParameterMode parameterMode)

invoke  addParameter

&l Call WSDL
&l paramName
&l xmlType
&l parameterMode

&l XML
ParameterMode.IN ParameterMode.OUT
ParameterMode.INOUT

Throws
javax.xml.rpc.JAXRPCException::
isParameterAndReturnSpecRequired  false
Throw IllegalArgumentException: XML
See also isParameterAndReturnSpecRequired

addParameter

void addParameter(String paramName, QName xmlType, ParameterMode parameterMode)

Adds a parameter type and mode for a specific operation. Note that the client code may not call any addParameter and setReturnType methods before calling the invoke method. In this case, the Call implementation class determines the parameter types by using reflection on parameters, using the WSDL description and configured type mapping registry.

Parameters:
- paramName - Name of the parameter
- xmlType - XML type of the parameter
- parameterMode - Mode of the parameter-whether ParameterMode.IN, ParameterMode.OUT, or ParameterMode.INOUT

Throws:
- javax.xml.rpc.JAXRPCException: - This exception may be thrown if the method isParameterAndReturnSpecRequired returns false for this operation.
- IllegalArgumentException - If any illegal parameter name or XML type is specified

See Also:
- isParameterAndReturnSpecRequired(javax.xml.namespace.QName)

public void addParameter(String paramName, javax.xml.namespace.QName xmlType, Class<T> javaType, ParameterMode parameterMode)

OUT INOUT Java

paramName
xmlType XML
**addParameter**

```java
void addParameter(String paramName,
                  QName xmlType,
                  Class javaType,
                  ParameterMode parameterMode)
```

Adds a parameter type and mode for a specific operation. This method is used to specify the Java type for either OUT or INOUT parameters.

**Parameters:**
- `paramName` - Name of the parameter
- `xmlType` - XML type of the parameter
- `javaType` - Java class of the parameter
- `parameterMode` - Mode of the parameter-whether ParameterMode.IN, OUT or INOUT

**Throws:**
- `JAXRPCException` -
  - This exception may be thrown if this method is invoked when the method `isParameterAndReturnSpecRequired` returns `false`.
  - If specified XML type and Java type mapping is not valid. For example, `TypeMappingRegistry` has no serializers for this mapping.
- `IllegalArgumentException` - If any illegal parameter name or
XML type is specified

UnsupportedOperationException - If this method is not supported

See Also:

isParameterAndReturnSpecRequired(javax.xml.namespace.QName)

---

public javax.xml.namespace.QName
getParameterTypeByName(String paramName)

XML

   paramName
   return

getParameterTypeByName

QName getParameterTypeByName(String paramName)

   Gets the XML type of a parameter by name

Parameters:

   paramName - Name of the parameter

Returns:

   Returns XML type for the specified parameter

---

public void setReturnType(javax.xml.namespace.QName xmlType)

setParameterType(null)   Call

xmlType   setReturnType(QName xmlType)

Throws JAXRPCException: isParameterAndReturnSpecRequired

false

Throws IllegalArgumentException: XML

---

setReturnType

void setReturnType(QName xmlType)
Sets the return type for a specific operation. Invoking `setReturnType(null)` removes the return type for this Call object.

**Parameters:**
- `xmlType` - XML data type of the return value

**Throws:**
- `JAXRPCException` - This exception may be thrown when the method `isParameterAndReturnSpecRequired` returns `false`.
- `IllegalArgumentException` - If an illegal XML type is specified

```java
public void setReturnType(javax.xml.namespace.QName xmlType, Class<T> javaType)
```

**type**
- XML

**javaType**
- Java

**Throws**
- `isParameterAndReturnSpecRequired` - `false`
- `TypeMapping` - XML, Java

**Throws**
- `UnsupportedOperationException`: 

**Throws**
- `IllegalArgumentException`: 

`setReturnType` 

```java
void setReturnType(QName xmlType,
                  Class javaType)
```

Sets the return type for a specific operation.

**Parameters:**
- `type` - XML data type of the return value
- `javaType` - Java Class of the return value

**Throws:**
- `JAXRPCException` - 
  - This exception may be thrown if this method is invoked when the method `isParameterAndReturnSpecRequired`
returns false.

- If XML type and Java type cannot be mapped using the standard type mapping or TypeMapping registry

  `UnsupportedOperationException` - If this method is not supported

  `IllegalArgumentException` - If an illegal XML type is specified

---

**public javax.xml.namespace.QName getReturnType()**

```java
return XML
```

**getReturnType**

```java
QName getReturnType()
```

Gets the return type for a specific operation

**Returns:**

- Returns the XML type for the return value

---

**public void removeAllParameters()**

```java
Call setReturnType(null)
```

**Throws:**

- `JAXRPCException`: `isParameterAndReturnSpecRequired`

---

**removeAllParameters**

```java
void removeAllParameters()
```

Removes all specified parameters from this `Call` instance. Note that this method removes only the parameters and not the return type. The `setReturnType(null)` is used to remove the return type.

**Throws:**

- `JAXRPCException` - This exception may be thrown if this method
is called when the method isParameterAndReturnSpecRequired returns false for this Call's operation.

```java
public javax.xml.namespace.QName getOperationName()
    Call
    return

getOperationName
QName getOperationName()

    Gets the name of the operation to be invoked using this call instance.

    Returns:
    Qualified name of the operation

public void setOperationName(javax.xml.namespace.QName operationName)
    Call
    operationName    Call    QName

setOperationName
void setOperationName(QName operationName)

    Sets the name of the operation to be invoked using this call instance.

    Parameters:
    operationName - QName of the operation to be invoked using the Call instance
```
public javax.xml.namespace.QName getPortTypeName()

    return

getPortTypeName
QName getPortTypeName()

    Gets the qualified name of the port type.

    Returns:
    Qualified name of the port type

public void setPortTypeName(javax.xml.namespace.QName portType)

    portType

setPortTypeName
void setPortTypeName(QName portType)

    Sets the qualified name of the port type.

    Parameters:
    portType - Qualified name of the port type

public void setTargetEndpointAddress(String address)

    address

setTargetEndpointAddress(String address)

    Call

    address URI
**setTargetEndpointAddress**

```java
void setTargetEndpointAddress(String address)
```

Sets the address of the target service endpoint. This address must correspond to the transport specified in the binding for this call instance.

**Parameters:**
- `address` - Address of the target service endpoint; specified as an URI

---

**public String getTargetEndpointAddress()**

```java
return URI
```

---

**getTargetEndpointAddress**

```java
String getTargetEndpointAddress()
```

Gets the address of a target service endpoint.

**Returns:**
- Address of the target service endpoint as an URI

---

**public void setProperty(String name, Object value)**

**JAX-RPC**

```java
name
value
```

**JAXRPCException**: Call

**Throws**: Call

- Call
setProperty

void setProperty(String name,
Object value)

Sets the value for a named property. JAX-RPC specification specifies a standard set of properties that may be passed to the Call.setProperty method.

Parameters:
- name - Name of the property
- value - Value of the property

Throws:
- JAXRPCException -
  - If an optional standard property name is specified, however this call implementation class does not support the configuration of this property.
  - If an invalid (or unsupported) property name is specified or if a value of mismatched property type is passed.
  - If there is any error in the configuration of a valid property.

public Object getProperty(String name)

name
return

Throws JAXRPCException:

getProperty

Object getProperty(String name)

Gets the value of a named property.

Parameters:
name - Name of the property

**Returns:**
Value of the named property

**Throws:**
[JAXRPCException](#) - if an invalid or unsupported property name is passed.

---

**public void removeProperty(String name)**

`name`

**return**
boolean true false

**Throws**
[JAXRPCException](#)

---

**removeProperty**

void removeProperty(String name)

Removes a named property.

**Parameters:**
name - Name of the property

**Throws:**
[JAXRPCException](#) - if an invalid or unsupported property name is passed.

---

**public java.util.Iterator<E> getPropertyNames()**

`Call`

**return**
Iterator

---

**getPropertyNames**

Iterator getPropertyNames()

Gets the names of configurable properties supported by this call.
object.

**Returns:**

Iterator for the property names

---

```java
public Object invoke(Object[] inputParams) throws java.rmi.RemoteException {
  Object[] --
  return null
  Throws java.rmi.RemoteException:
  throws SOAPFaultException: SOAP
  JAXRPCException:
  
  Throws
  • Call
  • inputParams addParameter WSDL
  •
}
```

**invoke**

```java
Object invoke(Object[] inputParams) throws RemoteException
```

Invokes a specific operation using a synchronous request-response interaction mode.

**Parameters:**

- **inputParams** - Object[]--Parameters for this invocation. This includes only the input params

**Returns:**

Returns the return value or null

**Throws:**

- `RemoteException` - if there is any error in the remote method invocation
- `SOAPFaultException` - Indicates a SOAP fault
- `JAXRPCException` -
- If there is an error in the configuration of the `Call` object
- If `inputParams` do not match the required parameter set (as specified through the `addParameter` invocations or in the corresponding WSDL)
- If parameters and return type are incorrectly specified

```java
public Object invoke(javax.xml.namespace.QName operationName, Object[] inputParams) throws java.rmi.RemoteException
-
    operationName QName
    inputParams Object[] --
    return null

Throws java.rmi.RemoteException:
Throws SOAPFaultException: SOAP
    JAXRPCException:

Throws
    Call

Throws
    inputParams addParameter WSDL
```

**invoke**

```java
Object invoke(QName operationName,
             Object[] inputParams)
throws RemoteException
```

Invokes a specific operation using a synchronous request-response interaction mode.

**Parameters:**
- `operationName` - QName of the operation
- `inputParams` - Object[]--Parameters for this invocation. This includes only the input params.

**Returns:**
Return value or null

Throws:
- RemoteException - if there is any error in the remote method invocation.
- SOAPFaultException - Indicates a SOAP fault
- JAXRPCException -
  - If there is an error in the configuration of the Call object
  - If inputParams do not match the required parameter set (as specified through the addParameter invocations or in the corresponding WSDL)
  - If parameters and return type are incorrectly specified

public void invokeOneWay(Object[] inputParams)
SOAP/HTTP  HTTP
JAXRPCException
  inputParams Object[] --
  Throws JAXRPCException: call void

void invokeOneWay(Object[] inputParams)

Invokes a remote method using the one-way interaction mode. The client thread does not normally block waiting for the completion of the server processing for this remote method invocation. When the protocol in use is SOAP/HTTP, this method should block until an HTTP response code has been received or an error occurs. This method must not throw any remote exceptions. This method may throw a JAXRPCException during the processing of the one-way remote call.

Parameters:
  inputParams - Object[]--Parameters for this invocation. This includes only the input params.

Throws:
  JAXRPCException - if there is an error in the configuration of the
call object (example: a non-void return type has been incorrectly specified for the one-way call) or if there is any error during the invocation of the one-way remote call

```java
public java.util.Map<K, V> getOutputParams()
    Map {name, value} Map return
    Map Call.invoke() Map
    Throws JAXRPCException: invoke
```

getOutputParams

```java
Map getOutputParams()
```

Returns a Map of {name, value} for the output parameters of the last invoked operation. The parameter names in the returned Map are of type java.lang.String.

Returns:
Map Output parameters for the last Call.invoke(). Empty Map is returned if there are no output parameters.

Throws:
JAXRPCException - If this method is invoked for a one-way operation or is invoked before any invoke method has been called.

```java
public java.util.List<E> getOutputValues()
    List return java.util.List
    List Call.invoke() List
    Throws JAXRPCException: invoke
```

getOutputValues

```java
List getOutputValues()
```
Returns a List values for the output parameters of the last invoked operation.

**Returns:**
java.util.List Values for the output parameters. An empty List is returned if there are no output values.

**Throws:**
JAXRPCException - If this method is invoked for a one-way operation or is invoked before any invoke method has been called.
javax.xml.registry  Interface CapabilityProfile

public interface CapabilityProfile

JAXR

Provides information about the capabilities of a JAXR provider.

Author:
    Farrukh S. Najmi

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>getCapabilityLevel()</td>
<td>Gets the capability level supported by the JAXR provider.</td>
</tr>
<tr>
<td>String</td>
<td>getVersion()</td>
<td>Gets the JAXR specification version supported by the JAXR provider.</td>
</tr>
</tbody>
</table>

Method Detail

public String getVersion() throws JAXRException
    JAXR  JAXR

    0

    return

    Throws  JAXRException:  JAXR
**getVersion**

```java
String getVersion() throws JAXRException
```

Gets the JAXR specification version supported by the JAXR provider.

**Capability Level: 0**

**Returns:**
the specification version

**Throws:**
  * JAXRException - If the JAXR provider encounters an internal error

---

**public int getCapabilityLevel() throws JAXRException**

**JAXR**

```java
0
```

```java
    return
```

**Throws**
  * JAXRException: JAXR

---

**getCapabilityLevel**

```java
int getCapabilityLevel() throws JAXRException
```

Gets the capability level supported by the JAXR provider.

**Capability Level: 0**

**Returns:**
the capability level

**Throws:**
  * JAXRException - If the JAXR provider encounters an internal error
error

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence  **Enum CascadeType**

**java.lang.Object**  
- **java.lang.Enum<CascadeType>**  
  - **javax.persistence.CascadeType**

**All Implemented Interfaces:**  
** Serializable, Comparable<CascadeType>**

---

```java
public enum CascadeType
    extends Enum<CascadeType>

Extends: Enum<E>
```

- `cascade=ALL`  
- `cascade={PERSIST, MERGE, REMOVE, REFRESH}`  

since  
Java Persistence 1.0

Defines the set of cascadable operations that are propagated to the associated entity. The value `cascade=ALL` is equivalent to `cascade={PERSIST, MERGE, REMOVE, REFRESH}`.

**Since:**  
Java Persistence 1.0

---

<table>
<thead>
<tr>
<th><strong>Enum Constant Summary</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALL</strong></td>
</tr>
<tr>
<td>Cascade all operations</td>
</tr>
<tr>
<td><strong>MERGE</strong></td>
</tr>
<tr>
<td>Cascade merge operation</td>
</tr>
<tr>
<td><strong>PERSIST</strong></td>
</tr>
<tr>
<td>Cascade persist operation</td>
</tr>
<tr>
<td><strong>REFRESH</strong></td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `java.lang.Enum`
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### Methods inherited from class `java.lang.Object`
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Enum Constant Detail

**ALL**

```java
public static final CascadeType ALL

    Cascade all operations
```

**PERSIST**
public static final CascadeType PERSIST

Cascade persist operation

MERGE

public static final CascadeType MERGE

Cascade merge operation

REMOVE

public static final CascadeType REMOVE

Cascade remove operation

REFRESH

public static final CascadeType REFRESH

Cascade refresh operation

Method Detail

final public static CascadeType[] values()

values

public static final CascadeType[] values()
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(CascadeType c : CascadeType.values())
    System.out.println(c);
```

**Returns:**
an array containing the constants of this enum type, in the order they're declared

---

```java
public static CascadeType valueOf(String name)
```

**valueOf**

```java
public static CascadeType valueOf(String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
name - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**
`IllegalArgumentException` - if this enum type has no constant with the specified name
PS:
javax.faces.convert Class CharacterConverter

java.lang.Object  
  javax.faces.convert.CharacterConverter

All Implemented Interfaces:
    Converter

public class CharacterConverter
  extends Object
  implements Converter

Implements: Converter

java.lang.Character char Converter

Converter implementation for java.lang.Character (and char primitive) values.

---

## Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>CHARACTER_ID</td>
<td>The message identifier of the <a href="https://javaee.specs.java.net/jfapi/">FacesMessage</a> to be created if the conversion to Character fails.</td>
</tr>
<tr>
<td>static</td>
<td>CONVERTER_ID</td>
<td>The standard converter id for this converter.</td>
</tr>
<tr>
<td>static</td>
<td>STRING_ID</td>
<td>The message identifier of the <a href="https://javaee.specs.java.net/jfapi/">FacesMessage</a> to be created if the conversion of the Character value to String fails.</td>
</tr>
</tbody>
</table>
Constructor Summary

CharacterConverter()

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAsObject</strong>(FacesContext context, UIComponent component, String value)</td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><strong>getAsString</strong>(FacesContext context, UIComponent component, Object value)</td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

CONVERTER_ID

public static final String CONVERTER_ID

The standard converter id for this converter.

See Also:

Constant Field Values
CHARACTER_ID

public static final String CHARACTER_ID

The message identifier of the FacesMessage to be created if the conversion to Character fails. The message format string for this message may optionally include the following placeholders:

- \{0\} replaced by the unconverted value.
- \{1\} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Character value to String fails. The message format string for this message may optionally include the following placeholders:

- \{0\} replaced by the unconverted value.
- \{1\} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail
public CharacterConverter()

CharacterConverter

public CharacterConverter()

### Method Detail

**public Object getAsObject(FacesContext context, UIComponent component, String value)**

Throws: ConverterException: NullPointerException

Throws: NullPointerException: NullPointerException context component null

---

**getAsObject**

public Object getAsObject(FacesContext context, UIComponent component, String value)

**Description copied from interface:** Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Specified by:**

getAsObject in interface Converter

**Parameters:**

- context - FacesContext for the request being processed
- component - UIComponent with which this model object value is associated
- value - String value to be converted (may be null)

**Returns:**

null if the value to convert is null, otherwise the result of the
Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null

public String getAsString(FacesContext context, UIComponent component, Object value)

Throws
ConverterException: NullPointerException

Throws
NullPointerException: NullPointerException

description

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
performed

NullPointerException - if context or component is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream.events Interface Characters

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface Characters
extends XMLEvent

Implements: XMLEvent

Characters  Characters  CData  Characters
IgnorableWhitespace  false

Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants
ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT

This describes the interface to Characters events. All text events get reported as Characters events. Content, CData and whitespace are all reported as Characters events. IgnorableWhitespace, in most cases, will be set to false unless an element declaration of element content is present for the current element.

Version:
   1.0

Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getData()</code></td>
<td>Get the character data of this event</td>
</tr>
<tr>
<td><code>boolean isCData()</code></td>
<td>Returns true if this is a CData section.</td>
</tr>
<tr>
<td><code>boolean isIgnorableWhiteSpace()</code></td>
<td>Return true if this is ignorableWhiteSpace.</td>
</tr>
<tr>
<td><code>boolean isWhiteSpace()</code></td>
<td>Returns true if this set of Characters is all whitespace.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.stream.events.XMLEvent`

- `asCharacters`, `asEndElement`, `asStartElement`, `getEventType`, `getLocation`, `getSchemaType`, `isAttribute`, `isCharacters`, `isEndDocument`, `isEndElement`, `isEntityReference`, `isNamespace`, `isProcessingInstruction`, `isStartDocument`, `isStartElement`, `writeAsEncodedUnicode`

Method Detail

public `String getData()`

`getData`

`String getData()`

Get the character data of this event

public `boolean isWhiteSpace()`

`isWhiteSpace`  
Characters  true CHARACTERS
CHARACTERS

isWhiteSpace

boolean isWhiteSpace()

Returns true if this set of Characters is all whitespace. Whitespace inside a document is reported as CHARACTERS. This method allows checking of CHARACTERS events to see if they are composed of only whitespace characters.

public boolean isCData()

boolean isCData()

Returns true if this is a CData section. If this event is CData its event type will be CDATA If javax.xml.stream.isCoalescing is set to true CDATA Sections that are surrounded by non CDATA characters will be reported as a single Characters event. This method will return false in this case.

public boolean isIgnorableWhiteSpace()

boolean isIgnorableWhiteSpace()

Returns true if this set of Characters is all whitespace. Whitespace inside a document is reported as CHARACTERS. This method allows checking of CHARACTERS events to see if they are composed of only whitespace characters.
boolean isIgnorableWhiteSpace()

    Return true if this is ignorableWhiteSpace. If this event is ignorableWhiteSpace its event type will be SPACE.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel  Interface Classification

All Superinterfaces:
  ExtensibleObject, RegistryObject

public interface Classification

extends RegistryObject

Implements: RegistryObject

Classification  RegistryObject  RegistryObject  0
Classification  RegistryObject  Organization
RegistryObject  3  Classification  RegistryObject
addClassification  RegistryObject  Classification

  1  Classification  ClassificationScheme  RegistryObject

    1. ClassificationScheme
JAXR

1. JAXR (Internal Taxonomy)  JAXR
2. JAXR (External Taxonomy)  JAXR
Classification Classification

Classification ClassificationScheme RegistryObject
ClassificationScheme Classification ClassificationScheme
Classification Classification Classification ClassificationScheme
Classification
Classification

Classification RegistryObject Classification
Classification Classification setConcept Classification
Concept setClassificationScheme Classification
Concept ClassificationScheme

Classification

2 Concept Classification NAICS Organization
Book Publisher ?Book Publishers? NAICS
ClassificationScheme Concept Concept Concept

2. Classification
Classification

Classification RegistryObject Classification
Classification setValue Classification
setClassificationScheme ClassificationScheme

Classification

3 Classification Classification NAICS Organization
Book Publisher JAXR NAICS Concept
Classification Book Publisher

3. Classification
4. Classification

See [javax.xml.registry.infomodel.RegistryObject](http://example.com), [javax.xml.registry.infomodel.Concept](http://example.com)

The Classification interface is used to classify RegistryObject instances. A RegistryObject may be classified along multiple dimensions by adding zero or more Classification instances to the RegistryObject. For example, an Organization may be classified by its industry, by the products it sells and by its geographical location. In this example the RegistryObject would have at least three Classification instances added to it. The RegistryObject interface provides several addClassification methods to allow a client to add Classification instances to a Registry Object.

Figure 1 shows how a Classification classifies a RegistryObject using a ClassificationScheme.

Figure 1. Using a ClassificationScheme to Classify an Object
**Internal vs. External Taxonomies**

A taxonomy may be represented within a JAXR provider in one of the following ways:

1. The taxonomy elements and their structural relationship with each other are available within the JAXR provider. This case is referred to as *Internal Taxonomy* since the structure of the taxonomy is available internal to the JAXR provider.
2. The taxonomy elements and their structural relationship with each other is represented somewhere external to the JAXR provider. This case is referred to as *External Taxonomy* since the structure of the taxonomy is not available to the JAXR provider.
Internal vs. External Classifications

The Classification interface allows the classification of RegistryObjects using a ClassificationScheme whether the ClassificationScheme represents an internal taxonomy or an external taxonomy. When a Classification instance uses a ClassificationScheme representing an internal taxonomy then it is referred to as an internal Classification. When a Classification instance uses a ClassificationScheme representing an external taxonomy then it is referred to as an external Classification.
Internal Classification

When a Classification instance is used to classify a RegistryObject using an internal taxonomy it is referred to as an internal Classification. A client must call the setConcept method on a Classification and define a reference to a Concept instance from the Classification instance in order for that Classification to use an internal taxonomy. It is not necessary for the client to call setClassificationScheme for internal Classifications since the classifying Concept already knows its root ClassificationScheme.

Example of Internal Classification

Figure 2 shows an example of internal classification using a Concept to represent a taxonomy element. The example classifies an Organization instance as a Book Publisher using the NAICS standard taxonomy available as an internal taxonomy. Note that the figure does not show all the Concepts between the Book Publishers node and the NAICS ClassificationScheme to save space. Had they been there they would have been linked together by the parent attribute of each Concept.

Figure 2. Example of Internal Classification
External Classification

When a Classification instance is used to classify a RegistryObject using an external taxonomy it is referred to as an external Classification. A client must call the setValue method on a Classification and define a unique value that logically represents a taxonomy element within the taxonomy whose structure is defined externally. It is necessary for the client call setClassificationScheme for external Classifications since there is no other way to infer the ClassificationScheme that represents the external taxonomy.

Example of External Classification

Figure 3 shows an example of external classification. The example uses the same scenario where a Classification classifies an Organization instance as a Book Publisher using the NAICS standard taxonomy. However, this time the structure of the NAICS taxonomy is not available internally to the JAXR provider and consequently there is no Concept instance. Instead, the name and value attributes of the Classification are used to pinpoint the Book Publisher's taxonomy element. Note that name is optional but value is required.

Figure 3. Example of External Classification
An Example of Multiple Classifications

The next example shows how a RegistryObject may be classified by multiple classification schemes. In this example, two internal ClassificationSchemes named Industry and Geography are used to classify several Organization RegistryObjects by their industry and Geography. In Figure 4, in order to save space and improve readability, the Classification instances are not explicitly shown but are implied as associations between the RegistryObjects (shaded leaf node) and the associated Concepts.

![Figure 4. Example of Multiple Classifications](image)

Author: Farrukh S. Najmi
See Also: RegistryObject, Concept

### Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClassificationScheme</td>
<td><code>getClassificationScheme()</code></td>
<td>Gets the ClassificationScheme that is used in classifying the object.</td>
</tr>
<tr>
<td>RegistryObject</td>
<td><code>get ClassifiedObject()</code></td>
<td>Gets the Object that is being classified.</td>
</tr>
<tr>
<td>Concept</td>
<td><code>getConcept()</code></td>
<td>Gets the Concept that is classifying the object.</td>
</tr>
<tr>
<td>String</td>
<td><code>getValue()</code></td>
<td>Gets the taxonomy value for this Classification.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>isExternal()</code></td>
<td>Returns true if this is an external classification.</td>
</tr>
<tr>
<td>void</td>
<td><code>setClassificationScheme(ClassificationScheme classificationScheme)</code></td>
<td>Sets the ClassificationScheme for this external classification.</td>
</tr>
<tr>
<td>void</td>
<td><code>set ClassifiedObject(RegistryObject classifiedObject)</code></td>
<td>Sets the classified Object for this external classification.</td>
</tr>
</tbody>
</table>
Sets the object that is being classified.

```java
void setConcept(Concept concept)
```
Sets the concept for this internal classification.

```java
void setValue(String value)
```
Sets the taxonomy value for this external classification.

Methods inherited from interface 
javax.xml.registry.infomodel.**RegistryObject**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addAssociation</td>
<td></td>
</tr>
<tr>
<td>addAssociations</td>
<td></td>
</tr>
<tr>
<td>addClassification</td>
<td></td>
</tr>
<tr>
<td>addClassifications</td>
<td></td>
</tr>
<tr>
<td>addExternalIdentifier</td>
<td></td>
</tr>
<tr>
<td>addExternalIdentifiers</td>
<td></td>
</tr>
<tr>
<td>addExternalLink</td>
<td></td>
</tr>
<tr>
<td>addExternalLinks</td>
<td></td>
</tr>
<tr>
<td>getAssociatedObjects</td>
<td></td>
</tr>
<tr>
<td>getAuditTrail</td>
<td></td>
</tr>
<tr>
<td>getClassifications</td>
<td></td>
</tr>
<tr>
<td>getDescription</td>
<td></td>
</tr>
<tr>
<td>getExternalIdentifiers</td>
<td></td>
</tr>
<tr>
<td>getExternalLinks</td>
<td></td>
</tr>
<tr>
<td>getKey</td>
<td></td>
</tr>
<tr>
<td>getLifeCycleManager</td>
<td></td>
</tr>
<tr>
<td>getName</td>
<td></td>
</tr>
<tr>
<td>getObjectType</td>
<td></td>
</tr>
<tr>
<td>getRegistryPackages</td>
<td></td>
</tr>
<tr>
<td>getSubmittingOrganization</td>
<td></td>
</tr>
<tr>
<td>removeAssociation</td>
<td></td>
</tr>
<tr>
<td>removeAssociations</td>
<td></td>
</tr>
<tr>
<td>removeClassification</td>
<td></td>
</tr>
<tr>
<td>removeClassifications</td>
<td></td>
</tr>
<tr>
<td>removeExternalIdentifier</td>
<td></td>
</tr>
<tr>
<td>removeExternalIdentifiers</td>
<td></td>
</tr>
<tr>
<td>removeExternalLink</td>
<td></td>
</tr>
<tr>
<td>removeExternalLinks</td>
<td></td>
</tr>
<tr>
<td>setAssociations</td>
<td></td>
</tr>
<tr>
<td>setClassifications</td>
<td></td>
</tr>
<tr>
<td>setDescription</td>
<td></td>
</tr>
<tr>
<td>setExternalIdentifiers</td>
<td></td>
</tr>
<tr>
<td>setExternalLinks</td>
<td></td>
</tr>
<tr>
<td>setKey</td>
<td></td>
</tr>
<tr>
<td>setName</td>
<td></td>
</tr>
<tr>
<td>toXML</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface 
javax.xml.registry.infomodel.**ExtensibleObject**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>addSlot</td>
<td></td>
</tr>
<tr>
<td>addSlots</td>
<td></td>
</tr>
<tr>
<td>getSlot</td>
<td></td>
</tr>
<tr>
<td>getSlots</td>
<td></td>
</tr>
<tr>
<td>removeSlot</td>
<td></td>
</tr>
<tr>
<td>removeSlots</td>
<td></td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public Concept getConcept() throws JAXRException
```

**Concept**

```java
0
```

  return Concept Classification null

  **Throws**

  JAXRException: JAXR

  supplierCardinality 0..*

  clientCardinality 0..*
getConcept

Concept getConcept() throws JAXRException

Gets the Concept that is classifying the object.

Capability Level: 0

Returns:
the Concept that is classifying the classified object. null if this is a external Classification

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setConcept(Concept concept) throws JAXRException
Classification Concept

setClassificationScheme

0

concept Concept

Throws JAXRException: JAXR

setConcept

void setConcept(Concept concept) throws JAXRException

Sets the concept for this internal classification.
This method should be used mutually exclusively with the setClassificationScheme method.

**Capability Level:** 0

**Parameters:**
- concept - the Concept that is classifying the classified object.

**Throws:**
- [JAXRException](#) - If the JAXR provider encounters an internal error

```java
public ClassificationScheme getClassificationScheme() throws JAXRException
```

ClassificationScheme Classification
Classification Concept getClassificationScheme

0

```
return Classification ClassificationScheme
```

**Throws**
- [JAXRException](#): JAXR

supplierCardinality

```
<ClassificationScheme>
0..1
```

label

```
classificationScheme
```

See also

[javax.xml.registry.infomodel.ClassificationScheme](#)

---

**getClassificationScheme**

ClassificationScheme getClassificationScheme() throws JAXRException

Gets the ClassificationScheme that is used in classifying the object. If the Classification is an internal Classification then this method should return the value returned by calling the getClassificationScheme method on the Concept representing the
public void setClassificationScheme(ClassificationScheme classificationScheme) throws JAXRException

Sets the ClassificationScheme for this external classification.

If this method is called then you must also call setValue method. This method should be used mutually exclusively with the setConcept method.

Capability Level: 0

Parameters:
classificationScheme - the ClassificationScheme used by this Classification

Throws:
    JAXRException - If the JAXR provider encounters an internal error

---

public String getValue() throws JAXRException
Classification

0

    return Classification Classification Concept

Throws: JAXRException: JAX

---

String getValue()

Throws: JAXRException

Gets the taxonomy value for this Classification.

Capability Level: 0

Returns:
    the value of the taxonomy element if external Classification; the
    value of the Concept representing the taxonomy element if internal Classification

Throws: JAXRException - If the JAXR provider encounters an internal error

---

public void setValue(String value) throws JAXRException
Classification

0
setValue

```java
void setValue(String value)
   throws JAXRException
```

Sets the taxonomy value for this external Classification.

**Capability Level: 0**

**Parameters:**

- `value` - the taxonomy value used by this external Classification

**Throws:**

- `JAXRException` - If the JAXR provider encounters an internal error

---

**public RegistryObject get ClassifiedObject() throws JAXRException**

**Object**

```java
return Classification RegistryObject
   throws JAXRException: JAXR
```

**get ClassifiedObject**

```java
RegistryObject get ClassifiedObject()
   throws JAXRException
```

Gets the Object that is being classified.

**Capability Level: 0**
Returns:
the RegistryObject that is classified by this Classification

Throws:
JAXRException - If the JAXR provider encounters an internal error

---

public void setClassifiedObject(RegistryObject classifiedObject) throws JAXRException

0

classifiedObject  Classification  RegistryObject

Throws:  JAXRException: JAXR

setClassifiedObject

void setClassifiedObject(RegistryObject classifiedObject) throws JAXRException

Sets the object that is being classified.

Capability Level: 0

Parameters:
classifiedObject - the RegistryObject that is classified by this Classification

Throws:
JAXRException - If the JAXR provider encounters an internal error

---

public boolean isExternal() throws JAXRException

Classification true

0
isExternal

boolean isExternal() throws JAXRException

Returns true if this is an external classification.

Capability Level: 0

Returns:
true if this is an external Classification; false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error
javax.xml.registry.infomodel **Interface ClassificationScheme**

**All Superinterfaces:**
ExtensibleObject, RegistryEntry, RegistryObject, Versionable

```java
class ClassificationScheme {
    extends RegistryEntry

    Implements RegistryEntry
}
```

**Classifications:**
ClassificationScheme, RegistryObject, ClassificationScheme
Dewey Decimal, North American Industry Classification System (NAICS)

1. Classification Scheme

A Classification Scheme instance represents a taxonomy that may be used to classify or categorize RegistryObject instances. A very common example of a classification scheme in science is the Classification of living things where living things are categorized in under a tree-like structure. Another example is the Dewey Decimal system used in libraries to categorize books and other publications. A common example in eBusiness is that of North American Industry Classification System (NAICS), which is a classification scheme used to classify businesses and services by the industry to which they belong.

Figure 1 shows how a Classification Scheme is used by a Classification to classify a RegistryObject.

**Figure 1. Using a Classification Scheme to Classify an Object**
### Field Summary

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Description</th>
</tr>
</thead>
</table>
| **static int** | **VALUE_TYPE_EMBEDDED_PATH**  
Each taxonomy value in ClassificationScheme embeds the full path from scheme to that Concept. |
| **static int** | **VALUE_TYPE_NON_UNIQUE**  
Taxonomy values in ClassificationScheme may be repeated within the same scheme. |
| **static int** | **VALUE_TYPE_UNIQUE**  
Each taxonomy value in ClassificationScheme is unique. |

### Fields inherited from interface javax.xml.registry.infomodel.RegistryEntry

- `STABILITY_DYNAMIC`, `STABILITY_DYNAMIC_COMPATIBLE`, `STABILITY_STATIC`, `STATUS_APPROVED`, `STATUS_DEPRECATED`, `STATUS_SUBMITTED`, `STATUS_WITHDRAWN`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>addChildConcept</code>(Concept concept)</td>
<td>Adds a child Concept.</td>
</tr>
<tr>
<td>void <code>addChildConcepts</code>(Collection concepts)</td>
<td>Adds a Collection of Concept children.</td>
</tr>
<tr>
<td>int <code>getChildConceptCount</code>()</td>
<td>Gets number of children.</td>
</tr>
<tr>
<td>Collection <code>getChildrenConcepts</code>()</td>
<td>Gets all immediate children Concepts.</td>
</tr>
<tr>
<td>Collection <code>getDescendantConcepts</code>()</td>
<td>Gets all descendant Concepts.</td>
</tr>
<tr>
<td>int <code>getValueType</code>()</td>
<td>Gets the value type for this object.</td>
</tr>
<tr>
<td>boolean <code>isExternal</code>()</td>
<td>Determines whether this ClassificationScheme is an external ClassificationScheme or an internal ClassificationScheme.</td>
</tr>
<tr>
<td>void <code>removeChildConcept</code>(Concept concept)</td>
<td></td>
</tr>
</tbody>
</table>
Removes a child Concept.

```java
void removeChildConcepts(Collection concepts)
```

Removes a Collection of children Concepts.

```java
void setValueType(int valueType)
```

Sets the value type for this object.

### Methods inherited from interface `javax.xml.registry.infomodel.RegistryEntry`

- `getExpiration`
- `getStability`
- `getStatus`
- `setExpiration`
- `setStability`

### Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`

- `addAssociation`
- `addAssociations`
- `addClassification`
- `addClassifications`
- `addExternalIdentifier`
- `addExternalIdentifiers`
- `addExternalLink`
- `addExternalLinks`
- `getAssociatedObjects`
- `getAuditTrail`
- `getClassifications`
- `getDescription`
- `getExternalIdentifiers`
- `getExternalLinks`
- `getKey`
- `getLifeCycleManager`
- `getName`
- `getObjectType`
- `getRegistryPackages`
- `getSubmittingOrganization`
- `removeAssociation`
- `removeAssociations`
- `removeClassification`
- `removeClassifications`
- `removeExternalIdentifier`
- `removeExternalIdentifiers`
- `removeExternalLink`
- `removeExternalLinks`
- `setAssociations`
- `setClassifications`
- `setDescription`
- `setExternalIdentifiers`
- `setExternalLinks`
- `setKey`
- `setName`
- `toXML`

### Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`

- `addSlot`
- `addSlots`
- `getSlot`
- `getSlots`
- `removeSlot`
- `removeSlots`

### Methods inherited from interface `javax.xml.registry.infomodel.Versionable`

- `getMajorVersion`
- `getMinorVersion`
- `getUserVersion`
- `setMajorVersion`
- `setMinorVersion`
- `setUserVersion`

### Field Detail
VALUE_TYPE_UNIQUE
static final int VALUE_TYPE_UNIQUE

Each taxonomy value in ClassificationScheme is unique.

See Also:
Constant Field Values

VALUE_TYPE_EMBEDDED_PATH
static final int VALUE_TYPE_EMBEDDED_PATH

Each taxonomy value in ClassificationScheme embeds the full path from scheme to that Concept. This also implies that each taxonomy value is unique.

See Also:
Constant Field Values

VALUE_TYPE_NON_UNIQUE
static final int VALUE_TYPE_NON_UNIQUE

Taxonomy values in ClassificationScheme may be repeated within the same scheme. However, two Concepts that have the same parent cannot have the same value.

See Also:
Constant Field Values
public void addChildConcept(Concept concept) throws JAXRException

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void addChildConcept(Concept concept) throws JAXRException</td>
</tr>
<tr>
<td>Concept</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

concept Concept
Throws JAXRException: JAXR

addChildConcept

void addChildConcept(Concept concept)
throws JAXRException

Add a child Concept.

Capability Level: 0

Parameters:
concept - the concept being added as a child of this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void addChildConcepts(java.util.Collection<E> concepts) throws JAXRException

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void addChildConcepts(java.util.Collection&lt;E&gt; concepts) throws JAXRException</td>
</tr>
<tr>
<td>Concept Collection</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

concepts Concept Collection
Throws JAXRException: JAXR
addChildConcepts

void addChildConcepts(Collection concepts)
throws JAXRException

Adds a Collection of Concept children.

Capability Level: 0

Parameters:
- concepts - the Collection of Concepts being added as a children of this object

Throws:
- JAXRException - If the JAXR provider encounters an internal error

public void removeChildConcept(Concept concept) throws JAXRException

concept

concept

Throws
- JAXRException: JAXR

removeChildConcept

void removeChildConcept(Concept concept)
throws JAXRException

Removes a child Concept.

Capability Level: 0

Parameters:
public void removeChildConcepts(java.util.Collection<E> concepts) throws JAXRException

Removes a Collection of children Concepts.

Parameters:

- concepts - the Collection of Concepts being removed as children Concepts of this object

Throws:

- JAXRException - If the JAXR provider encounters an internal error

public int getChildConceptCount() throws JAXRException
getChildConceptCount

```java
int getChildConceptCount() throws JAXRException
```

Gets number of children.

**Capability Level: 0**

**Returns:**
the number of children Concepts

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

public java.util.Collection<E> getChildrenConcepts() throws JAXRException

**getChildrenConcepts**

```java
Collection getChildrenConcepts() throws JAXRException
```

Gets all immediate children Concepts.

**See also**
- `javax.xml.registry.infomodel.Concept`
Capability Level: 0

Returns:
Collection of Concept instances. The Collection may be empty but not null.

Throws:
*JAXRException* - If the JAXR provider encounters an internal error

See Also:
*Concept*

```java
public java.util.Collection<E> getDescendantConcepts()
throws JAXRException

Concept

return Concept Collection Collection null

Throws JAXRException: JAXR

See also javax.xml.registry.infomodel.Concept
```

getDescendantConcepts

*Collection* getDescendantConcepts()

Gets all descendant Concepts.

Capability Level: 0

Returns:
Collection of Concept instances. The Collection may be empty but not null.

Throws:
*JAXRException* - If the JAXR provider encounters an internal error
public boolean isExternal() throws JAXRException
ClassificationScheme ClassificationScheme
ClassificationScheme

0

return ClassificationScheme true false
Throws JAXRException: JAXR

isExternal

boolean isExternal() throws JAXRException

Determines whether this ClassificationScheme is an external ClassificationScheme or an internal ClassificationScheme.

Capability Level: 0

Returns: true if this is an external ClassificationScheme; false otherwise
Throws: JAXRException - If the JAXR provider encounters an internal error

public int getValueType() throws JAXRException

1

return ClassificationScheme
Throws JAXRException: JAXR
getValueType

```java
int getValueType() throws JAXRException
```

Gets the value type for this object. The value type describes how taxonomy values are defined within the scheme.

**Capability Level: 1**

**Returns:**
- an integer constant that describes the type of values supported by this ClassificationScheme

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**See Also:**
- `VALUE_TYPE_UNIQUE`, `VALUE_TYPE_EMBEDDED_PATH`, `VALUE_TYPE_NON_UNIQUE`
void setValueType(int valueType)
    throws JAXRException

Sets the value type for this object.

**Capability Level: 1**

**Parameters:**
valueType - an integer constant that describes the type of values supported by this ClassificationScheme

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

**See Also:**
VALUE_TYPE_UNIQUE, VALUE_TYPE_EMBEDDED_PATH, VALUE_TYPE_NON_UNIQUE

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface ClassTransformer

PersistenceUnitInfo.addTransformer JVM
since Java Persistence 1.0

A persistence provider supplies an instance of this interface to the PersistenceUnitInfo.addTransformer method. The supplied transformer instance will get called to transform entity class files when they are loaded or redefined. The transformation occurs before the class is defined by the JVM.

Since: Java Persistence 1.0

### Method Summary

<table>
<thead>
<tr>
<th>transform(byte[], ClassLoader loader, String className, Class&lt;?&gt;, ClassBeingRedefined, ProtectionDomain protectionDomain, byte[] classfileBuffer)</th>
</tr>
</thead>
</table>
| Invoked when a class is being loaded or redefined.

### Method Detail

transform

byte[] transform(ClassLoader loader, String className, Class<?> classBeingRedefined,
ProtectionDomain protectionDomain,
byte[] classfileBuffer)
throws IllegalClassFormatException

Invoked when a class is being loaded or redefined. The implementation of this method may transform the supplied class file and return a new replacement class file.

Parameters:
- loader - The defining loader of the class to be transformed, may be null if the bootstrap loader
- className - The name of the class in the internal form of fully qualified class and interface names
- classBeingRedefined - If this is a redefine, the class being redefined, otherwise null
- protectionDomain - The protection domain of the class being defined or redefined
- classfileBuffer - The input byte buffer in class file format - must not be modified

Returns:
A well-formed class file buffer (the result of the transform), or null if no transform is performed

Throws:
- IllegalClassFormatException - If the input does not represent a well-formed class file
javax.enterprise.deploy.spi.status  Interface ClientConfiguration

All Superinterfaces:
    Serializable

public interface ClientConfiguration
    extends Serializable

Implements: java.io.Serializable

ClientConfiguration

The ClientConfiguration object installs, configures and executes an Application Client. This class resolves the settings for installing and running the application client.

---

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>void execute()</td>
</tr>
</tbody>
</table>

This method performs an exec and starts the application client running in another process.

---

Method Detail

public void execute() throws ClientExecuteException

Throws: ClientExecuteException:

execute
void execute() throws ClientExecuteException

This method performs an exec and starts the application client running in another process.

**Throws:**

*ClientExecuteException* - when the configuration is incomplete.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/legal/license.html).

---

PS:
javax.enterprise.deploy.spi.exceptions  Class  ClientExecuteException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.enterprise.deploy.spi.exceptions.ClientExecuteException

All Implemented Interfaces:
   Serializable

public class ClientExecuteException
   extends Exception

Extends: Throwable > Exception

This exception reports errors in setting up an application client for execution.

Author:
gfink
See Also:
   Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ClientExecuteException()</td>
<td>Creates new ClientExecuteException without detail message.</td>
</tr>
<tr>
<td>ClientExecuteException(String msg)</td>
<td>Constructs an ClientExecuteException with the specified detail message.</td>
</tr>
</tbody>
</table>
Method Summary

Methods inherited from class java.lang.Throwable

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace</td>
</tr>
<tr>
<td>getCause</td>
</tr>
<tr>
<td>getLocalizedMessage</td>
</tr>
<tr>
<td>getMessage</td>
</tr>
<tr>
<td>getStackTrace</td>
</tr>
<tr>
<td>initCause</td>
</tr>
<tr>
<td>printStackTrace</td>
</tr>
<tr>
<td>printStackTrace</td>
</tr>
<tr>
<td>printStackTrace</td>
</tr>
<tr>
<td>setStackTrace</td>
</tr>
<tr>
<td>toString</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone</td>
</tr>
<tr>
<td>equals</td>
</tr>
<tr>
<td>finalize</td>
</tr>
<tr>
<td>getClass</td>
</tr>
<tr>
<td>hashCode</td>
</tr>
<tr>
<td>notify</td>
</tr>
<tr>
<td>notifyAll</td>
</tr>
<tr>
<td>wait</td>
</tr>
<tr>
<td>wait</td>
</tr>
<tr>
<td>wait</td>
</tr>
</tbody>
</table>

Constructor Detail

public ClientExecuteException()

ClientExecuteException

public ClientExecuteException()

Creates new ClientExecuteException without detail message.

public ClientExecuteException(String msg)

ClientExecuteException

public ClientExecuteException(String msg)

Constructs an ClientExecuteException with the specified detail message.
Parameters:

msg - the detail message.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
**Class**

`javax.xml.bind.annotation.adapters.CollapsedStringAdapter`

`java.lang.Object`

\[ \downarrow \]

`javax.xml.bind.annotation.adapters.XmlAdapter<String,String>`

\[ \downarrow \]

`javax.xml.bind.annotation.adapters.CollapsedStringAdapter`

```java
public class CollapsedStringAdapter
extends XmlAdapter<String,String>
```

**Extends:** `XmlAdapter`

```xml
xs:token XmlAdapter
```

```
"CRLF SP
```

**since**

JAXB 2.0

Built-in `XmlAdapter` to handle `xs:token` and its derived types.

This adapter removes leading and trailing whitespaces, then truncate any sequence of tab, CR, LF, and SP by a single whitespace character `'`

**Since:**

JAXB 2.0

**Author:**

Kohsuke Kawaguchi

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CollapsedStringAdapter()</code></td>
</tr>
</tbody>
</table>

---
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isWhiteSpace(char ch)</td>
<td>returns true if the specified char is a white space character.</td>
</tr>
<tr>
<td>String marshal(String s)</td>
<td>No-op.</td>
</tr>
<tr>
<td>String unmarshal(String text)</td>
<td>Removes leading and trailing whitespaces of the string given as the parameter, then truncate any sequence of tab, CR, LF, and SP by a single whitespace character '.'.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public CollapsedStringAdapter()

CollapsedStringAdapter

public CollapsedStringAdapter()

Method Detail

public String unmarshal(String text)
" ")CRLF  SP

unmarshal

public String unmarshal(String text)
Removes leading and trailing whitespaces of the string given as the parameter, then truncate any sequence of tab, CR, LF, and SP by a single whitespace character ".

Specified by:

unmarshal in class 'XmlAdapter<String, String>

Parameters:

text - The value to be converted. Can be null.

public String marshal(String s)

marshal

public String marshal(String s)

No-op. Just return the same string given as the parameter.

Specified by:

marshal in class 'XmlAdapter<String, String>

Parameters:

s - The value to be converted. Can be null.

protected static boolean isWhiteSpace(char ch)

char true

isWhiteSpace

protected static boolean isWhiteSpace(char ch)

returns true if the specified char is a white space character.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type Column

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Column

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Column

1
@Column(name="DESC", nullable=false, length=512)
public String getDescription() { return description; }

2
@Column(name="DESC",
columnDefinition="CLOB NOT NULL",
table="EMP_DETAIL")
@Lob
public String getDescription() { return description; }

3
@Column(name="ORDER_COST",
updatable=false, precision=12, scale=2)
public BigDecimal getCost() { return cost; }

since Java Persistence 1.0

Is used to specify a mapped column for a persistent property or field. If no Column annotation is specified, the default values are applied.

Examples:

Example 1:
@Column(name="DESC", nullable=false, length=512)
public String getDescription() { return description; }
Example 2:
@Column(name="DESC",
        columnDefinition="CLOB NOT NULL",
        table="EMP_DETAIL")
@Lob
public String getDescription() {
    return description;
}

Example 3:
@Column(name="ORDER_COST",
        updatable=false,
        precision=12,
        scale=2)
public BigDecimal getCost() {
    return cost;
}

Since:
Java Persistence 1.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>columnDefinition</td>
<td>(Optional) The SQL fragment that is used when generating the DDL for the column.</td>
</tr>
<tr>
<td>boolean</td>
<td>insertable</td>
<td>(Optional) Whether the column is included in SQL INSERT statements generated by the persistence provider.</td>
</tr>
<tr>
<td>int</td>
<td>length</td>
<td>(Optional) The column length.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>(Optional) The name of the column.</td>
</tr>
<tr>
<td>boolean</td>
<td>nullable</td>
<td>(Optional) Whether the database column is nullable.</td>
</tr>
<tr>
<td>int</td>
<td>precision</td>
<td>(Optional) The precision for a decimal (exact numeric) column.</td>
</tr>
<tr>
<td>int</td>
<td>scale</td>
<td>(Optional) The scale for a decimal (exact numeric) column.</td>
</tr>
<tr>
<td>String</td>
<td>table</td>
<td>(Optional) The name of the table that contains the column.</td>
</tr>
<tr>
<td>boolean</td>
<td>unique</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>--------</td>
<td></td>
</tr>
<tr>
<td>Optional</td>
<td>Whether the property is a unique key.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th>updatable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional</td>
<td>Whether the column is included in SQL UPDATE statements generated by the persistence provider.</td>
</tr>
</tbody>
</table>

abstract public String name()

name

public abstract String name

(Optional) The name of the column. Defaults to the property or field name.

Default: ""

abstract public boolean unique()
UniqueConstraint

unique

public abstract boolean unique

(Optional) Whether the property is a unique key. This is a shortcut for the UniqueConstraint annotation at the table level and is useful for when the unique key constraint is only a single field. This constraint applies in addition to any constraint entailed by primary key mapping and to constraints specified at the table level.

Default: false
abstract public boolean nullable()
    null

nullable

public abstract boolean nullable
    (Optional) Whether the database column is nullable.
    Default: true

abstract public boolean insertable()
    SQL INSERT

insertable

public abstract boolean insertable
    (Optional) Whether the column is included in SQL INSERT statements generated by the persistence provider.
    Default: true

abstract public boolean updatable()
    SQL UPDATE

updatable

public abstract boolean updatable
Whether the column is included in SQL UPDATE statements generated by the persistence provider.

**Default:**
true

---

**abstract public String columnDefinition()**

**DDL SQL**

**SQL**

---

**columnDefinition**

**public abstract String columnDefinition**

(Optional) The SQL fragment that is used when generating the DDL for the column.

Defaults to the generated SQL to create a column of the inferred type.

**Default:**

""

---

**abstract public String table()**

---

**table**

**public abstract String table**

(Optional) The name of the table that contains the column. If absent the column is assumed to be in the primary table.
abstract public int length()

length

public abstract int length

(Optional) The column length. (Applies only if a string-valued column is used.)

Default:

255

abstract public int precision()

DDL

precision

public abstract int precision

(Optional) The precision for a decimal (exact numeric) column. (Applies only if a decimal column is used.) Value must be set by developer if used when generating the DDL for the column.

Default:

0

abstract public int scale()
public abstract int \texttt{scale}

(Optional) The scale for a decimal (exact numeric) column. (Applies only if a decimal column is used.)

\textbf{Default:}
\begin{itemize}
  \item 0
\end{itemize}
javax.persistence Annotation Type ColumnResult

@Target(value={})
@Retention(value=RUNTIME)
public @interface ColumnResult

Implements: Annotation
@Target(value={})
@Retention(value=RUNTIME)

SQL SELECT

Query q = em.createNativeQuery(
   "SELECT o.id AS order_id, " +
   "o.quantity AS order_quantity, " +
   "o.item AS order_item, " +
   "i.name AS item_name, " +
   "FROM Order o, Item i " +
   "WHERE (order_quantity > 25) AND (order_item = i.id)",
   "OrderResults");

@SqlResultSetMapping(name="OrderResults",
entities={
   @EntityResult(entityClass=com.acme.Order.class, fields={
      @FieldResult(name="id", column="order_id"),
      @FieldResult(name="quantity", column="order_quantity"),
      @FieldResult(name="item", column="order_item"))},
   columns={
      @ColumnResult(name="item_name")
   })

since Java Persistence 1.0

References name of a column in the SELECT clause of a SQL query - i.e., column alias, if applicable. Scalar result types can be included in the query result by specifying this annotation in the metadata.

Example:
Query q = em.createNativeQuery(
   "SELECT o.id AS order_id, " +
"o.quantity AS order_quantity, " +
"o.item AS order_item, " +
"i.name AS item_name, " +
"FROM Order o, Item i " +
"WHERE (order_quantity > 25) AND (order_item = i.id),
"OrderResults";

@SqlResultSetMapping(name="OrderResults",
    entities={
        @EntityResult(entityClass=com.acme.Order.class, fields={
            @FieldResult(name="id", column="order_id"),
            @FieldResult(name="quantity", column="order_quantity"
            @FieldResult(name="item", column="order_item"}}},
        columns={
            @ColumnResult(name="item_name")
        }
    )
)

Since:
Java Persistence 1.0

### Required Element Summary

| String name | The name of a column in the SELECT clause of a SQL query |

### Element Detail

abstract public String name()

SQL SELECT

name

public abstract String name

    The name of a column in the SELECT clause of a SQL query
javax.activation Class CommandInfo

java.lang.Object
  ↓ javax.activation.CommandInfo

public class CommandInfo extends Object

CommandMap CommandInfo bean
CommandInfo CommandMap CommandMap JavaBean

The CommandInfo class is used by CommandMap implementations to describe the results of command requests. It provides the requestor with both the verb requested, as well as an instance of the bean. There is also a method that will return the name of the class that implements the command but it is not guaranteed to return a valid value. The reason for this is to allow CommandMap implementations that subclass CommandInfo to provide special behavior. For example a CommandMap could dynamically generate JavaBeans. In this case, it might not be possible to create an object with all the correct state information solely from the class name.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommandInfo(String verb, String className)</td>
<td>The Constructor for CommandInfo.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getCommandClass()</td>
<td>Return the command's class name.</td>
</tr>
<tr>
<td>getCommandName()</td>
<td></td>
</tr>
</tbody>
</table>
Return the command verb.

`Object getCommandObject(DataHandler dh, ClassLoader loader)`
Return the instantiated JavaBean component.

Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

`public CommandInfo(String verb, String className)`

CommandInfo

`verb` CommandInfo

`className`

CommandInfo

`public CommandInfo(String verb, String className)`

The Constructor for CommandInfo.

**Parameters:**

`verb` - The command verb this CommandInfo describes.
`className` - The command's fully qualified class name.

Method Detail

`public String getCommandName()`

return
getCommandName

public String getCommandName()

Return the command verb.

Returns:
the command verb.

getCommandClass

public String getCommandClass()

CommandMap CommandInfo null
return null

getCommandObject

public Object getCommandObject(DataHandler dh, ClassLoader loader) throws java.io.IOException, ClassNotFoundException, ClassNotFoundException

JavaBean

Beans.instantiate()

bean javax.activation.CommandObject
setCommandContext

DataHandler null bean : 
InputStream DataHandler IOExceptions bean

bean CommandObject java.io.Externalizable 
CommandObject DataHandler 
InputStream bean readExternal

dh DataHandler
loader bean ClassLoader
return bean
See also instantiate, javax.activation.CommandObject

getCommandObject

public Object getCommandObject(DataHandler dh, 
ClassLoader loader) 
throws IOException, 
ClassNotFoundException

Return the instantiated JavaBean component.

Begin by instantiating the component with Beans.instantiate().

If the bean implements the javax.activation.CommandObject interface, call its setCommandContext method.

If the DataHandler parameter is null, then the bean is instantiated with no data. NOTE: this may be useful if for some reason the DataHandler that is passed in throws IOExceptions when this method attempts to access its InputStream. It will allow the caller to retrieve a reference to the bean if it can be instantiated.

If the bean does NOT implement the CommandObject interface, this
method will check if it implements the java.io.Externalizable interface. If it does, the bean's readExternal method will be called if an InputStream can be acquired from the DataHandler.

**Parameters:**

- dh - The DataHandler that describes the data to be passed to the command.
- loader - The ClassLoader to be used to instantiate the bean.

**Returns:**
The bean

**Throws:**

- IOException
- ClassNotFoundException

**See Also:**

- Beans.instantiate(java.lang.ClassLoader, java.lang.String), CommandObject
javax.activation  Class CommandMap

java.lang.Object
   └ javax.activation.CommandMap

Direct Known Subclasses:
   MailcapCommandMap

public abstract class CommandMap
    extends Object

Extended by: MailcapCommandMap

CommandMap (MailcapCommandMap) CommandMap

The CommandMap class provides an interface to a registry of command objects available in the system. Developers are expected to either use the CommandMap implementation included with this package (MailcapCommandMap) or develop their own. Note that some of the methods in this class are abstract.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
<th>CommandMap ()</th>
</tr>
</thead>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>createDataContentHandler (String mimeType)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Locate a DataContentHandler that corresponds to the MIME type.</td>
</tr>
<tr>
<td></td>
<td>createDataContentHandler (String mimeType, DataSource ds)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>DataContentHandler</code></td>
<td>Locate a <code>DataContentHandler</code> that corresponds to the MIME type.</td>
</tr>
<tr>
<td><strong>abstract</strong> <code>CommandInfo[] getAllCommands(String mimeType)</code></td>
<td>Get all the available commands for this type.</td>
</tr>
<tr>
<td><code>CommandInfo[] getAllCommands(String mimeType, DataSource ds)</code></td>
<td>Get all the available commands for this type.</td>
</tr>
<tr>
<td><strong>abstract</strong> <code>CommandInfo getCommand(String mimeType, String cmdName)</code></td>
<td>Get the default command corresponding to the MIME type.</td>
</tr>
<tr>
<td><code>CommandInfo getCommand(String mimeType, String cmdName, DataSource ds)</code></td>
<td>Get the default command corresponding to the MIME type.</td>
</tr>
<tr>
<td>static <code>CommandMap getDefaultCommandMap()</code></td>
<td>Get the default <code>CommandMap</code>.</td>
</tr>
<tr>
<td><code>String[] getMimeTypes()</code></td>
<td>Get all the MIME types known to this command map.</td>
</tr>
<tr>
<td><strong>abstract</strong> <code>CommandInfo[] getPreferredCommands(String mimeType)</code></td>
<td>Get the preferred command list from a MIME Type.</td>
</tr>
<tr>
<td><code>CommandInfo[] getPreferredCommands(String mimeType, DataSource ds)</code></td>
<td>Get the preferred command list from a MIME Type.</td>
</tr>
<tr>
<td>static void <code>setDefaultCommandMap(CommandMap commandMap)</code></td>
<td>Set the default <code>CommandMap</code>.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`

**Constructor Detail**
public CommandMap()

CommandMap

public CommandMap()

### Method Detail

```java
public static CommandMap getDefaultCommandMap()
```

**CommandMap**

```java
public static CommandMap getDefaultCommandMap()
```

- CommandMap  
  ```java
  setDefaultCommandMap
  ```

- CommandMap CommandMap
  ```java
  MailcapCommandMap
  ```

  return CommandMap

### getDefaultsCommandMap

```java
public static CommandMap getDefaultsCommandMap()
```

Get the default CommandMap.

- In cases where a CommandMap instance has been previously set to some value (via `setDefaultCommandMap`) return the CommandMap.
- In cases where no CommandMap has been set, the CommandMap creates an instance of `MailcapCommandMap` and set that to the default, returning its value.

**Returns:**

the CommandMap
public static void setDefaultCommandMap(CommandMap commandMap)

CommandMap commandMap

null CommandMap

Throws SecurityException:

setDefaultCommandMap

public static void setDefaultCommandMap(CommandMap commandMap)

Set the default CommandMap. Reset the CommandMap to the default by calling this method with null.

Parameters:

commandMap - The new default CommandMap.

Throws:

SecurityException - if the caller doesn't have permission to change the default

abstract public CommandInfo[] getPreferredCommands(String mimeType)

MIME CommandMap

mimeType MIME return Bean CommandInfo

getPreferredCommands

public abstract CommandInfo[] getPreferredCommands(String mimeType)

Get the preferred command list from a MIME Type. The actual semantics are determined by the implementation of the CommandMap.

Parameters:
public CommandInfo[] getPreferredCommands(String mimeType, DataSource ds)

Get the preferred command list from a MIME Type. The actual semantics are determined by the implementation of the CommandMap.

The DataSource provides extra information, such as the file name, that a CommandMap implementation may use to further refine the list of commands that are returned. The implementation in this class simply calls the getPreferredCommands method that ignores this argument.

**Parameters:**

- mimeType - the MIME type
- ds - a DataSource for the data

**Returns:**

the CommandInfo classes that represent the command Beans.
Since:
JAF 1.1

abstract public CommandInfo[] getAllCommands(String mimeType)
MIME

mimeType
return

getAllCommands

public abstract CommandInfo[] getAllCommands(String mimeType)

Get all the available commands for this type. This method should return all the possible commands for this MIME type.

Parameters:
mimeType - the MIME type

Returns:
the CommandInfo objects representing all the commands.

public CommandInfo[] getAllCommands(String mimeType, DataSource ds)

DataSource CommandMap

mimeType
ds
return
since

getAllCommands

MIME

DataSource

CommandInfo

since
JAF 1.1
getAllCommands

public CommandInfo[] getAllCommands(String mimeType, DataSource ds)

Get all the available commands for this type. This method should return all the possible commands for this MIME type.

The DataSource provides extra information, such as the file name, that a CommandMap implementation may use to further refine the list of commands that are returned. The implementation in this class simply calls the getAllCommands method that ignores this argument.

Parameters:
mimeType - the MIME type
ds - a DataSource for the data

Returns:
the CommandInfo objects representing all the commands.

Since:
JAF 1.1

abstract public CommandInfo getCommand(String mimeType, String cmdName)

Get the default command corresponding to the MIME type.

Parameters:
mimeType - the MIME type
public CommandInfo getCommand(String mimeType, String cmdName, DataSource ds)

Get the default command corresponding to the MIME type.

The DataSource provides extra information, such as the file name, that a CommandMap implementation may use to further refine the command that is chosen. The implementation in this class simply calls the getCommand method that ignores this argument.

Parameters:
- mimeType - the MIME type
- cmdName - the command name
- ds - a DataSource for the data

Returns:
- the CommandInfo corresponding to the command.

Since:
abstract public DataContentHandler createDataContentHandler(String mimeType)
MIME DataContentHandler CommandMap

    mimeType MIME
    return MIME DataContentHandler

createDataContentHandler

public abstract DataContentHandler createDataContentHandler(String mimeType)

    Locate a DataContentHandler that corresponds to the MIME type. The mechanism and semantics for determining this are determined by the implementation of the particular CommandMap.

Parameters:
    mimeType - the MIME type

Returns:
    the DataContentHandler for the MIME type

public DataContentHandler createDataContentHandler(String mimeType, DataSource ds)
MIME DataContentHandler CommandMap

    DataSource CommandMap DataContentHandler
    mimeType MIME
    ds DataSource
    return MIME DataContentHandler
    since JAF 1.1
createDataContentHandler

public DataContentHandler createDataContentHandler(String mimeType, DataSource ds)

Locate a DataContentHandler that corresponds to the MIME type. The mechanism and semantics for determining this are determined by the implementation of the particular CommandMap.

The DataSource provides extra information, such as the file name, that a CommandMap implementation may use to further refine the choice of DataContentHandler. The implementation in this class simply calls the createDataContentHandler method that ignores this argument.

Parameters:
- mimeType: the MIME type
- ds: a DataSource for the data

Returns:
- the DataContentHandler for the MIME type

Since:
- JAF 1.1

getMimeTypes

public String[] getMimeTypes()

Get all the MIME types known to this command map. If the command map doesn't support this operation, null is returned.

Since:
- JAF 1.1
**Returns:**
array of MIME types as strings, or null if not supported

**Since:**
JAF 1.1
javax.activation  Interface CommandObject

public interface CommandObject

Activation Framework  JavaBean  DataHandler
JavaBean  Externalizable

JavaBeans components that are Activation Framework aware implement this interface to find out which command verb they're being asked to perform, and to obtain the DataHandler representing the data they should operate on. JavaBeans that don't implement this interface may be used as well. Such commands may obtain the data using the Externalizable interface, or using an application-specific method.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void setCommandContext(String verb, DataHandler dh)</td>
<td>Initialize the Command with the verb it is requested to handle and the DataHandler that describes the data it will operate on.</td>
</tr>
</tbody>
</table>

Method Detail

public void setCommandContext(String verb, DataHandler dh) throws java.io.IOException

DataHandler  Command

verb  Command Verb
dh  DataHandler
void setCommandContext(String verb, DataHandler dh)
throws IOException

Initialize the Command with the verb it is requested to handle and the DataHandler that describes the data it will operate on. **NOTE:** it is acceptable for the caller to pass *null* as the value for DataHandler.

**Parameters:**
- *verb* - The Command Verb this object refers to.
- *dh* - The DataHandler.

**Throws:**
- IOException

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.enterprise.deploy.shared Class CommandType

java.lang.Object
    javax.enterprise.deploy.shared.CommandType

public class CommandType
    extends Object

CommandType DeploymentStatus

Class CommandTypes defines enumeration values for the DeploymentStatus object.

Author:
    rsearls

Field Summary

<table>
<thead>
<tr>
<th>static CommandType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static DISTRIBUTE</td>
<td>The DeploymentManager action operation being processed is distribute.</td>
</tr>
<tr>
<td>static REDEPLOY</td>
<td>The DeploymentManager action operation being processed is redeploy.</td>
</tr>
<tr>
<td>static START</td>
<td>The DeploymentManager action operation being processed is start.</td>
</tr>
<tr>
<td>static STOP</td>
<td>The DeploymentManager action operation being processed is stop.</td>
</tr>
<tr>
<td>static UNDEPLOY</td>
<td>The DeploymentManager action operation being processed is undeploy.</td>
</tr>
</tbody>
</table>
Constructor Summary

<table>
<thead>
<tr>
<th>protected</th>
<th>CommandType(int value)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Construct a new enumeration value with the given integer value.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>static</th>
<th>CommandType.CommandType(int value)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return an object of the specified value.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>CommandType.CommandType()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>getEnumValueTable()</td>
</tr>
<tr>
<td></td>
<td>Returns the enumeration value table for class CommandType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>int getOffset()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the lowest integer value used by this enumeration value's enumeration class.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>String[] getStringTable()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the string table for class CommandType</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>getValue()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns this enumeration value's integer value.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>toString()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the string name of this CommandType or the integer value if outside the bounds of the table</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait
public static final CommandType DISTRIBUTED

The DeploymentManger action operation being processed is distribute.

START

public static final CommandType START

The DeploymentManger action operation being processed is start.

STOP

public static final CommandType STOP

The DeploymentManger action operation being processed is stop.

UNDEPLOY

public static final CommandType UNDEPLOY

The DeploymentManger action operation being processed is undeploy.

REDEPLOY

public static final CommandType REDEPLOY
The DeploymentManger action operation being processed is redeploy.

Constructor Detail

protected CommandType(int value)

Construct a new enumeration value with the given integer value.

Parameters:
value - Integer value.

Method Detail

public int getValue()

Returns this enumeration value's integer value.

Returns:
the value
protected String[] getStringTable()

getStringTable

protected String[] getStringTable()

  Returns the string table for class CommandType

protected CommandType[] getEnumValueTable()

getEnumValueTable

protected CommandType[] getEnumValueTable()

  Returns the enumeration value table for class CommandType

public static CommandType getCommandType(int value)

getCommandType

public static CommandType getCommandType(int value)

  Return an object of the specified value.

  Parameters:

  value - a designator for the object.
public String toString()

CommandType

toString

public String toString()

Return the string name of this CommandType or the integer value if outside the bounds of the table

Overrides:

toString in class Object

protected int getOffset()

0

return

getOffset

protected int getOffset()

Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0.

Returns:

the offset of the lowest enumeration value.
javax.xml.stream.events Interface Comment

All Superinterfaces:

XMLEvent, XMLStreamConstants

public interface Comment

extends XMLEvent

Implements: XMLEvent

version 1.0

An interface for comment events

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

| Fields inherited from interface javax.xml.stream.XMLStreamConstants |
| ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT |

Method Summary

| String getText () |
| Return the string data of the comment, returns empty string |
Methods inherited from interface javax.xml.stream.events.XMLEvent
- asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public String getText()

String getText()

Return the string data of the comment, returns empty string if it does not exist
javax.resource.spi  Class CommException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.resource.ResourceException
              └ javax.resource.spi.CommException

All Implemented Interfaces:
  Serializable

public class CommException
extends ResourceException

Extends: Throwable > Exception > ResourceException

EIS

    version 1.0

This indicates errors related to failed or interrupted communication with an EIS instance. Examples of common error conditions represented by this exception type are communication protocol error and invalidated connection due to server failure.

Version:
  1.0

Author:
  Rahul Sharma, Ram Jeyaraman

See Also:
  Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>CommException()</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
**CommException** *(String message)*
Constructs a new instance with the specified detail message.

**CommException** *(String message, String errorCode)*
Constructs a new throwable with the specified detail message and an error code.

**CommException** *(String message, Throwable cause)*
Constructs a new throwable with the specified detail message and cause.

**CommException** *(Throwable cause)*
Constructs a new throwable with the specified cause.

**Method Summary**

Methods inherited from class javax.resource.ResourceException
getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

**Constructor Detail**

public_CommException()  
null

CommException
public CommException()

Constructs a new instance with null as its detail message.

public CommException(String message)

message

CommException

public CommException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.

public CommException(Throwable cause)

cause throwable

cause Throwable

CommException

public CommException(Throwable cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

public CommException(String message, Throwable cause)

cause throwable

message
CommException

public CommException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public CommException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Class ComparisonTerm

javax.mail.search  
java.lang.Object
   ↓javax.mail.search.SearchTerm
      ↓javax.mail.search.ComparisonTerm

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   DateTerm, IntegerComparisonTerm

public abstract class ComparisonTerm
   extends SearchTerm

Extends: SearchTerm
Extended by: DateTerm, IntegerComparisonTerm

This class models the comparison operator. This is an abstract class; subclasses implement comparisons for different datatypes.

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected int</td>
<td>comparison</td>
</tr>
<tr>
<td></td>
<td>The comparison.</td>
</tr>
<tr>
<td>static int</td>
<td>EQ</td>
</tr>
</tbody>
</table>
Constructor Summary

ComparisonTerm()
LE

public static final int LE

See Also:
   Constant Field Values

_______________________________________________________________

LT

public static final int LT

See Also:
   Constant Field Values

_______________________________________________________________

EQ

public static final int EQ

See Also:
   Constant Field Values

_______________________________________________________________

NE

public static final int NE

See Also:
   Constant Field Values

_______________________________________________________________
GT
public static final int GT

See Also:
Constant Field Values

GE
public static final int GE

See Also:
Constant Field Values

comparison
protected int comparison

The comparison.

Constructor Detail

public ComparisonTerm()
public boolean equals(Object obj)

equals

public boolean equals(Object obj)

Equality comparison.

Overrides:
equals in class Object

public int hashCode()

hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

Overrides:
hashCode in class Object
javax.el  **Class CompositeELResolver**

```java
java.lang.Object  
    ▼ javax.el.ELResolver  
        ▼ javax.el.CompositeELResolver
```

public class **CompositeELResolver**

extends **ELResolver**

**Extends:** **ELResolver**

```
ELResolver  ELContext  ELResolver
CompositeELResolver
```

```
#getValue#getType#setValue  #isReadOnly  ELResolver (base, property)
propertyResolved  true  propertyResolved  false
```

CompositeELResolver  ELContext.propertyResolved  false

```
#getFeatureDescriptors  #getCommonPropertyType
ELContext.propertyResolved  ELResolver
```

since **JSP 2.1**

See also  **javax.el.ELContext, javax.el.ELResolver**

Maintains an ordered composite list of child **ELResolver**s.

Though only a single **ELResolver** is associated with an **ELContext**, there are usually multiple resolvers considered for any given variable or property resolution. **ELResolver**s are combined together using a **CompositeELResolver**, to define rich semantics for evaluating an expression.
For the `getValue(javax.el.ELContext, java.lang.Object, java.lang.Object)`, `getType(javax.el.ELContext, java.lang.Object, java.lang.Object)`, `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` and `isReadOnly(javax.el.ELContext, java.lang.Object, java.lang.Object)` methods, an `ELResolver` is not responsible for resolving all possible (base, property) pairs. In fact, most resolvers will only handle a base of a single type. To indicate that a resolver has successfully resolved a particular (base, property) pair, it must set the `propertyResolved` property of the `ELContext` to true. If it could not handle the given pair, it must leave this property alone. The caller must ignore the return value of the method if `propertyResolved` is false.

The `CompositeELResolver` initializes the `ELContext.propertyResolved` flag to false, and uses it as a stop condition for iterating through its component resolvers.

The `ELContext.propertyResolved` flag is not used for the design-time methods `getFeatureDescriptors(javax.el.ELContext, java.lang.Object)` and `getCommonPropertyType(javax.el.ELContext, java.lang.Object)`. Instead, results are collected and combined from all child `ELResolver`S for these methods.

Since: JSP 2.1

See Also: `ELContext`, `ELResolver`

---

### Field Summary

Fields inherited from class `javax.el.ELResolver`

| RESOLVABLE_AT_DESIGN_TIME, TYPE |

---

### Constructor Summary

`CompositeELResolver()`
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void add(ELResolver elResolver)</code></td>
<td>Adds the given resolver to the list of component resolvers.</td>
</tr>
<tr>
<td><code>Class&lt;?&gt; getCommonPropertyType(ELContext context, Object base)</code></td>
<td>Returns the most general type that this resolver accepts for the property argument, given a base object.</td>
</tr>
<tr>
<td><code>Iterator&lt;FeatureDescriptor&gt; getFeatureDescriptors(ELContext context, Object base)</code></td>
<td>Returns information about the set of variables or properties that can be resolved for the given base object.</td>
</tr>
<tr>
<td><code>Class&lt;?&gt; getType(ELContext context, Object base, Object property)</code></td>
<td>For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the <code>setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)</code> method.</td>
</tr>
<tr>
<td><code>Object getValue(ELContext context, Object base, Object property)</code></td>
<td>Attempts to resolve the given property object on the given base object by querying all component resolvers.</td>
</tr>
<tr>
<td><code>boolean isReadOnly(ELContext context, Object base, Object property)</code></td>
<td>For a given base and property, attempts to determine whether a call to <code>setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)</code> will always fail.</td>
</tr>
<tr>
<td><code>Object setValue(ELContext context, Object base, Object property, Object val)</code></td>
<td></td>
</tr>
</tbody>
</table>
void

Attempts to set the value of the given property object on the given base object.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public CompositeELResolver()

CompositeELResolver

public CompositeELResolver()

Method Detail

public void add(ELResolver elResolver)

elResolver

Throws

NullPointeException: null

add

public void add(ELResolver elResolver)

Adds the given resolver to the list of component resolvers.

Resolvers are consulted in the order in which they are added.
Parameters:
   elResolver - The component resolver to add.

Throws:
   NullPointerException - If the provided resolver is null.

```java
public Object getValue(ELContext context, Object base, Object property)
```

(base, property) ELContext propertyResolved true true
   ELContext propertyResolved false

1. getValue() context base property
2. ELContext propertyResolved false
3. getValue()
   null propertyResolved false

context base base null
property return ELContext propertyResolved true
Throws NullPointerException: context null
Throws PropertyNotFoundException: ELResolver (base, property)
Throws ELException: cause

getValue
public Object getValue(ELContext context, Object base, Object property)

Attempts to resolve the given property object on the given base object by querying all component resolvers.

If this resolver handles the given (base, property) pair, the propertyResolved property of the ELContext object must be set to true by the resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

First, propertyResolved is set to false on the provided ELContext.

Next, for each component resolver in this composite:

1. The getValue() method is called, passing in the provided context, base and property.
2. If the ELContext's propertyResolved flag is false then iteration continues.
3. Otherwise, iteration stops and no more component resolvers are considered. The value returned by getValue() is returned by this method.

If none of the component resolvers were able to perform this operation, the value null is returned and the propertyResolved flag remains set to false.

Any exception thrown by component resolvers during the iteration is propagated to the caller of this method.

Specified by: getValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - The base object whose property value is to be returned, or null to resolve a top-level variable.
property - The property or variable to be resolved.
Returns:
If the `propertyResolved` property of `ELContext` was set to `true`, then the result of the variable or property resolution; otherwise undefined.

Throws:
- `NullPointerException` - if context is `null`
- `PropertyNotFoundException` - if the given (base, property) pair is handled by this `ELResolver` but the specified variable or property does not exist or is not readable.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public Class<T> getType(ELContext context, Object base, Object property)
    base   property   #setValue   value
(base, property)   ELContext   propertyResolved
true   true

ELContext   propertyResolved   false
```

1. `getType()` `context` `base` `property`
2. `ELContext` `propertyResolved` `false`
3. `getType()`
   `null` `propertyResolved` `false`

```java
context
base   base   null
property
```
**get**"Type**

```java
public Class<?> getType(ELContext context, Object base, Object property)
```

For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` method. The result is obtained by querying all component resolvers.

If this resolver handles the given (base, property) pair, the `propertyResolved` property of the `ELContext` object must be set to `true` by the resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

First, `propertyResolved` is set to `false` on the provided `ELContext`.

Next, for each component resolver in this composite:

1. The `getType()` method is called, passing in the provided context, base and property.
2. If the `ELContext`'s `propertyResolved` flag is `false` then iteration continues.
3. Otherwise, iteration stops and no more component resolvers are considered. The value returned by `getType()` is returned by this method.

If none of the component resolvers were able to perform this operation, the value `null` is returned and the `propertyResolved` flag remains set to `false`.
Any exception thrown by component resolvers during the iteration is propagated to the caller of this method.

**Specified by:**
`getType` in class `ELResolver`

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The base object whose property value is to be analyzed, or `null` to analyze a top-level variable.
- `property` - The property or variable to return the acceptable type for.

**Returns:**
If the `propertyResolved` property of `ELContext` was set to `true`, then the most general acceptable type; otherwise undefined.

**Throws:**
- `NullPointerException` - if `context` is `null`
- `PropertyNotFoundException` - if the given (base, property) pair is handled by this `ELResolver` but the specified variable or property does not exist or is not readable.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public void setValue(ELContext context, Object base, Object property, Object val)

(base, property) ELContext propertyResolved
true true

ELContext propertyResolved false
```

1. `setValue()` context base property value
2. ELContext propertyResolved false
3. propertyResolved false

context
base base null
property
val

Throws NullPointerException: context null
Throws PropertyNotFoundException: ELResolver (base, property)
Throws PropertyNotWritableException: ELResolver (base, property)
Throws ELException: cause

setValue

public void setValue(ELContext context,
        Object base,
        Object property,
        Object val)

Attempts to set the value of the given property object on the given base object. All component resolvers are asked to attempt to set the value.

If this resolver handles the given (base, property) pair, the propertyResolved property of the ELContext object must be set to true by the resolver, before returning. If this property is not true after this method is called, the caller can safely assume no value has been set.

First, propertyResolved is set to false on the provided ELContext.
Next, for each component resolver in this composite:

1. The `setValue()` method is called, passing in the provided context, base, property and value.
2. If the `ELContext`'s propertyResolved flag is `false` then iteration continues.
3. Otherwise, iteration stops and no more component resolvers are considered.

If none of the component resolvers were able to perform this operation, the propertyResolved flag remains set to `false`.

Any exception thrown by component resolvers during the iteration is propagated to the caller of this method.

**Specified by:**

`setValue` in class `ELResolver`

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The base object whose property value is to be set, or `null` to set a top-level variable.
- `property` - The property or variable to be set.
- `val` - The value to set the property or variable to.

**Throws:**
- `NullPointerException` - if `context` is `null`
- `PropertyNotFoundException` - if the given (base, property) pair is handled by this `ELResolver` but the specified variable or property does not exist.
- `PropertyNotWritableException` - if the given (base, property) pair is handled by this `ELResolver` but the specified variable or property is not writable.
- `ELException` - if an exception was thrown while attempting to set the property or variable. The thrown exception must be included as the cause property of this exception, if available.

```java
public boolean isReadOnly(ELContext context, Object
```
base, Object property)
(base, property) ELContext propertyResolved true true

1. isReadOnly() context base property
2. ELContext propertyResolved false
3. isReadOnly()

false propertyResolved false

context
base base null
property
return ELContext propertyResolved true true false
Throws NullPointerException: context null
Throws PropertyNotFoundException: ELResolver (base, property)
Throws ELException: cause

isReadOnly

public boolean isReadOnly(ELContext context,
Object base,
Object property)

For a given base and property, attempts to determine whether a call to setValue(javax.el.ELContext, java.lang.Object, java.lang.Object) will always fail. The result is
obtained by querying all component resolvers.

If this resolver handles the given (base, property) pair, the propertyResolved property of the ELContext object must be set to true by the resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

First, propertyResolved is set to false on the provided ELContext.

Next, for each component resolver in this composite:

1. The isReadOnly() method is called, passing in the provided context, base and property.
2. If the ELContext's propertyResolved flag is false then iteration continues.
3. Otherwise, iteration stops and no more component resolvers are considered. The value returned by isReadOnly() is returned by this method.

If none of the component resolvers were able to perform this operation, the value false is returned and the propertyResolved flag remains set to false.

Any exception thrown by component resolvers during the iteration is propagated to the caller of this method.

Specified by:

isReadOnly in class ELResolver

Parameters:

- context - The context of this evaluation.
- base - The base object whose property value is to be analyzed, or null to analyze a top-level variable.
- property - The property or variable to return the read-only status for.

Returns:

If the propertyResolved property of ELContext was set to true, then true if the property is read-only or false if not; otherwise undefined.
Throws:

- NullPointerException - if context is null
- PropertyNotFoundException - if the given (base, property) pair is handled by this ELResolver but the specified variable or property does not exist.
- ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

public java.util.Iterator<E> getFeatureDescriptors(ELContext context, Object base)

Returns information about the set of variables or properties that can be resolved for the given base object. One use for this method is to assist tools in auto-completion. The results are collected from all component resolvers.

The propertyResolved property of the ELContext is not relevant to this method. The results of all ELResolvers are concatenated.
The Iterator returned is an iterator over the collection of FeatureDescriptor objects returned by the iterators returned by each component resolver’s getFeatureDescriptors method. If null is returned by a resolver, it is skipped.

Specified by:

getFeatureDescriptors in class ELResolver

Parameters:

context - The context of this evaluation.
base - The base object whose set of valid properties is to be enumerated, or null to enumerate the set of top-level variables that this resolver can evaluate.

Returns:

An Iterator containing zero or more (possibly infinitely more) FeatureDescriptor objects, or null if this resolver does not handle the given base object or that the results are too complex to represent with this method.

See Also:

FeatureDescriptor

```
public Class<T> getCommonPropertyType(ELContext context, Object base)
    base property
    Class getCommonPropertyType null
        context
        base
        return null
    base
    return ELResolver base null property
    Object.class base property
```

getCommonPropertyType

```
public Class<? super T> getCommonPropertyType(ELContext context, Object base)
```

Returns the most general type that this resolver accepts for the property argument, given a base object. One use for this method is to assist tools in auto-completion. The result is obtained by querying all component resolvers.

The class returned is the most specific class that is a common superclass of all the classes returned by each component resolver's getCommonPropertyType method. If null is returned by a resolver, it is skipped.

**Specified by:**

getCommonPropertyType in class ELResolver

**Parameters:**

- context - The context of this evaluation.
- base - The base object to return the most general property type for, or null to enumerate the set of top-level variables that this resolver can evaluate.

**Returns:**

null if this ELResolver does not know how to handle the given base object; otherwise Object.class if any type of property is accepted; otherwise the most general property type accepted for the given base.
The `Concept` interface is used to represent taxonomy elements and their structural relationship with each other in order to describe an internal taxonomy. Concept instances are used to define tree structures where the root of the tree is a `ClassificationScheme` instance and each node in the tree is a `Concept` instance. Two Concepts may be defined as being equivalent.

Figure 1 shows how Concept instances are used to represent taxonomy elements and their structural relationship with each other in order to describe an internal taxonomy.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addChildConcept(Concept concept)</td>
<td>Adds a child Concept.</td>
</tr>
<tr>
<td>void addChildConcepts(Collection concepts)</td>
<td>Adds a Collection of Concept children.</td>
</tr>
<tr>
<td>int getChildConceptCount()</td>
<td>Gets number of children.</td>
</tr>
<tr>
<td>Collection getChildrenConcepts()</td>
<td>Gets all immediate children Concepts.</td>
</tr>
<tr>
<td>ClassificationScheme getClassificationScheme()</td>
<td>Gets the ClassificationScheme that this Concept is a descendant of.</td>
</tr>
<tr>
<td>Collection getDescendantConcepts()</td>
<td>Gets all descendant Concepts.</td>
</tr>
<tr>
<td>RegistryObject getParent()</td>
<td>Gets the parent Concept or ClassificationScheme for this object.</td>
</tr>
<tr>
<td>Concept getParentConcept()</td>
<td>Gets the parent Concept or null if parent is a ClassificationScheme.</td>
</tr>
<tr>
<td>String getPath()</td>
<td>Gets the canonical path representation for this Concept.</td>
</tr>
<tr>
<td>String getValue()</td>
<td>Gets the value (usually a code in a taxonomy) associated with this Concept.</td>
</tr>
<tr>
<td>void removeChildConcept(Concept concept)</td>
<td>Removes a child Concept.</td>
</tr>
<tr>
<td>void removeChildConcepts(Collection concepts)</td>
<td></td>
</tr>
</tbody>
</table>
Removes a Collection of children Concepts.

`setValue(String value)`

Sets the value (usually a taxonomy value) associated with this Concept.

Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`

- addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML

Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`

- addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

### Method Detail

`public String getValue() throws JAXRException`  

Concept

```java
0

return Concept

Throws JAXRException: JAXR
```

`getValue`
String getValue() throws JAXRException

Gets the value (usually a code in a taxonomy) associated with this Concept.

**Capability Level: 0**

**Returns:**
- the value (usually a taxonomy value) associated with this Concept

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

public void setValue(String value) throws JAXRException

Concept

0

value - the value (usually a taxonomy value) associated with this Concept

**Throws:**
- JAXRException: JAXR

---

void setValue(String value) throws JAXRException

Sets the value (usually a taxonomy value) associated with this Concept.

**Capability Level: 0**

**Parameters:**
- value - the value (usually a taxonomy value) associated with this Concept

**Throws:**
public void addChildConcept(Concept concept) throws JAXRException

Adds a child Concept.

Capability Level: 0

Parameters:
concept - the concept being added as a child of this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void addChildConcepts(java.util.Collection<E> concepts) throws JAXRException

Concept Collection
Throws JAXRException: JAXR

addChildConcepts

void addChildConcepts(Collection concepts) throws JAXRException

Adds a Collection of Concept children.

Capability Level: 0

Parameters:
    concepts - the Collection of Concepts being added as a children
    of this object

Throws:
    JAXRException - If the JAXR provider encounters an internal
    error

public void removeChildConcept(Concept concept) throws JAXRException

Concept

0

concept Concept

Throws JAXRException: JAXR

removeChildConcept

void removeChildConcept(Concept concept) throws JAXRException

Removes a child Concept.

Capability Level: 0
public void removeChildConcepts(java.util.Collection<E> concepts) throws JAXRException
Concep Collection

0

concepts Concept Collection

Throws JAXRException: JAXR

removeChildConcepts

void removeChildConcepts(Collection concepts)
throws JAXRException

Removes a Collection of children Concepts.

Capability Level: 0

Parameters:
concepts - the Collection of Concepts being removed as children Concepts of this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public int getChildConceptCount() throws JAXRException
getChildConceptCount

```java
int getChildConceptCount() throws JAXRException
```

Gets number of children.

**Capability Level:** 0

**Returns:**
the number of children Concepts

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

-------------------------------

**public java.util.Collection<E> getChildrenConcepts()**

```java
throws JAXRException
```

**Concept**

0

```java
return Concept CollectionCollection null
```

**Throws**
*JAXRException* - JAXR

**See also**
[javax.xml.registry.infomodel.Concept](https://docs.oracle.com/javaee/8/api/javax/xml/registry/infomodel/Concept.html)

getChildrenConcepts

```java
Collection getChildrenConcepts()
```

```java
throws JAXRException
```
Gets all immediate children Concepts.

**Capability Level: 0**

**Returns:**
Collection of Concept instances. The Collection may be empty but not null.

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**See Also:**
- `Concept`

---

```java
public java.util.Collection<E> getDescendantConcepts()
throws JAXRException
```

```
Concept 0

    return Concept Collection Collection null

Throws  JAXRException: JAXR
See also  java.xml.registry.infomodel.Concept
```

### getDescendantConcepts

**Returns:**
Collection of Concept instances. The Collection may be empty but not null.

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**See Also:**
- `javax.xml.registry.infomodel.Concept`
public Concept getParentConcept() throws JAXRException
Concept Concept  ClassificationScheme null

0

return Concept
Throws JAXRException: JAXR parent
label

directed
supplierCardinality 0..1
associates

getParentConcept

Concept getParentConcept() throws JAXRException

Gets the parent Concept or null if parent is a ClassificationScheme.

Capability Level: 0

Returns:
the Concept that is the parent of this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public ClassificationScheme getClassificationScheme() throws JAXRException
Concept ClassificationScheme
getClassificationScheme

getClassificationScheme():

0

return ClassificationScheme null

Throws

JAXRException: JAXR

label classificationScheme

directed

supplierCardinality 0..1

associates

\{javax.xml.registry.infomodel.ClassificationScheme\}

Concept

public String getPath() throws JAXRException

0

return String

Throws

JAXRException: JAXR

getPath
String getPath() throws JAXRException

Gets the canonical path representation for this Concept.

**Capability Level: 0**

**Returns:**
the canonical path String representing this object

**Throws:**
  JAXRException - If the JAXR provider encounters an internal error

---

public RegistryObject getParent() throws JAXRException

Concept ClassificationScheme

0

    return Concept ClassificationScheme

**Throws**
  JAXRException: JAXR

**getParent**

RegistryObject getParent() throws JAXRException

Gets the parent Concept or ClassificationScheme for this object.

**Capability Level: 0**

**Returns:**
the parent Concept or ClassificationScheme for this object

**Throws:**
  JAXRException - If the JAXR provider encounters an internal error

---
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Class ConcurrentAccessException

java.lang.Object  
   ▼ java.lang.Throwable  
      ▼ java.lang.Exception  
         ▼ java.lang.RuntimeException  
            ▼ javax.ejb.EJBException  
               ▼ javax.ejb.ConcurrentAccessException

All Implemented Interfaces:
   Serializable

public class ConcurrentAccessException
extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

ConcurrentAccessException Bean

A ConcurrentAccessException indicates that the client has attempted an invocation on a stateful session bean while another invocation is in progress.

See Also:
   Serialized Form

Constructor Summary

ConcurrentAccessException()  
   Constructs an ConcurrentAccessException with no detail message.

ConcurrentAccessException(String message)  
   Constructs an ConcurrentAccessException with the specified detailed message.

ConcurrentAccessException(String message, Exception ex)
Constructs an ConcurrentAccessException with the specified detail message and a nested exception.

**Method Summary**

Methods inherited from class javax.ejb.EJBException
- getCausedByException
- getMessage
- printStackTrace
- printStackTrace
- printStackTrace

Methods inherited from class java.lang.Throwable
- fillInStackTrace
- getCause
- getLocalizedMessage
- getStackTrace
- initCause
- setStackTrace
- toString

Methods inherited from class java.lang.Object
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

**Constructor Detail**

public ConcurrentAccessException()
ConcurrentAccessException

ConcurrentAccessException

public ConcurrentAccessException()

Constructs an ConcurrentAccessException with no detail message.

public ConcurrentAccessException(String message)
ConcurrentAccessException
ConcurrentAccessException

public ConcurrentAccessException(String message)

Constructs an ConcurrentAccessException with the specified detailed message.

public ConcurrentAccessException(String message, Exception ex)
ConcurrentAccessException

ConcurrentAccessException

public ConcurrentAccessException(String message, Exception ex)

Constructs an ConcurrentAccessException with the specified detail message and a nested exception.
Class
ConfigurationException

javax.enterprise.deploy.spi.exceptions

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
     └ javax.enterprise.deploy.spi.exceptions.ConfigurationException

All Implemented Interfaces:
  Serializable

public classConfigurationException
extends Exception

Extends: Throwable > Exception

Bean

This exception reports errors in generating a configuration bean.

Author:
  gfink

See Also:
  Serialized Form

Constructor Summary

ConfigurationException()
  Creates newConfigurationException without detail message.

ConfigurationException(String msg)
  Constructs anConfigurationException with the specified detail message.
### Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public ConfigurationException()
    ConfigurationException
```

**ConfigurationException**

```java
public ConfigurationException()
    Creates new ConfigurationException without detail message.
```

```java
public ConfigurationException(String msg)
    ConfigurationException
```

**ConfigurationException**

```java
public ConfigurationException(String msg)
    Constructs an ConfigurationException with the specified detail message.
```
Parameters:

msg - the detail message.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface Connection

All Known Subinterfaces:
   QueueConnection, TopicConnection, XAConnection,
  XAQueueConnection, XATopicConnection

public interface Connection

Implemented by: QueueConnection, TopicConnection, XAConnection

Connection  JMS  Java (JVM)

- JMS  TCP/IP
- ConnectionMetaData
- ExceptionListener

JMS API

JMS

start

version 1.1 - February 1, 2002

See  
also  
javax.jms.ConnectionFactory,  
javax.jms.QueueConnection,  
javax.jms.TopicConnection
A `Connection` object is a client's active connection to its JMS provider. It typically allocates provider resources outside the Java virtual machine (JVM).

Connections support concurrent use.

A connection serves several purposes:

- It encapsulates an open connection with a JMS provider. It typically represents an open TCP/IP socket between a client and the service provider software.
- Its creation is where client authentication takes place.
- It can specify a unique client identifier.
- It provides a `ConnectionMetaData` object.
- It supports an optional `ExceptionListener` object.

Because the creation of a connection involves setting up authentication and communication, a connection is a relatively heavyweight object. Most clients will do all their messaging with a single connection. Other more advanced applications may use several connections. The JMS API does not architect a reason for using multiple connections; however, there may be operational reasons for doing so.

A JMS client typically creates a connection, one or more sessions, and a number of message producers and consumers. When a connection is created, it is in stopped mode. That means that no messages are being delivered.

It is typical to leave the connection in stopped mode until setup is complete (that is, until all message consumers have been created). At that point, the client calls the connection's `start` method, and messages begin arriving at the connection's consumers. This setup convention minimizes any client confusion that may result from asynchronous message delivery while the client is still in the process of setting itself up.

A connection can be started immediately, and the setup can be done afterwards. Clients that do this must be prepared to handle asynchronous message delivery while they are still in the process of setting up.

A message producer can send messages while a connection is stopped.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void close()</td>
<td>Closes the connection.</td>
</tr>
<tr>
<td>ConnectionConsumer createConnectionConsumer(Destination destination, String messageSelector, ServerSessionPool sessionPool, int maxMessages)</td>
<td>Creates a connection consumer for this connection (optional operation).</td>
</tr>
<tr>
<td>ConnectionConsumer createDurableConnectionConsumer(Topic topic, String subscriptionName, String messageSelector, ServerSessionPool sessionPool, int maxMessages)</td>
<td>Create a durable connection consumer for this connection (optional operation).</td>
</tr>
<tr>
<td>Session createSession(boolean transacted, int acknowledgeMode)</td>
<td>Creates a Session object.</td>
</tr>
<tr>
<td>String getClientID()</td>
<td>Gets the client identifier for this connection.</td>
</tr>
<tr>
<td>ExceptionListener getExceptionHandler()</td>
<td>Gets the ExceptionListener object for this connection.</td>
</tr>
<tr>
<td>ConnectionMetaData getMetaData()</td>
<td>Gets the metadata for this connection.</td>
</tr>
<tr>
<td>void setClientID(String clientId)</td>
<td>Sets the client identifier for this connection.</td>
</tr>
<tr>
<td>void setExceptionHandler(ExceptionListener listener)</td>
<td>Sets an exception listener for this connection.</td>
</tr>
<tr>
<td>void start()</td>
<td></td>
</tr>
</tbody>
</table>
### void start()
Starts (or restarts) a connection's delivery of incoming messages.

### void stop()
Temporarily stops a connection's delivery of incoming messages.

## Method Detail

**public** [Session](createSession) **createSession**(boolean transacted, int acknowledgeMode) **throws** [JMSException](JMSException)

**Session**

- **transacted**

  - Indicates whether the session is transacted

**acknowledgeMode**

- Session.AUTO_ACKNOWLEDGE
- Session.CLIENT_ACKNOWLEDGE
- Session.DUPS_OK_ACKNOWLEDGE

**return**

**Throws** [JMSException](JMSException): Connection since 1.1

**AUTO_ACKNOWLEDGE, CLIENT_ACKNOWLEDGE, DUPS_OK_ACKNOWLEDGE**

**createSession**

**Session** **createSession**(boolean transacted, int acknowledgeMode) **throws** [JMSException](JMSException)

Creates a Session object.

**Parameters:**

- **transacted** - Indicates whether the session is transacted
- **acknowledgeMode** - Indicates whether the consumer or the client will acknowledge any messages it receives; ignored if the session is transacted. Legal values are
Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, and Session.DUPS_OK_ACKNOWLEDGE.

**Returns:**
a newly created session

**Throws:**
*JMSException* - if the Connection object fails to create a session due to some internal error or lack of support for the specific transaction and acknowledgement mode.

**Since:**
1.1

**See Also:**
*Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, Session.DUPS_OK_ACKNOWLEDGE*

---

**public String getClientID() throws JMSException**

**JMS** ConnectionFactory setClientID

return

**Throws** *JMSException*: JMS ID

**getClientID**

**String** getClientID() throws JMSException

Gets the client identifier for this connection.

This value is specific to the JMS provider. It is either preconfigured by an administrator in a ConnectionFactory object or assigned dynamically by the application by calling the setClientID method.

**Returns:**
the unique client identifier

**Throws:**
*JMSException* - if the JMS provider fails to return the client ID for
this connection due to some internal error.

```java
public void setClientID(String clientID) throws JMSException {
    JMS ConnectionFactory Connection
    IllegalStateException
    IllegalStateException

    JMS API

    clientId  JMS ID  InvalidClientIDException
    clientId
    Throws  JMSException: JMS ID
    Throws  InvalidClientIDException: JMS ID
    Throws  IllegalStateException: JMS ID

    setClientID

    void setClientID(String clientID)
        throws JMSException

    Sets the client identifier for this connection.

    The preferred way to assign a JMS client's client identifier is for it to be configured in a client-specific ConnectionFactory object and transparently assigned to the Connection object it creates.

    Alternatively, a client can set a connection's client identifier using a provider-specific value. The facility to set a connection's client identifier explicitly is not a mechanism for overriding the identifier that has been administratively configured. It is provided for the case
where no administratively specified identifier exists. If one does exist, an attempt to change it by setting it must throw an `IllegalStateException`. If a client sets the client identifier explicitly, it must do so immediately after it creates the connection and before any other action on the connection is taken. After this point, setting the client identifier is a programming error that should throw an `IllegalStateException`.

The purpose of the client identifier is to associate a connection and its objects with a state maintained on behalf of the client by a provider. The only such state identified by the JMS API is that required to support durable subscriptions.

If another connection with the same `clientID` is already running when this method is called, the JMS provider should detect the duplicate ID and throw an `InvalidClientIDException`.

**Parameters:**
- `clientID` - the unique client identifier

**Throws:**
- `JMSException` - if the JMS provider fails to set the client ID for this connection due to some internal error.
- `InvalidClientIDException` - if the JMS client specifies an invalid or duplicate client ID.
- `IllegalStateException` - if the JMS client attempts to set a connection's client ID at the wrong time or when it has been administratively configured.

```java
public ConnectionMetaData getMetaData() throws JMSException
```

**Throws**
- `JMSException`: JMS

**See also**
- `javax.jms.ConnectionMetaData`
ConnectionMetaData getMetaData() throws JMSException

Gets the metadata for this connection.

**Returns:**
the connection metadata

**Throws:**
JMSException - if the JMS provider fails to get the connection metadata for this connection.

**See Also:**
ConnectionMetaData

public ExceptionListener getExceptionListener() throws JMSException

```
ExceptionListener return
Connection   ExceptionListener   ExceptionListener   null
Throws      JMSException: JMS
```

See also
setExceptionListener

getExceptionListener

ExceptionListener getExceptionListener() throws JMSException

Gets the ExceptionListener object for this connection. Not every Connection has an ExceptionListener associated with it.

**Returns:**
the ExceptionListener for this connection, or null. if no ExceptionListener is associated with this connection.

**Throws:**
JMSException - if the JMS provider fails to get the ExceptionListener for this connection.

**See Also:**
setExceptionListener(javax.jms.ExceptionListener)
public void setExceptionListener(ExceptionListener listener) throws JMSException

JMS ExceptionListener onException
JMSException

ExceptionListener

JMS

/listener

Throws JMSException: JMS

setExceptionListener

void setExceptionListener(ExceptionListener listener)
throws JMSException

Sets an exception listener for this connection.

If a JMS provider detects a serious problem with a connection, it informs the connection's ExceptionListener, if one has been registered. It does this by calling the listener's onException method, passing it a JMSException object describing the problem.

An exception listener allows a client to be notified of a problem asynchronously. Some connections only consume messages, so they would have no other way to learn their connection has failed.

A connection serializes execution of its ExceptionListener.

A JMS provider should attempt to resolve connection problems itself before it notifies the client of them.
Parameters:
   listener - the exception listener

Throws:
   JMSException - if the JMS provider fails to set the exception listener for this connection.

---

public void start() throws JMSException

Starts (or restarts) a connection's delivery of incoming messages. A call to start on a connection that has already been started is ignored.

Throws:
   JMSException - if the JMS provider fails to start message delivery due to some internal error.

See Also:
   stop()
void stop() throws JMSException

Temporarily stops a connection's delivery of incoming messages. Delivery can be restarted using the connection's start method. When the connection is stopped, delivery to all the connection's message consumers is inhibited: synchronous receives block, and messages are not delivered to message listeners.

This call blocks until receives and/or message listeners in progress have completed.

Stopping a connection has no effect on its ability to send messages. A call to stop on a connection that has already been stopped is ignored.

A call to stop must not return until delivery of messages has paused. This means that a client can rely on the fact that none of its message listeners will be called and that all threads of control waiting for receive calls to return will not return with a message until the connection is restarted. The receive timers for a stopped connection continue to advance, so receives may time out while the connection is stopped.

If message listeners are running when stop is invoked, the stop call must wait until all of them have returned before it may return. While these message listeners are completing, they must have the full services of the connection available to them.

Throws:
JMSException - if the JMS provider fails to stop message delivery
due to some internal error.

See Also:

start()

public void close() throws JMSException

JVM

null close JMS

commit rollback

acknowledge IllegalStateException

Throws JMSException: JMS

close

void close() throws JMSException

Closes the connection.

Since a provider typically allocates significant resources outside the JVM on behalf of a connection, clients should close these resources when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

There is no need to close the sessions, producers, and consumers of a closed connection.
Closing a connection causes all temporary destinations to be deleted.

When this method is invoked, it should not return until message processing has been shut down in an orderly fashion. This means that all message listeners that may have been running have returned, and that all pending receives have returned. A close terminates all pending message receives on the connection's sessions' consumers. The receives may return with a message or with null, depending on whether there was a message available at the time of the close. If one or more of the connection's sessions' message listeners is processing a message at the time when connection close is invoked, all the facilities of the connection and its sessions must remain available to those listeners until they return control to the JMS provider.

Closing a connection causes any of its sessions' transactions in progress to be rolled back. In the case where a session's work is coordinated by an external transaction manager, a session's commit and rollback methods are not used and the result of a closed session's work is determined later by the transaction manager.

Closing a connection does NOT force an acknowledgment of client-acknowledged sessions.

Invoking the acknowledge method of a received message from a closed connection's session must throw an IllegalStateException. Closing a closed connection must NOT throw an exception.

Throws:

- JMSException - if the JMS provider fails to close the connection due to some internal error. For example, a failure to release resources or to close a socket connection can cause this exception to be thrown.

```java
public ConnectionConsumer createConnectionConsumer(Destination destination, String messageSelector, ServerSessionPool sessionPool, int
```
maxMessages) throws JMSException
JMS

  destination
messageSelector null
sessionPool
maxMessages
return

Throws JMSException: Connection sessionPool
messageSelector

Throws InvalidDestinationException:

Throws InvalidSelectorException:
since 1.1
See also javax.jms.ConnectionConsumer

createConnectionConsumer

ConnectionConsumer createConnectionConsumer(Destination destination,
String messageSelector,
ServerSessionPool sessionPool,
int maxMessages)
throws JMSException

Creates a connection consumer for this connection (optional operation). This is an expert facility not used by regular JMS clients.

Parameters:
  destination - the destination to access
messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
sessionPool - the server session pool to associate with this connection consumer
maxMessages - the maximum number of messages that can be assigned to a server session at one time

Returns:
  the connection consumer
Throws:

- `JMSException` - if the `Connection` object fails to create a connection consumer due to some internal error or invalid arguments for `sessionPool` and `messageSelector`.
- `InvalidDestinationException` - if an invalid destination is specified.
- `InvalidSelectorException` - if the message selector is invalid.

Since: 1.1

See Also:

- `ConnectionConsumer`

```java
public ConnectionConsumer createDurableConnectionConsumer(Topic topic, String subscriptionName, String messageSelector, ServerSessionPool sessionPool, int maxMessages) throws JMSException
```

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>topic</code></td>
<td>JMS topic</td>
</tr>
<tr>
<td><code>subscriptionName</code></td>
<td>String</td>
</tr>
<tr>
<td><code>messageSelector</code></td>
<td>null</td>
</tr>
<tr>
<td><code>sessionPool</code></td>
<td>ServerSessionPool</td>
</tr>
<tr>
<td><code>maxMessages</code></td>
<td>int</td>
</tr>
</tbody>
</table>

Returns

`ConnectionConsumer`

Throws

- `JMSException`: Connection, `sessionPool`, `messageSelector`.
- `InvalidDestinationException`.
- `InvalidSelectorException`.

Since: 1.1

See also

- `javax.jms.ConnectionConsumer`
Create a durable connection consumer for this connection (optional operation). This is an expert facility not used by regular JMS clients.

Parameters:
- topic - topic to access
- subscriptionName - durable subscription name
- messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- sessionPool - the server session pool to associate with this durable connection consumer
- maxMessages - the maximum number of messages that can be assigned to a server session at one time

Returns:
- the durable connection consumer

Throws:
- JMSException - if the Connection object fails to create a connection consumer due to some internal error or invalid arguments for sessionPool and messageSelector.
- InvalidDestinationException - if an invalid destination is specified.
- InvalidSelectorException - if the message selector is invalid.

Since:
- 1.1

See Also:
- ConnectionConsumer
to license terms.

PS:
A Connection represents an application-level handle that is used by a client to access the underlying physical connection. The actual physical connection associated with a Connection instance is represented by a ManagedConnection instance.

A client gets a Connection instance by using the `getConnection` method on a `ConnectionFactory` instance. A connection can be associated with zero or more `Interaction` instances.

**Version:**
0.8

**Author:**
Rahul Sharma

**See Also:**
`ConnectionFactory`, `Interaction`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>close()</code></td>
<td>void</td>
<td>Initiates close of the connection handle at the application level.</td>
</tr>
</tbody>
</table>
**Method Detail**

```java
public Interaction createInteraction() throws ResourceException
```

<table>
<thead>
<tr>
<th>Public Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>createInteraction()</strong></td>
<td>Creates an Interaction associated with this Connection. <strong>Interaction</strong> throws <strong>ResourceException</strong>: Interaction</td>
</tr>
<tr>
<td><strong>getLocalTransaction()</strong></td>
<td>Returns an LocalTransaction instance that enables a component to demarcate resource manager local transactions on the Connection.</td>
</tr>
<tr>
<td><strong>getMetaData()</strong></td>
<td>Gets the information on the underlying EIS instance represented through an active connection.</td>
</tr>
<tr>
<td><strong>getResultSetInfo()</strong></td>
<td>Gets the information on the ResultSet functionality supported by a connected EIS instance.</td>
</tr>
</tbody>
</table>

**createInteraction**

- **Interaction createInteraction()**
- **throws** **ResourceException**

Creates an Interaction associated with this Connection. An Interaction enables an application to execute EIS functions.

**Returns:**
- Interaction instance

**Throws:**
- **ResourceException** - Failed to create an Interaction
public `LocalTransaction` `getLocalTransaction()` throws `ResourceException`

Returns an `LocalTransaction` instance that enables a component to demarcate resource manager local transactions on the `Connection`. If a resource adapter does not allow a component to demarcate local transactions on a `Connection` using `LocalTransaction` interface, then the method `getLocalTransaction` should throw a `NotSupportedException`.

**Returns:**
- `LocalTransaction` instance

**Throws:**
- `ResourceException` - Failed to return a `LocalTransaction` instance because of a resource adapter error
- `NotSupportedException` - Demarcation of Resource manager local transactions is not supported on this `Connection`

**See Also:**
- `LocalTransaction`

public `ConnectionMetaData` `getMetaData()` throws `ResourceException`
getMetaData

ConnectionMetaData getMetaData()
throws ResourceException

Gets the information on the underlying EIS instance represented through an active connection.

Returns:
ConnectionMetaData instance representing information about the EIS instance

Throws:
ResourceException - Failed to get information about the connected EIS instance. Error can be resource adapter-internal, EIS-specific or communication related.

public ResultSetInfo getResultSetInfo() throws ResourceException

EIS ResultSet

return ResultSetInfo
Threws ResourceException: ResultSet

Throws NotSupportedException: ResultSet

getResultSetInfo

ResultSetInfo getResultSetInfo()
throws ResourceException

Gets the information on the ResultSet functionality supported by a connected EIS instance.
public void close() throws ResourceException

EIS

ResourceException:

close

Initiates close of the connection handle at the application level. A client should not use a closed connection to interact with an EIS.

Throws:

ResourceException - Exception thrown if close on a connection handle fails.

Any invalid connection close invocation--example, calling close on a connection handle that is already closed--should also throw this exception.

Resources:

Returns:
ResultSetInfo instance

Throws:
ResourceException - Failed to get ResultSet related information
NotSupportedException - ResultSet functionality is not supported
PS:
javax.xml.registry Interface Connection

All Known Subinterfaces:
  FederatedConnection

public interface Connection

Implemented by: FederatedConnection

JAXR  JAXR
See also javax.xml.registry.ConnectionFactory

This class represents a connection between a JAXR client and a JAXR provider.

Author:
  Farrukh S. Najmi
See Also:
  ConnectionFactory

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void close()</td>
</tr>
<tr>
<td>Closes a Connection when it is no longer needed.</td>
</tr>
<tr>
<td>Set getCredentials()</td>
</tr>
<tr>
<td>Gets the credentials associated with this client.</td>
</tr>
<tr>
<td>RegistryService getRegistryService()</td>
</tr>
<tr>
<td>Gets the RegistryService interface associated with the Connection.</td>
</tr>
<tr>
<td>boolean isClosed()</td>
</tr>
<tr>
<td>Indicated whether this Connection has been closed or not.</td>
</tr>
<tr>
<td>boolean isSynchronous()</td>
</tr>
<tr>
<td>Indicates whether a client uses synchronous</td>
</tr>
</tbody>
</table>
communication with JAXR provider or not.

```java
void setCredentials(Set credentials)
Sets the Credentials associated with this client.
```

```java
void setSynchronous(boolean sync)
Sets whether the client uses synchronous communication or not.
```

## Method Detail

```java
public RegistryService getRegistryService() throws JAXRException
Connection RegistryService Connection
getRegistryService Connection
RegistryService
```

```java
return RegistryService
Throws JAXRException: JAXR associates
<RegistryService>
See also javax.xml.registry.RegistryService
```

### getRegistryService

```java
RegistryService getRegistryService()
throws JAXRException
```

Gets the RegistryService interface associated with the Connection. If a Connection property (e.g. credentials) is set after the client calls `getRegistryService` then the newly set Connection property is visible to the RegistryService previously returned by this call.

**Capability Level: 0**
Returns:
the RegistryService associated with this object

 Throws:
  JAXRException - If the JAXR provider encounters an internal error

 See Also:
  RegistryService

---

public void close() throws JAXRException

Connection  Connection  JVM

0

Throws
  JAXRException: JAXR

close

void close()
  throws JAXRException

Closes a Connection when it is no longer needed. Since a provider typically allocates significant resources outside the JVM on behalf of a Connection, clients should close them when they are not needed.

Capability Level: 0

Throws:
  JAXRException - If the JAXR provider encounters an internal error

---

public boolean isClosed() throws JAXRException

Connection

0
**isClosed**

```java
boolean isClosed() throws JAXRException
```

Indicated whether this Connection has been closed or not.

**Capability Level:** 0

**Returns:**
true if Connection is closed; false otherwise

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

---

**public boolean isSynchronous() throws JAXRException**

JAXR JAXR JAXR

```java
0
```

**return**
Connection
ture false

**Throws**
JAXRException: JAXR

---

**isSynchronous**

```java
boolean isSynchronous() throws JAXRException
```

Indicates whether a client uses synchronous communication with JAXR provider or not. A JAXR provider must support both modes of communication. A JAXR client can choose which mode it wants to use.
Capability Level: 0

Returns:
true if Connection is synchronous (default); false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setSynchronous(boolean sync) throws JAXRException
JAXR
0

sync Connection true false

Throws JAXRException: JAXR

setSynchronous

void setSynchronous(boolean sync)
throws JAXRException

Sets whether the client uses synchronous communication or not. A JAXR client may dynamically change its communication style preference.

Capability Level: 0

Parameters:
sync - true if Connection is desired to be synchronous; false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error
public void setCredentials(java.util.Set<E> credentials)
throws JAXRException
Credential JAXR JAXR

0

credentials java.lang.Object Collection
Throws JAXRException: JAXR

setCredentials

void setCredentials(Set credentials)
throws JAXRException

Sets the Credentials associated with this client. The credentials is used to authenticate the client with the JAXR provider. A JAXR client may dynamically change its identity by changing the credentials associated with it.

Capability Level: 0

Parameters:
  credentials - a Collection of java.lang.Objects which provide identity related information for the caller.

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public java.util.Set<E> getCredentials() throws JAXRException

0

return java.lang.Object SetCollection null
getCredentials

Set getCredentials() throws JAXRException

Gets the credentials associated with this client.

Capability Level: 0

Returns:
Set of java.lang.Object instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error
javax.mail.event  **Class ConnectionAdapter**

java.lang.Object  
  ```
  - javax.mail.event.ConnectionAdapter
  ```

**All Implemented Interfaces:**
  
  [EventListener](https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/javax/mail/event/EventListener.html),  
  [ConnectionListener](https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/javax/mail/event/ConnectionListener.html)

---

```
public abstract class ConnectionAdapter
extends Object
implements ConnectionListener
```

**Implements:**  
  [ConnectionListener](https://docs.oracle.com/en/java/javase/11/docs/api/java.desktop/javax/mail/event/ConnectionListener.html)

---

The adapter which receives connection events. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.

**Author:**
  
  John Mani

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionAdapter()</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void closed(ConnectionEvent e)</code></td>
<td>Invoked when a Store/Folder/Transport is closed.</td>
</tr>
<tr>
<td><code>void disconnected(ConnectionEvent e)</code></td>
<td></td>
</tr>
</tbody>
</table>

---
void invoked when a Store is disconnected.

void opened (ConnectionEvent e)
Invoked when a Store/Folder/Transport is opened.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ConnectionAdapter()

ConnectionAdapter

public ConnectionAdapter()

Method Detail

public void opened (ConnectionEvent e)

opened

public void opened (ConnectionEvent e)

Description copied from interface: ConnectionListener
Invoked when a Store/Folder/Transport is opened.

Specified by:
opened in interface ConnectionListener

public void disconnected (ConnectionEvent e)
disconnected

public void disconnected(ConnectionEvent e)

Description copied from interface: ConnectionListener
Invoked when a Store is disconnected. Note that a folder cannot be disconnected, so a folder will not fire this event.

Specified by:
   disconnected in interface ConnectionListener

public void closed(ConnectionEvent e)

closed

public void closed(ConnectionEvent e)

Description copied from interface: ConnectionListener
Invoked when a Store/Folder/Transport is closed.

Specified by:
   closed in interface ConnectionListener
public interface ConnectionConsumer

Connection	ConnectionConsumer	Destination
ConnectionConsumer	ServerSessionPool

ConnectionConsumer	ServerSession
ConnectionConsumer	ServerSession

For application servers, Connection objects provide a special facility for creating a ConnectionConsumer (optional). The messages it is to consume are specified by a Destination and a message selector. In addition, a ConnectionConsumer must be given a ServerSessionPool to use for processing its messages.

Normally, when traffic is light, a ConnectionConsumer gets a ServerSession from its pool, loads it with a single message, and starts it. As traffic picks up, messages can back up. If this happens, a ConnectionConsumer can load each ServerSession with more than one message. This reduces the thread context switches and minimizes resource use at the expense of some serialization of message processing.

Version:
1.1 February 8, 2002

Author:
Mark Hapner, Rich Burridge
See Also:

Connection.createConnectionConsumer(javax.jms.Destination, java.lang.String, javax.jms.ServerSessionPool, int),
Connection.createDurableConnectionConsumer(javax.jms.Topic, java.lang.String, java.lang.String, javax.jms.ServerSessionPool, int),
QueueConnection.createConnectionConsumer(javax.jms.Queue, java.lang.String, javax.jms.ServerSessionPool, int),
TopicConnection.createDurableConnectionConsumer(javax.jms.Topic, java.lang.String, javax.jms.ServerSessionPool, int),

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>close()</td>
<td>Closes the connection consumer.</td>
</tr>
<tr>
<td>getServerSessionPool()</td>
<td>Gets the server session pool associated with this connection consumer.</td>
</tr>
</tbody>
</table>

Method Detail

public ServerSessionPool getServerSessionPool() throws JMSException

return

Throws JMSException: JMS

g.getServerSessionPool

ServerSessionPool g.getServerSessionPool() throws JMSException
Gets the server session pool associated with this connection consumer.

**Returns:**
the server session pool used by this connection consumer

**Throws:**
- `JMSException` - if the JMS provider fails to get the server session pool associated with this consumer due to some internal error.

```java
public void close() throws JMSException
```

**Java**

throws `JMSException`: JMS

**close**

```java
void close() throws JMSException
```

Closes the connection consumer.

Since a provider may allocate some resources on behalf of a connection consumer outside the Java virtual machine, clients should close these resources when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

**Throws:**
- `JMSException` - if the JMS provider fails to release resources on behalf of the connection consumer or fails to close the connection consumer.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public class ConnectionEvent extends MailEvent

Extends: java.util.EventObject > MailEvent

Connection

This class models Connection events.

Author: John Mani

See Also: Serialized Form

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static int CLOSED</strong></td>
</tr>
<tr>
<td><strong>static int DISCONNECTED</strong></td>
</tr>
<tr>
<td><strong>static int OPENED</strong></td>
</tr>
<tr>
<td><strong>protected int type</strong></td>
</tr>
</tbody>
</table>
**Fields inherited from class java.util.EventObject**

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

**Constructor Summary**

**ConnectionEvent**(Object source, int type)

Constructor

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void dispatch(Object listener)</td>
<td>Invokes the appropriate ConnectionListener method</td>
</tr>
<tr>
<td>int getType()</td>
<td>Return the type of this event</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.util.EventObject**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>getSource, toString</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

**Field Detail**

**OPENED**

public static final int OPENED

A connection was opened.

**See Also:**
Constant Field Values
DISCONNECTED

public static final int DISCONNECTED

A connection was disconnected (not currently used).

See Also:
Constant Field Values

CLOSED

public static final int CLOSED

A connection was closed.

See Also:
Constant Field Values

type

protected int type

The event type.

Constructor Detail

public ConnectionEvent(Object source, int type)
Constructor

source
ConnectionEvent

public ConnectionEvent(Object source, int type)

Constructor

Parameters:
source - The source object

Method Detail

public int getType()

return

getType

public int getType()

Return the type of this event

Returns:
type

public void dispatch(Object listener)
ConnectionListener

dispatch

public void dispatch(Object listener)
Invokes the appropriate ConnectionListener method

Specified by:  
`dispatch` in class `MailEvent`
javax.resource.spi Class ConnectionEvent

java.lang.Object
   ▼ java.util.EventObject
      ▼ javax.resource.spi.ConnectionEvent

All Implemented Interfaces:
   Serializable

public class ConnectionEvent
extends EventObject

Extends: java.util.EventObject

ConnectionEvent ConnectionEvent

  • ManagedConnection ConnectionEvent.getSource
    ManagedConnection
  • ManagedConnection CONNECTION_CLOSED
  • CONNECTION_ERROR_OCCURRED

  • CONNECTION_CLOSED
  • LOCAL_TRANSACTION_STARTED
  • LOCAL_TRANSACTION_COMMITTED
  • LOCAL_TRANSACTION_ROLLED_BACK
  • CONNECTION_ERROR_OCCURRED

version 0.5
See also javax.resource.spi.ConnectionEventListener

The ConnectionEvent class provides information about the source of a connection related event. A ConnectionEvent instance contains the following information:
- Type of the connection event
- ManagedConnection instance that generated the connection event. A ManagedConnection instance is returned from the method ConnectionEvent.getSource.
- Connection handle associated with the ManagedConnection instance; required for the CONNECTION_CLOSED event and optional for the other event types.
- Optionally, an exception indicating the connection related error. Note that exception is used for CONNECTION_ERROR_OCCURRED.

This class defines following types of event notifications:

- CONNECTION_CLOSED
- LOCAL_TRANSACTION_STARTED
- LOCAL_TRANSACTION_COMMITTED
- LOCAL_TRANSACTION_ROLLED_BACK
- CONNECTION_ERROR_OCCURRED

Version:
0.5

Author:
Rahul Sharma

See Also:
ConnectionEventListener, Serialized Form

**Field Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>CONNECTION_CLOSED</td>
</tr>
<tr>
<td></td>
<td>Event notification that an application component has closed the connection</td>
</tr>
<tr>
<td>static int</td>
<td>CONNECTION_ERROR_OCCURRED</td>
</tr>
<tr>
<td></td>
<td>Event notification that an error occurred on the connection.</td>
</tr>
<tr>
<td>protected int</td>
<td>id</td>
</tr>
<tr>
<td></td>
<td>Type of the event</td>
</tr>
<tr>
<td>static int</td>
<td>LOCAL_TRANSACTION_COMMITTED</td>
</tr>
<tr>
<td></td>
<td>Event notification that a Resource Manager Local Transaction has committed</td>
</tr>
</tbody>
</table>
Transaction was committed on the connection

<table>
<thead>
<tr>
<th>static int</th>
<th>LOCAL_TRANSACTION_ROLLED_BACK</th>
</tr>
</thead>
</table>
| Event notification that a Resource Manager Local Transaction was rolled back on the connection

<table>
<thead>
<tr>
<th>static int</th>
<th>LOCAL_TRANSACTION_STARTED</th>
</tr>
</thead>
</table>
| Event notification that a Resource Manager Local Transaction was started on the connection

Fields inherited from class java.util.EventObject

source

Constructor Summary

**ConnectionEvent** *(ManagedConnection source, int eid)*
Construct a ConnectionEvent object.

**ConnectionEvent** *(ManagedConnection source, int eid, Exception exception)*
Construct a ConnectionEvent object.

Method Summary

<table>
<thead>
<tr>
<th>Object</th>
<th>getConnectionHandle()</th>
</tr>
</thead>
</table>
| Get the connection handle associated with the Managed Connection instance.

<table>
<thead>
<tr>
<th>Exception</th>
<th>getException()</th>
</tr>
</thead>
</table>
| Get the exception.

<table>
<thead>
<tr>
<th>int</th>
<th>getId()</th>
</tr>
</thead>
</table>
| Get the type of event

<table>
<thead>
<tr>
<th>void</th>
<th>setConnectionHandle(Object connectionHandle)</th>
</tr>
</thead>
</table>
| Set the connection handle.

Methods inherited from class java.util.EventObject

ggetSource, toString

Methods inherited from class java.lang.Object
Field Detail

CONNECTION_CLOSED

public static final int CONNECTION_CLOSED

Event notification that an application component has closed the connection

See Also:
  Constant Field Values

LOCAL_TRANSACTION_STARTED

public static final int LOCAL_TRANSACTION_STARTED

Event notification that a Resource Manager Local Transaction was started on the connection

See Also:
  Constant Field Values

LOCAL_TRANSACTION_COMMITTED

public static final int LOCAL_TRANSACTION_COMMITTED

Event notification that a Resource Manager Local Transaction was
committed on the connection

See Also:
Constant Field Values

LOCAL_TRANSACTION_ROLLED_BACK

public static final int LOCAL_TRANSACTION_ROLLED_BACK

Event notification that a Resource Manager Local Transaction was rolled back on the connection

See Also:
Constant Field Values

CONNECTION_ERROR_OCCURRED

public static final int CONNECTION_ERROR_OCCURRED

Event notification that an error occurred on the connection. This event indicates that the ManagedConnection instance is now invalid and unusable.

See Also:
Constant Field Values

id

protected int id

Type of the event
public ConnectionEvent(ManagedConnection source, int eid)

ConnectionEvent null

source ManagedConnection

eid Connection

ConnectionEvent

public ConnectionEvent(ManagedConnection source, int eid)

Construct a ConnectionEvent object. Exception defaults to null.

Parameters:

source - ManagedConnection that is the source of the event
eid - type of the Connection event

ConnectionEvent

public ConnectionEvent(ManagedConnection source, int eid, Exception exception)

ConnectionEvent

public ConnectionEvent(ManagedConnection source, int eid, Exception exception)

Construct a ConnectionEvent object.
Parameters:
  source - ManagedConnection that is the source of the event
  exception - exception about to be thrown to the application
  eid - type of the Connection event

Method Detail

public Object getConnectionHandle()
ManagedConnection CONNECTION_CLOSED

    return null

getConnectionHandle

public Object getConnectionHandle()

    Get the connection handle associated with the Managed Connection instance. Used for CONNECTION_CLOSED event.

    Returns:
    the connection handle. May be null

public void setConnectionHandle(Object connectionHandle)
    CONNECTION_CLOSED

setConnectionHandle

public void setConnectionHandle(Object connectionHandle)

    Set the connection handle. Used for CONNECTION_CLOSED event
public Exception getException()
    null
    return

getException

public Exception getException()

    Get the exception. May be null.

    Returns:
    the exception about to be thrown.

public int getId()

getId

public int getId()

    Get the type of event
<table>
<thead>
<tr>
<th>Summary</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Detail</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
</table>

PREV CLASS    NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTRUCTOR | METHOD

FRAMES    NO FRAMES
DETAIL: FIELD | CONSTRUCTOR | METHOD
The `ConnectionEventListener` interface provides an event callback mechanism to enable an application server to receive notifications from a `ManagedConnection` instance.

An application server uses these event notifications to manage its connection pool, to clean up any invalid or terminated connections and to manage local transactions.

An application server implements the `ConnectionEventListener` interface. It registers a connection listener with a `ManagedConnection` instance by using `ManagedConnection.addConnectionEventListener` method.

**Version:**

0.5
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>connectionClosed</code></td>
<td>Notifies that an application component has closed the connection.</td>
</tr>
<tr>
<td><code>connectionErrorOccurred</code></td>
<td>Notifies a connection related error.</td>
</tr>
<tr>
<td><code>localTransactionCommitted</code></td>
<td>Notifies that a Resource Manager Local Transaction was committed on the ManagedConnection instance.</td>
</tr>
<tr>
<td><code>localTransactionRolledback</code></td>
<td>Notifies that a Resource Manager Local Transaction was rolled back on the ManagedConnection instance.</td>
</tr>
<tr>
<td><code>localTransactionStarted</code></td>
<td>Notifies that a Resource Manager Local Transaction was started on the ManagedConnection instance.</td>
</tr>
</tbody>
</table>

Method Detail

```java
public void connectionClosed(ConnectionEvent event)

ManagedConnection
ConnectionEventListener.connectionClosed
ManagedConnection

event
```
connectionClosed

```java
void connectionClosed(ConnectionEvent event)
```

Notifies that an application component has closed the connection.

A ManagedConnection instance notifies its registered set of listeners by calling `ConnectionEventListener.connectionClosed` method when an application component closes a connection handle. The application server uses this connection close event to put the ManagedConnection instance back in to the connection pool.

**Parameters:**

- **event** - event object describing the source of the event

---

public void localTransactionStarted(ConnectionEvent event)

```java
ManagedConnection event
```

localTransactionStarted

```java
void localTransactionStarted(ConnectionEvent event)
```

Notifies that a Resource Manager Local Transaction was started on the ManagedConnection instance.

**Parameters:**

- **event** - event object describing the source of the event

---

public void localTransactionCommitted(ConnectionEvent event)

```java
ManagedConnection event
```

**localTransactionCommitted**

```java
void localTransactionCommitted(ConnectionEvent event)
```

Notifies that a Resource Manager Local Transaction was committed on the ManagedConnection instance.
**localTransactionCommitted**

```java
void localTransactionCommitted(ConnectionEvent event)
```

Notifies that a Resource Manager Local Transaction was committed on the ManagedConnection instance.

**Parameters:**
- `event` - event object describing the source of the event

---

**public void localTransactionRolledback(ConnectionEvent event)**

ManagedConnection

```java
void localTransactionRolledback(ConnectionEvent event)
```

Notifies that a Resource Manager Local Transaction was rolled back on the ManagedConnection instance.

**Parameters:**
- `event` - event object describing the source of the event

---

**public void connectionErrorOccurred(ConnectionEvent event)**

ManagedConnection

```java
ConnectionEventListener.connectionErrorOccurred
```

```java
connectionErrorOccurred
```

```java
ManagedConnection
```

```java
ManagedConnection.destroy
```

```java
event
```
connectionErrorOccurred

void connectionErrorOccurred(ConnectionEvent event)

Notifies a connection related error. The ManagedConnection instance calls the method ConnectionEventListener.connectionErrorOccurred to notify its registered listeners of the occurrence of a physical connection-related error. The event notification happens just before a resource adapter throws an exception to the application component using the connection handle. The connectionErrorOccurred method indicates that the associated ManagedConnection instance is now invalid and unusable. The application server handles the connection error event notification by initiating application server-specific cleanup (for example, removing ManagedConnection instance from the connection pool) and then calling ManagedConnection.destroy method to destroy the physical connection.

Parameters:
  
  event - event object describing the source of the event
javax.jms Interface ConnectionFactory

All Known Subinterfaces:
QueueConnectionFactory, TopicConnectionFactory, XAQueueConnectionFactory, XATopicConnectionFactory

public interface ConnectionFactory

Implemented by: QueueConnectionFactory, TopicConnectionFactory

ConnectionFactory JMS
ConnectionFactory JMS

JMS JMS JMS API

Java Naming and Directory Interface (JNDI) API JMS API JMS JNDI

JMS API

JMS JNDI JMS javax.jndi.Referenceable java.io.Serializable JNDI JavaBeans

• JMS
• Java “Java ”
• JNDI JMS

JNDI API

Java

version 1.1 - February 1, 2002

See also javax.jms.Connection, javax.jms.QueueConnectionFactory, javax.jms.TopicConnectionFactory
A `ConnectionFactory` object encapsulates a set of connection configuration parameters that has been defined by an administrator. A client uses it to create a connection with a JMS provider.

A `ConnectionFactory` object is a JMS administered object and supports concurrent use.

JMS administered objects are objects containing configuration information that are created by an administrator and later used by JMS clients. They make it practical to administer the JMS API in the enterprise.

Although the interfaces for administered objects do not explicitly depend on the Java Naming and Directory Interface (JNDI) API, the JMS API establishes the convention that JMS clients find administered objects by looking them up in a JNDI namespace.

An administrator can place an administered object anywhere in a namespace. The JMS API does not define a naming policy.

It is expected that JMS providers will provide the tools an administrator needs to create and configure administered objects in a JNDI namespace. JMS provider implementations of administered objects should be both `javax.jndi.Referenceable` and `java.io.Serializable` so that they can be stored in all JNDI naming contexts. In addition, it is recommended that these implementations follow the JavaBeans™ design patterns.

This strategy provides several benefits:

- It hides provider-specific details from JMS clients.
- It abstracts administrative information into objects in the Java programming language ("Java objects") that are easily organized and administered from a common management console.
- Since there will be JNDI providers for all popular naming services, this means that JMS providers can deliver one implementation of administered objects that will run everywhere.
An administered object should not hold on to any remote resources. Its lookup should not use remote resources other than those used by the JNDI API itself.

Clients should think of administered objects as local Java objects. Looking them up should not have any hidden side effects or use surprising amounts of local resources.

Version:
1.1 - February 1, 2002

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
Connection, QueueConnectionFactory, TopicConnectionFactory

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection createConnection()</td>
<td>Creates a connection with the default user identity.</td>
</tr>
<tr>
<td>Connection createConnection(String userName, String password)</td>
<td>Creates a connection with the specified user identity.</td>
</tr>
</tbody>
</table>

---

### Method Detail

```java
public Connection createConnection() throws JMSException
```

    return Connection.state

    @Override
    throws JMSException
```

createConnection
Connection createConnection() throws JMSException

Creates a connection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

Returns:
a newly created connection

Throws:

JMSException - if the JMS provider fails to create the connection due to some internal error.
JMSSecurityException - if client authentication fails due to an invalid user name or password.

Since: 1.1

public Connection createConnection(String userName, String password) throws JMSException

userName
password
return

Throws JMSException: JMS

Throws JMSSecurityException:

since 1.1

createConnection

Connection createConnection(String userName, String password) throws JMSException

Creates a connection with the specified user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.
**Parameters:**
- `userName` - the caller's user name
- `password` - the caller's password

**Returns:**
a newly created connection

**Throws:**
- `JMSException` - if the JMS provider fails to create the connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

**Since:**
1.1

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.resource.cci  Interface ConnectionFactory

All Superinterfaces:
   Referenceable, Serializable

public interface ConnectionFactory
extends Serializable, Referenceable

Implements: java.io.Serializable, Referenceable

ConnectionFactory  EIS ConnectionFactory

JNDI  ConnectionFactory  EIS

ConnectionFactory  java.io.Serializable
javax.resource.Referenceable  JNDI

version 0.8
See javax.resource.cci.Connection,
also javax.resource.Referenceable

ConnectionFactory provides an interface for getting connection to an EIS instance. An implementation of ConnectionFactory interface is provided by a resource adapter.

Application code looks up a ConnectionFactory instance from JNDI namespace and uses it to get EIS connections.

An implementation class for ConnectionFactory is required to implement java.io.Serializable and javax.resource.Referenceable interfaces to support JNDI registration.

Version:
   0.8

Author:
See Also:

Connection, Referenceable

### Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>getConnection()</td>
<td>Gets a connection to an EIS instance.</td>
</tr>
<tr>
<td>Connection</td>
<td>getConnection()</td>
<td>Gets a connection to an EIS instance. (ConnectionSpec properties)</td>
</tr>
<tr>
<td>RecordFactory</td>
<td>getRecordFactory()</td>
<td>Gets a RecordFactory instance.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.resource.Referenceable

setReference

Methods inherited from interface javax.naming.Referenceable

getReference

### Method Detail

```java
public Connection getConnection() throws ResourceException

EIS EIS getConnection

return Connection

ResourceException: EIS

- ManagedConnectionFactory —
- ...

Throws
- EIS —EIS
- ...
```
getConnection

Connection getConnection() throws ResourceException

Gets a connection to an EIS instance. This getConnection variant should be used when a component wants the container to manage EIS sign-on. This case is termed container-managed sign-on. The component does not pass any security information.

Returns:
Connection instance

Throws:
ResourceException - Failed to get a connection to the EIS instance. Examples of error cases are:
- Invalid configuration of ManagedConnectionFactory--example: invalid server name
- Application server-internal error--example: connection pool related error
- Communication error
- EIS-specific error--example: EIS not active
- Resource adapter-internal error
- Security related error; example: invalid user
- Failure to allocate system resources

public Connection getConnection(ConnectionSpec properties) throws ResourceException
EIS javax.resource.cci.ConnectionSpec getConnection ConnectionSpec

getConnection EIS
EIS ManagedConnectionFactory
getConnection

Get a connection to an EIS instance. A component should use the getConnection variant with javax.resource.cci.ConnectionSpec parameter, if it needs to pass any resource adapter specific security information and connection parameters. In the component-managed sign-on case, an application component passes security information (example: username, password) through the ConnectionSpec instance.

It is important to note that the properties passed through the getConnection method should be client-specific (example: username, password, language) and not related to the configuration of a target EIS instance (example: port number, server name). The ManagedConnectionFactory instance is configured with complete set of properties required for the creation of a connection to an EIS instance.

**Parameters:**

- properties - Connection parameters and security information

See also [javax.resource.cci.ConnectionSpec](https://docs.oracle.com/en/java/javase/11/docs/api/jdk.xml/javax/resource/cdi/ConnectionSpec.html)
specified as ConnectionSpec instance

**Returns:**
Connection instance

**Throws:**
- `ResourceException` - Failed to get a connection to the EIS instance. Examples of error cases are:
  - Invalid specification of input parameters
  - Invalid configuration of ManagedConnectionFactory--example: invalid server name
  - Application server-internal error--example: connection pool related error
  - Communication error
  - EIS-specific error--example: EIS not active
  - Resource adapter-internal error
  - Security related error; example: invalid user
  - Failure to allocate system resources

**See Also:**
- `ConnectionSpec`

---

```java
public RecordFactory getRecordFactory() throws ResourceException
```

**RecordFactory**  
**Record**

**return**  
RecordFactory RecordFactory

**Throws**  
- `ResourceException`: RecordFactory

**Throws**  
- `NotSupportedException`:

---

**getRecordFactory**

```java
RecordFactory getRecordFactory() throws ResourceException
```

Gets a RecordFactory instance. The RecordFactory is used for the creation of generic Record instances.

**Returns:**
RecordFactory RecordFactory instance
public ResourceAdapterMetaData getMetaData() throws ResourceException

getMetaData

ResourceAdapterMetaData getMetaData() throws ResourceException

Gets metadata for the Resource Adapter. Note that the metadata information is about the ResourceAdapter and not the EIS instance. An invocation of this method does not require that an active connection to an EIS instance should have been established.

Returns:
ResourceAdapterMetaData instance

Throws:
ResourceException - Failed to get metadata information about the resource adapter
PS:
javax.xml.registry Class ConnectionFactory

java.lang.Object
   \ javax.xml.registry.ConnectionFactory

public abstract class ConnectionFactory
extends Object

JAXR JAXR ConnectionFactory
JNDI API  ConnectionFactory

JAXR ConnectionFactory  Java Naming and Directory Interface (JNDI) API ConnectionFactory  JNDI API ConnectionFactory  ConnectionFactory  ConnectionFactory
JNDI API  ConnectionFactory

JAXR API  JAXR ConnectionFactory  JNDI API  JAXR API
ConnectionFactory  newInstance  newInstance  JAXR
ConnectionFactory  javax.xml.registry.ConnectionFactoryClass
newInstance  JAXR  ConnectionFactory

See also  javax.xml.registry.Connection

This is the abstract base class for factory classes for creating a JAXR connection. A JAXR ConnectionFactory object is configured in a provider-specific way to create connections with registry providers.
Looking Up a ConnectionFactory Using the JNDI API

The preferred way for a client to look up a JAXR ConnectionFactory is within the Java Naming and Directory Interface™ (JNDI) API. A ConnectionFactory object is registered with a naming service in a provider specific way, such as one based on the JNDI API. This registration associates the ConnectionFactory object with a logical name. When an application wants to establish a connection with the provider associated with that ConnectionFactory object, it does a lookup, providing the logical name. The application can then use the ConnectionFactory object that is returned to create a connection to the messaging provider.
Looking Up a ConnectionFactory Without Using the JNDI API

The JAXR API provides an alternative way to look up a JAXR ConnectionFactory that does not require the use of the JNDI API. This is done using the newInstance static method on the abstract class ConnectionFactory provided in the JAXR API. The newInstance method returns a JAXR ConnectionFactory. The client may indicate which factory class should be instantiated by the newInstance method by defining the system property javax.xml.registry.ConnectionFactoryClass. If this property is not set, the JAXR provider must return a default ConnectionFactory instance.

Author:  
Farrukh S. Najmi

See Also:  
Connection

## Constructor Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionFactory</td>
<td>()</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract Connection createConnection()</td>
<td>Create a named connection.</td>
</tr>
<tr>
<td>abstract FederatedConnection createFederatedConnection(Collection connections)</td>
<td>Create a FederatedConnection.</td>
</tr>
<tr>
<td>abstract Properties getProperties()</td>
<td>Gets the Properties used during createConnection and createFederatedConnection calls.</td>
</tr>
<tr>
<td>static ConnectionFactory newInstance()</td>
<td>Creates a default ConnectionFactory object.</td>
</tr>
</tbody>
</table>
abstract void setProperties(Properties properties)  
Sets the Properties used during createConnection and createFederatedConnection calls.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ConnectionFactory()

ConnectionFactory

public ConnectionFactory()

Method Detail

abstract public void setProperties(java.util.Properties properties) throws JAXRException
createConnection  createFederatedConnection

javax.xml.registry.queryManagerURL - URL String
javax.xml.registry.lifeCycleManagerURL - URL String queryManagerURL
javax.xml.registry.semanticEquivalences - String
javax.xml.registry.security.authenticationMethod - JAXR String
"UDDI_GET_AUTHTOKEN""HTTP_BASIC""CLIENT_ID"
"MS_PASSPORT"
javax.xml.registry.uddi.maxRows - UDDI
javax.xml.registry.postalAddressScheme - ClassificationScheme
ID String
ClassificationScheme

0

properties JAXR
Throws JAXRException: JAXR

setProperties

public abstract void setProperties(Properties properties) throws JAXRException

Sets the Properties used during createConnection and createFederatedConnection calls.

Standard Connection Properties:
javax.xml.registry.queryManagerURL - URL String for the query manager service within the target registry provider
javax.xml.registry.lifeCycleManagerURL - URL String for the life cycle manager service within the target registry provider. If unspecified, must default to value of the queryManagerURL described above
javax.xml.registry.semanticEquivalences - String that allows specification of semantic equivalences
javax.xml.registry.security.authenticationMethod - string that provides a hint to the JAXR provider on the authentication method to be used when authenticating with the registry provider. Possible value include but are not limited to "UDDI_GET_AUTHTOKEN", "HTTP_BASIC", "CLIENT_CERTIFICATE", "MS_PASSPORT"
javax.xml.registry.uddi.maxRows - integer that specifies the maximum number of rows to be returned for find operations. This property is specific for UDDI providers
javax.xml.registry.postalAddressScheme - String that specifies
the id of a ClassificationScheme that is used as the default postal address scheme for this connection

**Capability Level:** 0

**Parameters:**
- **properties** - configuration properties that are either specified by JAXR specification or are provider specific.

**Throws:**
- [JAXRException](#) - If the JAXR provider encounters an internal error

---

```java
abstract public java.util.Properties getProperties() throws JAXRException
createConnection  createFederatedConnection

0

return

Throws               JAXRException:  JAXR
```

**getProperties**

```java
public abstract Properties getProperties() throws JAXRException
```

Gets the Properties used during createConnection and createFederatedConnection calls.

**Capability Level:** 0

**Returns:**
- the Properties defined for this object

**Throws:**
- [JAXRException](#) - If the JAXR provider encounters an internal error

---
abstract public Connection createConnection() throws JAXRException

JAXR

0

return Connection

Throws JAXRException: JAXR

link dependency

label creates

associates <{Connection}>

createConnection

public abstract Connection createConnection() throws JAXRException

Create a named connection. Such a connection can be used to communicate with a JAXR provider.

Capability Level: 0

Returns:
the Connection created by this call

Throws:
JAXRException - If the JAXR provider encounters an internal error

abstract public FederatedConnection createFederatedConnection(java.util.Collection<E> connections) throws JAXRException

FederatedConnection
### createFederatedConnection

```java
public abstract FederatedConnection createFederatedConnection(Collection<Connection> connections) throws JAXRException;
```

Create a FederatedConnection.

**Capability Level: 0 (optional)**

**Parameters:**
- `connections`: Is a Collection of Connection objects. Note that Connection objects may also be FederatedConnection objects.

**Returns:**
- the FederatedConnection created by this call

**Throws:**
- `JAXRException`: If the JAXR provider encounters an internal error

---

### newInstance

```java
public static ConnectionFactory newInstance() throws JAXRException
```

```java
ConnectionFactory return
```

```java
Throws JAXRException: JAXR
```
newInstance

public static ConnectionFactory newInstance() throws JAXRException

Creates a default ConnectionFactory object.

Returns:
a new instance of a ConnectionFactory

Throws:
JAXRException - If the JAXR provider encounters an internal error
<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL:</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>INDEX</th>
<th>Help</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>OVERVIEW</th>
<th>PACKAGE</th>
<th>TREE</th>
<th>DEPRECATED</th>
<th>INDEX</th>
<th>HELP</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>FRAME</th>
<th>NO FRAME</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL:</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>FRAME</th>
<th>NO FRAME</th>
</tr>
</thead>
</table>
javax.mail.event  Interface ConnectionListener

All Superinterfaces:  
EventListener

All Known Implementing Classes:  
ConnectionAdapter

class ConnectionListener  
extends EventListener

Implements: java.util.EventListener
Implemented by: ConnectionAdapter

Connection Listener

This is the Listener interface for Connection events.

Author:  
John Mani

<table>
<thead>
<tr>
<th>Method Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>void <strong>closed</strong> <em>(ConnectionEvent e)</em></td>
<td>Invoked when a Store/Folder/Transport is closed.</td>
</tr>
<tr>
<td>void <strong>disconnected</strong> <em>(ConnectionEvent e)</em></td>
<td>Invoked when a Store is disconnected.</td>
</tr>
<tr>
<td>void <strong>opened</strong> <em>(ConnectionEvent e)</em></td>
<td>Invoked when a Store/Folder/Transport is opened.</td>
</tr>
</tbody>
</table>

Method Detail
public void opened(\texttt{ConnectionEvent} e) 
\textit{Store/Folder/Transport}

\textbf{opened} 

\texttt{void opened(\texttt{ConnectionEvent} e)}

Invoked when a Store/Folder/Transport is opened.

\textbf{public void disconnected(\texttt{ConnectionEvent} e) 
\textit{Store}}

\textbf{disconnected} 

\texttt{void disconnected(\texttt{ConnectionEvent} e)}

Invoked when a Store is disconnected. Note that a folder cannot be disconnected, so a folder will not fire this event.

\textbf{public void closed(\texttt{ConnectionEvent} e) 
\textit{Store/Folder/Transport}}

\textbf{closed} 

\texttt{void closed(\texttt{ConnectionEvent} e)}

Invoked when a Store/Folder/Transport is closed.
javax.resource.spi  Interface ConnectionManager

All Superinterfaces:
   Serializable

public interface ConnectionManager
extends Serializable

Implements: java.io.Serializable

ConnectionManager
ConnectionManager
ConnectionManager  (QoS) - /
EIS
ManagedConnectionFactory
ConnectionManager  java.io.Serializable
ConnectionManager  ConnectionManager
QOS

since 0.6
See also javax.resource.spi.ManagedConnectionFactory

ConnectionManager interface provides a hook for the resource adapter to pass a connection request to the application server.

An application server provides implementation of the ConnectionManager interface. This implementation is not specific to any particular type of the resource adapter or connection factory interface.
The ConnectionManager implementation delegates to the application server to enable latter to provide quality of services (QoS) - security, connection pool management, transaction management and error logging/tracing.

An application server implements these services in a generic manner, independent of any resource adapter and EIS specific mechanisms. The connector architecture does not specify how an application server implements these services; the implementation is specific to an application server.

After an application server hooks-in its services, the connection request gets delegated to a ManagedConnectionFactory instance either for the creation of a new physical connection or for the matching of an already existing physical connection.

An implementation class for ConnectionManager interface is required to implement the java.io.Serializable interface.

In the non-managed application scenario, the ConnectionManager implementation class can be provided either by a resource adapter (as a default ConnectionManager implementation) or by application developers. In both cases, QOS can be provided as components by third party vendors.

Since: 0.6
Author: Rahul Sharma
See Also: ManagedConnectionFactory

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocateConnection(ManagedConnectionFactory mcf, ConnectionRequestInfo cxRequestInfo)</td>
</tr>
<tr>
<td>The method allocateConnection gets called by the resource adapter's connection factory instance.</td>
</tr>
</tbody>
</table>
The method allocateConnection gets called by the resource adapter’s connection factory instance. This lets connection factory instance (provided by the resource adapter) pass a connection request to the ConnectionManager instance.

The connectionRequestInfo parameter represents information
specific to the resource adapter for handling of the connection request.

**Parameters:**
- `mcf` - used by application server to delegate connection matching/creation
- `cxRequestInfo` - connection request Information

**Returns:**
connection handle with an EIS specific connection interface.

**Throws:**
- `ResourceException` - Generic exception
- `ApplicationServerInternalException` - Application server specific exception
- `SecurityException` - Security related error
- `ResourceAllocationException` - Failed to allocate system resources for connection request
- `ResourceAdapterInternalException` - Resource adapter related error condition
javax.jms Interface ConnectionMetaData

public interface ConnectionMetaData

ConnectionMetaData Connection

version 1.0 - 13 March 1998

A ConnectionMetaData object provides information describing the Connection object.

Version:
1.0 - 13 March 1998
Author:
Mark Hapner, Rich Burridge

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getJMSMajorVersion()</td>
<td>Gets the JMS major version number.</td>
</tr>
<tr>
<td>int getJMSMinorVersion()</td>
<td>Gets the JMS minor version number.</td>
</tr>
<tr>
<td>String getJMSProviderName()</td>
<td>Gets the JMS provider name.</td>
</tr>
<tr>
<td>String getJMSVersion()</td>
<td>Gets the JMS API version.</td>
</tr>
<tr>
<td>Enumeration getJMSXPropertyName()</td>
<td>Gets an enumeration of the JMSX property names.</td>
</tr>
<tr>
<td>int getProviderMajorVersion()</td>
<td>Gets the JMS provider major version number.</td>
</tr>
<tr>
<td>int getProviderMinorVersion()</td>
<td>Gets the JMS provider minor version number.</td>
</tr>
</tbody>
</table>
public String getProviderVersion() throws JMSException

Gets the JMS provider version.

Method Detail

public String getJMSVersion() throws JMSException

JMS API

    return JMS API

    Throws JMSException: JMS

getJMSVersion

String getJMSVersion()

Throws JMSException

Gets the JMS API version.

Returns: the JMS API version

Throws: JMSException - if the JMS provider fails to retrieve the metadata due to some internal error.

public int getJMSMajorVersion() throws JMSException

JMS

    return JMS API

    Throws JMSException: JMS

getJMSMajorVersion

int getJMSMajorVersion()

Throws JMSException
Gets the JMS major version number.

Returns:  
the JMS API major version number

Throws:  
JMSException - if the JMS provider fails to retrieve the metadata due to some internal error.

```
public int getJMSMinorVersion() throws JMSException

return JMS API

Throws JMSException: JMS
```

getJMSMinorVersion

int getJMSMinorVersion() throws JMSException

Gets the JMS minor version number.

Returns:  
the JMS API minor version number

Throws:  
JMSException - if the JMS provider fails to retrieve the metadata due to some internal error.

```
public String getJMSProviderName() throws JMSException

return JMS

Throws JMSException: JMS
```

getJMSProviderName

String getJMSProviderName()
Throws: `JMSException` - if the JMS provider fails to retrieve the metadata due to some internal error.

```java
public String getProviderVersion() throws JMSException
```

getProviderVersion

```java
public int getProviderMajorVersion() throws JMSException
```

getProviderMajorVersion
int getProviderMajorVersion() throws JMSException

Gets the JMS provider major version number.

**Returns:**
the JMS provider major version number

**Throws:**
JMSException - if the JMS provider fails to retrieve the metadata
due to some internal error.

---

public int getProviderMinorVersion() throws JMSException

JMS

return JMS

**Throws**
JMSException: JMS

---

getProviderMinorVersion

int getProviderMinorVersion() throws JMSException

Gets the JMS provider minor version number.

**Returns:**
the JMS provider minor version number

**Throws:**
JMSException - if the JMS provider fails to retrieve the metadata
due to some internal error.

---

public java.util Enumeration<E> getJMSXPropertyNames() throws JMSException

JMSX

return JMSX

**Throws**
JMSException: JMS
getJMSXPropertyNames

Enumeration getJMSXPropertyNames() throws JMSException

Gets an enumeration of the JMSX property names.

Returns:
   an Enumeration of JMSX property names

Throws:
   JMSException - if the JMS provider fails to retrieve the metadata
due to some internal error.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
public interface ConnectionMetaData

ConnectionMetaData Connection EIS
Connection.getMetaData ConnectionMetaData

version 0.8
See javax.resource.cci.Connection,
also javax.resource.cci.ResultSetInfo

The interface ConnectionMetaData provides information about an EIS instance connected through a Connection instance. A component calls the method Connection.getMetaData to get a ConnectionMetaData instance.

Version:
0.8
Author:
Rahul Sharma
See Also:
Connection, ResultSetInfo

Method Summary

<table>
<thead>
<tr>
<th>String</th>
<th>getEISProductName()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns product name of the underlying EIS instance connected through the Connection that produced this metadata.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getEISProductVersion()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns product version of the underlying EIS instance.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getUserName()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the user name for an active connection as known to the underlying EIS instance.</td>
<td></td>
</tr>
</tbody>
</table>
public String getEISProductName() throws ResourceException

Connection EIS

    return EIS

Throws ResourceException: EIS

getEISProductName

String getEISProductName() throws ResourceException

    Returns product name of the underlying EIS instance connected through the Connection that produced this metadata.

    Returns:
    Product name of the EIS instance

    Throws:
    ResourceException - Failed to get the information for the EIS instance

public String getEISProductVersion() throws ResourceException

EIS

    return EIS

Throws ResourceException: EIS

getEISProductVersion

String getEISProductVersion() throws ResourceException

    Returns product name of the underlying EIS instance connected through the Connection that produced this metadata.

    Returns:
    Product name of the EIS instance

    Throws:
    ResourceException - Failed to get the information for the EIS instance
Returns product version of the underlying EIS instance.

**Returns:**

Product version of an EIS instance.

**Throws:**

ResourceException - Failed to get the information for the EIS instance

---

public String getUserName() throws ResourceException

```java
EIS EIS
return
```

**Throws**

ResourceException: EIS

---

### getUserName

**String** getUserName()

```java
throws ResourceException
```

Returns the user name for an active connection as known to the underlying EIS instance. The name corresponds the resource principal under whose security context a connection to the EIS instance has been established.

**Returns:**

String representing the user name

**Throws:**

ResourceException - Failed to get the information for the EIS instance
PS:
javax.resource.spi Interface ConnectionRequestInfo

public interface ConnectionRequestInfo

ConnectionRequestInfo

ID match/createManagedConnection

ConnectionRequestInfo ManagedConnectionFactory
match/createManagedConnection

version 0.8

See javax.resource.spi.ManagedConnectionFactory,
also javax.resource.spi.ManagedConnection

The ConnectionRequestInfo interface enables a resource adapter to pass its own request specific data structure across the connection request flow. A resource adapter extends the empty interface to supports its own data structures for connection request.

A typical use allows a resource adapter to handle application component specified per-connection request properties (example - client ID, language). The application server passes these properties back across to match/createManagedConnection calls on the resource adapter. These properties remain opaque to the application server during the connection request flow.

Once the ConnectionRequestInfo reaches match/createManagedConnection methods on the ManagedConnectionFactory instance, resource adapter uses this additional per-request information to do connection creation and matching.

Version:
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object other)</td>
<td>Checks whether this instance is equal to another.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Returns the hashCode of the ConnectionRequestInfo.</td>
</tr>
</tbody>
</table>

Method Detail

public boolean equals(Object other)

connectionRequestInfo

    return true

equals

boolean equals(Object other)

Checks whether this instance is equal to another. Since connectionRequestInfo is defined specific to a resource adapter, the resource adapter is required to implement this method. The conditions for equality are specific to the resource adapter.

Overrides:

equals in class Object

Returns:

True if the two instances are equal.
public int hashCode()
ConnectionRequestInfo

return

hashCode

int hashCode()

Returns the hashCode of the ConnectionRequestInfo.

Overrides:
hashCode in class Object

Returns:
hashCode of this instance

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface ConnectionSpec

ConnectionSpec  ConnectionFactory.getConnection

ConnectionSpec  JavaBean  ConnectionSpec

CCI  ConnectionSpec  ConnectionSpec  EIS

See also  javax.resource.cci.ConnectionFactory

ConnectionSpec is used by an application component to pass connection request-specific properties to the ConnectionFactory. getConnection method.

It is recommended that the ConnectionSpec interface be implemented as a JavaBean to support tools. The properties on the ConnectionSpec implementation class must be defined through the getter and setter methods pattern.

The CCI specification defines a set of standard properties for an ConnectionSpec. The properties are defined either on a derived interface or an implementation class of an empty ConnectionSpec interface. In addition, a resource adapter may define additional properties specific to its underlying EIS.

Version:
  1.0 Public Draft 1

Author:
  Rahul Sharma

See Also:
  ConnectionFactory
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public class ContentDisposition
extends Object

This class represents a MIME ContentDisposition value. It provides methods to parse a ContentDisposition string into individual components and to generate a MIME style ContentDisposition string.

Version: 1.9, 07/05/04
Author: John Mani

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentDisposition()</td>
</tr>
<tr>
<td>No-arg Constructor.</td>
</tr>
<tr>
<td>ContentDisposition(String s)</td>
</tr>
<tr>
<td>Constructor that takes a ContentDisposition string.</td>
</tr>
<tr>
<td>ContentDisposition(String disposition, ParameterList list)</td>
</tr>
<tr>
<td>Constructor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><code>String getDisposition()</code></td>
</tr>
<tr>
<td><code>String getParameter(String name)</code></td>
</tr>
<tr>
<td><code>ParameterList getParameterList()</code></td>
</tr>
<tr>
<td><code>void setDisposition(String disposition)</code></td>
</tr>
<tr>
<td><code>void setParameter(String name, String value)</code></td>
</tr>
<tr>
<td><code>void setParameterList(ParameterList list)</code></td>
</tr>
<tr>
<td><code>String toString()</code></td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

**Constructor Detail**

**public ContentDisposition()**

ContentDisposition

**public ContentDisposition()**

No-arg Constructor.
public ContentDisposition(String disposition, ParameterList list)

Constructor.

Parameters:
- disposition - disposition
- list - ParameterList

Since:
JavaMail 1.2

public ContentDisposition(String s) throws ParseException

Throws
- ParseException:

Since
JavaMail 1.2

Constructor that takes a ContentDisposition string. The String is
parsed into its constituents: disposition and parameters. A ParseException is thrown if the parse fails.

**Parameters:**

s - the ContentDisposition string.

**Throws:**

ParseException - if the parse fails.

**Since:**

JavaMail 1.2

### Method Detail

#### public String getDisposition()

```java
definition
return disposition
since JavaMail 1.2
```

**getDisposition**

public String getDisposition()

Return the disposition value.

**Returns:**

the disposition

**Since:**

JavaMail 1.2

#### public String getParameter(String name)

```java
null
return
since JavaMail 1.2
```

```java
null
return
since JavaMail 1.2
```
getParameter

public String getParameter(String name)

Return the specified parameter value. Returns null if this parameter is absent.

Retruns:
parameter value

Since:
JavaMail 1.2

getParameterList

public ParameterList getParameterList()

ParameterList null

return ParameterList

since JavaMail 1.2

getParameterList

public ParameterList getParameterList()

Return a ParameterList object that holds all the available parameters. Returns null if no parameters are available.

Returns:
ParameterList

Since:
JavaMail 1.2

disposition

public void setDisposition(String disposition)

disposition disposition

since JavaMail 1.2
**setDisposition**

public void **setDisposition** (String disposition)

Set the disposition. Replaces the existing disposition.

**Parameters:**

disposition - the disposition

**Since:**

JavaMail 1.2

---

**public void setParameter(String name, String value)**

name
value
since
JavaMail 1.2

**setParameter**

public void **setParameter** (String name, String value)

Set the specified parameter. If this parameter already exists, it is replaced by this new value.

**Parameters:**

name - parameter name
value - parameter value

**Since:**

JavaMail 1.2

---

**public void setParameterList(ParameterList list)**

ParameterList

list

**ParameterList**
setParameterList

public void setParameterList(ParameterList list)

Set a new ParameterList.

Parameters:
  list - ParameterList

Since:
    JavaMail 1.2

public String toString()

ContentDisposition RFC2045

return RFC2045

since JavaMail 1.2

toString

public String toString()

Retrieve a RFC2045 style string representation of this ContentDisposition. Returns null if the conversion failed.

Overrides:
  toString in class Object

Returns:
  RFC2045 style string

Since:
  JavaMail 1.2
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.internet  Class ContentType

java.lang.Object
   \ javax.mail.internet.ContentType

public class ContentType
extends Object

MIME ContentType  ContentType  MIME  ContentType

version 1.10, 07/05/04

This class represents a MIME ContentType value. It provides methods to parse a ContentType string into individual components and to generate a MIME style ContentType string.

Version: 1.10, 07/05/04
Author: John Mani

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ContentType</strong>()</td>
<td>No-arg Constructor.</td>
</tr>
<tr>
<td><strong>ContentType</strong> <em>(String s)</em></td>
<td>Constructor that takes a Content-Type string.</td>
</tr>
<tr>
<td><strong>ContentType</strong> <em>(String primaryType, String subType, ParameterList list)</em></td>
<td>Constructor.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>String getBaseType()</code></td>
<td>Return the MIME type string, without the parameters.</td>
</tr>
<tr>
<td><code>String getParameter(String name)</code></td>
<td>Return the specified parameter value.</td>
</tr>
<tr>
<td><code>ParameterList getParameterList()</code></td>
<td>Return a ParameterList object that holds all the available parameters.</td>
</tr>
<tr>
<td><code>String getPrimaryType()</code></td>
<td>Return the primary type.</td>
</tr>
<tr>
<td><code>String getSubType()</code></td>
<td>Return the subType.</td>
</tr>
<tr>
<td><code>boolean match(ContentType cType)</code></td>
<td>Match with the specified ContentType object.</td>
</tr>
<tr>
<td><code>boolean match(String s)</code></td>
<td>Match with the specified content-type string.</td>
</tr>
<tr>
<td><code>void setParameter(String name, String value)</code></td>
<td>Set the specified parameter.</td>
</tr>
<tr>
<td><code>void setParameterList(ParameterList list)</code></td>
<td>Set a new ParameterList.</td>
</tr>
<tr>
<td><code>void setPrimaryType(String primaryType)</code></td>
<td>Set the primary type.</td>
</tr>
<tr>
<td><code>void setSubType(String subType)</code></td>
<td>Set the subType.</td>
</tr>
<tr>
<td><code>String toString()</code></td>
<td>Retrieve a RFC2045 style string representation of this Content-Type.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail
public ContentType()

ContentType

public ContentType()
    No-arg Constructor.

public ContentType(String primaryType, String subType, ParameterList list)

    primaryType
    subType
    list

public ContentType(String s) throws ParseException

Content-Type String primaryTypesubType
ParseException
    s  Content-Type
Throws

ParseException:

**ContentType**

```java
public ContentType(String s) throws ParseException

Constructor that takes a Content-Type string. The String is parsed into its constituents: primaryType, subType and parameters. A ParseException is thrown if the parse fails.

**Parameters:**

  s - the Content-Type string.

**Throws:**

  ParseException - if the parse fails.
```

### Method Detail

**public String getPrimaryType()**

```java
return

getPrimaryType
```

**Returns:**

  the primary type

**public String getSubType()**

```java
subType

return
```
**getSubType**

```java
public String getSubType()
```

Return the subType.

**Returns:**
the subType

---

**public String getBaseType()**

```java
MIME primaryType'/' secondaryType
return
```

**getBaseType**

```java
public String getBaseType()
```

Return the MIME type string, without the parameters. The returned value is basically the concatenation of the primaryType, the '/*' character and the secondaryType.

**Returns:**
the type

---

**public String getParameter(String name)**

```java
null
return
```

**getParameter**

```java
public String getParameter(String name)
```
Return the specified parameter value. Returns null if this parameter is absent.

**Returns:**
- parameter value

```java
public ParameterList getParameterList()
ParameterList null
                   return ParameterList
```

getParameterList

```java
public ParameterList getParameterList()
```

Return a ParameterList object that holds all the available parameters. Returns null if no parameters are available.

**Returns:**
- ParameterList

```java
public void setPrimaryType(String primaryType)
```

```java
primaryType
```

setPrimaryType

```java
public void setPrimaryType(String primaryType)
```

Set the primary type. Overrides existing primary type.

**Parameters:**
- primaryType - primary type
public void setSubType(String subType)
subType subType

setSubType

public void setSubType(String subType)

Set the subType. Replaces the existing subType.

Parameters:

subType - the subType

public void setParameter(String name, String value)

setParameter

public void setParameter(String name, String value)

Set the specified parameter. If this parameter already exists, it is replaced by this new value.

Parameters:

name - parameter name
value - parameter value

public void setParameterList(ParameterList list)

setParameterList

public void setParameterList(ParameterList list)
**setParameterList**

```java
public void setParameterList(ParameterList list)

Set a new ParameterList.

**Parameters:**

- list - ParameterList
```

**toString**

```java
public String toString()

Content-Type  RFC2045

return  RFC2045

toString
```

**toString**

```java
public String toString()

Retrieve a RFC2045 style string representation of this Content-Type. Returns null if the conversion failed.

**Overrides:**

toString in class Object

**Returns:**

RFC2045 style string
```

**match**

```java
public boolean match(ContentType cType)

ContentType primaryType  subType

ContentType charset=foobar"  "text/plain"  "text/plain;

true  subtype  

ContentType  "text/plain"  "text/*"  true

ContentType  cType  ContentType
```
public boolean match(ContentType cType)

Match with the specified ContentType object. This method compares only the primaryType and subType. The parameters of both operands are ignored.

For example, this method will return true when comparing the ContentTypes for "text/plain" and "text/plain; charset=foobar". If the subType of either operand is the special character '*', then the subtype is ignored during the match. For example, this method will return true when comparing the ContentTypes for "text/plain" and "text/*"

Parameters:

cType - ContentType to compare this against

---

public boolean match(String s)

Match with the specified content-type string. This method compares only the primaryType and subType. The parameters of both operands are ignored.

contentType primaryType subType
"text/plain" true "text/plain; charset=foobar" subType *
"text/plain" "text/*" true

match

public boolean match(String s)
For example, this method will return true when comparing the ContentType for "text/plain" with "text/plain; charset=foobar". If the subtype of either operand is the special character ",", then the subtype is ignored during the match. For example, this method will return true when comparing the ContentType for "text/plain" with "text/*".
javax.faces.component Interface ContextCallback

public interface ContextCallback

UIComponent facet

A simple callback interface that enables taking action on a specific UIComponent (either facet or child) in the view while preserving any contextual state for that component instance in the view.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void invokeContextCallback(FacesContext context, UIComponent target)</code></td>
</tr>
</tbody>
</table>

This method will be called by an implementation of `UIComponent.invokeOnComponent(javax.faces.context.FacesContext, java.lang.String, javax.faces.component.ContextCallback)` and must be passed the component with the clientId given as an argument to `invokeOnComponent`.

### Method Detail

```
public void invokeContextCallback(FacesContext context, UIComponent target)
```

```java
UIComponent#invokeOnComponent
clientId
invokeOnComponent

context

FacesContext

clientId

UIComponent

clientId
```
invokeContextCallback

void invokeContextCallback(FacesContext context, UIComponent target)

This method will be called by an implementation of
UIComponent.invokeOnComponent(javax.faces.context.FacesContext,
java.lang.String, javax.faces.component.ContextCallback) and
must be passed the component with the clientId given as an
argument to invokeOnComponent. At the point in time when this
method is called, the argument target is guaranteed to be in the
proper state with respect to its ancestors in the View.

Parameters:
context - the FacesContext for this request.
target - the UIComponent that was located by clientId by a call
to UIComponent.invokeOnComponent(javax.faces.context.FacesConte

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.
javax.faces.convert Interface Converter

All Known Implementing Classes:

- BigDecimalConverter
- BigIntegerConverter
- BooleanConverter
- ByteConverter
- CharacterConverter
- DateTimeConverter
- DoubleConverter
- EnumConverter
- FloatConverter
- IntegerConverter
- LongConverter
- NumberConverter
- ShortConverter

---

```java
public interface Converter

Implemented by: BigDecimalConverter, BigIntegerConverter, BooleanConverter, ByteConverter, CharacterConverter, DateTimeConverter, DoubleConverter, EnumConverter, FloatConverter, IntegerConverter, LongConverter, NumberConverter, ShortConverter

Converter String Object-to-String String-to-Object Java

<table>
<thead>
<tr>
<th>Converter</th>
<th>Converter</th>
<th>StateHolder</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.2</td>
<td>Class</td>
<td>Class</td>
</tr>
<tr>
<td>Java</td>
<td>java.util.Locale</td>
<td>FacesContext</td>
</tr>
<tr>
<td>javax.faces.componentUIViewRoot</td>
<td>Locale</td>
<td>Converter</td>
</tr>
</tbody>
</table>

Converter is an interface describing a Java class that can perform Object-to-String and String-to-Object conversions between model data objects and a String representation of those objects that is suitable for rendering.

Converter implementations must have a zero-arguments public constructor. In addition, if the Converter class wishes to have configuration property values saved and restored with the component tree, the implementation must also implement StateHolder.
Starting with version 1.2 of the specification, an exception to the above zero-arguments constructor requirement has been introduced. If a converter has a single argument constructor that takes a class instance and the class of the data to be converted is known at converter instantiation time, this constructor must be used to instantiate the converter instead of the zero-argument version. This enables the per-class conversion of Java enumerated types.

If any Converter implementation requires a java.util.Locale to perform its job, it must obtain that Locale from the UIViewRoot of the current FacesContext, unless the Converter maintains its own Locale as part of its state.

---

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
<td><code>getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><code>getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

---

**Method Detail**

```java
public Object getAsObject(FacesContext context, UIComponent component, String value)
```

UIComponent
getAsObject

**Object** getAsObject(FacesContext context, UIComponent component, String value)

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Parameters:**
- **context** - FacesContext for the request being processed
- **component** - UIComponent with which this model object value is associated
- **value** - String value to be converted (may be null)

**Returns:**
- null if the value to convert is null, otherwise the result of the conversion

**Throws:**
- ConverterException - if conversion cannot be successfully performed
- NullPointerException - if context or component is null

---

public String getAsString(FacesContext context, UIComponent component, Object value)

UIComponent
getAsString

String getAsString(FacesContext context, UIComponent component, Object value)

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Parameters:
- context - FacesContext for the request being processed
- component - UIComponent with which this model object value is associated
- value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
- ConverterException - if conversion cannot be successfully performed
- NullPointerException - if context or component is null
to license terms.

PS:
Overview Package Tree Deprecated Index Help
PREV CLASS  NEXT CLASS  FRAMES NO FRAMES
SUMMARY: NESTED  FIELD  CONSTR  METHOD  DETAIL: FIELD  CONSTR  METHOD
public abstract class ConverterELTag
extends TagSupport

Extends: TagSupport

ConverterELTag is a base class for all JSP custom actions that create and register a Converter instance on the ValueHolder associated with our most immediate surrounding instance of a tag whose implementation class is a subclass of UICOMPONENTCLASSICTAGBASE. To avoid creating duplicate instances when a page is redisplayed, creation and registration of a Converter occurs only if the corresponding UICOMPONENT was created (by the owning UICOMPONENTTAG) during the execution of the current page.

This class may be used as a base class for tag instances that support
specific Converter subclasses.

Subclasses of this class must implement the createConverter() method, which creates and returns a Converter instance. Any configuration properties that specify behavior of this Converter must have been set by the createConverter() method. Generally, this occurs by copying corresponding attribute values on the tag instance.

This tag creates no output to the page currently being created. It is used solely for the side effect of Converter creation.

See Also:
   Serialized Form

---

### Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.servlet.jsp.tagext.TagSupport</th>
</tr>
</thead>
<tbody>
<tr>
<td>id, pageContext</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.servlet.jsp.tagext.IterationTag</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVAL_BODY_AGAIN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.servlet.jsp.tagext.Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE</td>
</tr>
</tbody>
</table>

### Constructor Summary

ConverterELTag()

### Method Summary

<table>
<thead>
<tr>
<th>protected abstract Converter createConverter()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create and return a new Converter to be registered on our surrounding UIComponent.</td>
</tr>
</tbody>
</table>
doStartTag()

Create a new instance of the specified Converter class, and register it with the UICOMPONENT instance associated with our most immediately surrounding UIComponentClassicTagBase instance, if the UICOMPONENT instance was created by this execution of the containing JSP page.

Methods inherited from class javax.servlet.jsp.tagext.TagSupport

doAfterBody, doEndTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, getParent, setValue

Methods inherited from class java.lang.Object

cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ConverterELTag()

ConverterELTag

public ConverterELTag()

Method Detail

public int doStartTag() throws JspException

Converter JSP UICOMPONENT
UICOMPONENTClassicTagBase UICOMPONENT UICOMPONENT
localValue String

Throws JspException: JSP
doStartTag

public int doStartTag() throws JspException

Create a new instance of the specified Converter class, and register it with the UIComponent instance associated with our most immediately surrounding UICOMPONENT instance associated with our most immediately surrounding UICOMPONENT instance, if the UICOMPONENT instance was created by this execution of the containing JSP page. If the localValue of the UIComponent is a String, attempt to convert it.

Specified by:
doStartTag in interface Tag

Overrides:
doStartTag in class TagSupport

Returns:
SKIP_BODY

Throws:
JspException - if a JSP error occurs

See Also:
Tag.doStartTag()

abstract protected Converter createConverter() throws JspException

Converter UICOMPONENT

Throws JspException:

createConverter

protected abstract Converter createConverter() throws JspException
Create and return a new \texttt{Converter} to be registered on our surrounding \texttt{UIComponent}.

\textbf{Throws:}

\textit{JspException} - if a new instance cannot be created

\textbf{Overview} \hspace{1cm} \textbf{Package} \hspace{1cm} \textbf{Tree} \hspace{1cm} \textbf{Deprecated} \hspace{1cm} \textbf{Index} \hspace{1cm} \textbf{Help}

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to \texttt{license terms}.

PS:
javax.faces.convert Class ConverterException

java.lang.Object
   java.lang.Throwable
      java.lang.Exception
         java.lang.RuntimeException
         javax.faces.FacesException
            javax.faces.convert.ConverterException

All Implemented Interfaces:
   Serializable

public class ConverterException
   extends FacesException

   Extends: Throwable > Exception > RuntimeException > FacesException

ConverterException Converter getAsObject() getAsText()

ConverterException is an exception thrown by the getAsObject() or getAsText() method of a Converter, to indicate that the requested conversion cannot be performed.

See Also:
   Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConverterException()</td>
<td>Construct a new exception with no detail message or root cause.</td>
</tr>
<tr>
<td>ConverterException(FacesMessage message)</td>
<td>Construct a new exception with the specified detail message and</td>
</tr>
</tbody>
</table>
no root cause.

**ConverterException** *(FacesMessage message, Throwable cause)*

Construct a new exception with the specified detail message and root cause.

**ConverterException** *(String message)*

Construct a new exception with the specified detail message and no root cause.

**ConverterException** *(String message, Throwable cause)*

Construct a new exception with the specified detail message and root cause.

**ConverterException** *(Throwable cause)*

Construct a new exception with the specified root cause.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getFacesMessage()</td>
<td>Returns the FacesMessage associated with this exception; this will only be available if the converter that threw this exception instance placed it there.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.faces.FacesException

- `getCause`

### Methods inherited from class java.lang.Throwable

- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public ConverterException()

ConverterException

public ConverterException()

    Construct a new exception with no detail message or root cause.

-------------------------------------

public ConverterException(String message)

    message

ConverterException

public ConverterException(String message)

    Construct a new exception with the specified detail message and no root cause.

    Parameters:
    message - The detail message for this exception

-------------------------------------

public ConverterException(Throwables cause)

    (cause == null ? null : cause.toString())
    cause
ConverterException
public ConverterException(Throwable cause)

Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

Parameters:
cause - The root cause for this exception

ConverterException
public ConverterException(String message, Throwable cause)

ConverterException
public ConverterException(FacesMessage message)

message
ConverterException

public ConverterException(FacesMessage message)

    Construct a new exception with the specified detail message and no root cause.

    Parameters:
        message - The detail message for this exception

public ConverterException(FacesMessage message, Throwable cause)

    message
    cause

ConverterException

public ConverterException(FacesMessage message, Throwable cause)

    Construct a new exception with the specified detail message and root cause.

    Parameters:
        message - The detail message for this exception
        cause - The root cause for this exception

Method Detail

public FacesMessage getFacesMessage()
FacesMessage

getFacesMessage

public FacesMessage getFacesMessage()

Returns the FacesMessage associated with this exception; this will only be available if the converter that threw this exception instance placed it there.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.webapp  Class ConverterTag

java.lang.Object  
  \ javax.servlet.jsp.tagext.TagSupport
  \ javax.faces.webapp.ConverterTag

All Implemented Interfaces:  Serializable, IterationTag, JspTag, Tag

Deprecated. This has been partially replaced by ConverterELTag. The remainder of the functionality, namely, the binding facility and the implementation of the createConverter() method, is now an implementation detail.

Extends:  TagSupport

ConverterTag  JSP  UIComponentTag  ValueHolder
  Converter  UIComponentTag  UIComponent  Converter

  converterId  ID  Converter  converterId  ID
  faces-config.xml  ID

  createConverter()  Converter  Converter

  createConverter()

Converter

deprecated  ConverterELTag  #createConverter

public class ConverterTag
extends TagSupport

ConverterTag is a base class for all JSP custom actions that create and register a Converter instance on the ValueHolder associated with our most immediate surrounding instance of a tag whose implementation
class is a subclass of UIComponentTag. To avoid creating duplicate instances when a page is redisplayed, creation and registration of a Converter occurs only if the corresponding UIComponent was created (by the owning UIComponentTag) during the execution of the current page.

This class may be used directly to implement a generic converter registration tag (based on the converter-id specified by the converterId attribute), or as a base class for tag instances that support specific Converter subclasses. This converterId attribute must refer to one of the well known converter-ids, or a custom converter-id as defined in a faces-config.xml file.

Subclasses of this class must implement the createConverter() method, which creates and returns a Converter instance. Any configuration properties that specify behavior of this Converter must have been set by the createConverter() method. Generally, this occurs by copying corresponding attribute values on the tag instance.

This tag creates no output to the page currently being created. It is used solely for the side effect of Converter creation.

See Also:
   Serialized Form

Field Summary

Fields inherited from class javax.servlet.jsp.tagext.TagSupport
id, pageContext

Fields inherited from interface javax.servlet.jsp.tagext.IterationTag
EVAL_BODY_AGAIN

Fields inherited from interface javax.servlet.jsp.tagext.Tag
EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
<th>Deprecated.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConverterTag()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected Converter</td>
<td><strong>createConverter()</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Create and return a new Converter to be registered on our surrounding UIComponent.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td><strong>doStartTag()</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Create a new instance of the specified Converter class, and register it with the UIComponent instance associated with our most immediately surrounding UIComponentTag instance, if the UIComponent instance was created by this execution of the containing JSP page.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td><strong>release()</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Release references to any acquired resources.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td><strong>setBinding(String binding)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Set the expression that will be used to create a ValueExpression that references a backing bean property of the Converter instance to be created.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>void</td>
<td><strong>setConverterId(String converterId)</strong></td>
</tr>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Set the identifier of the Converter instance to be created.</td>
</tr>
</tbody>
</table>

## Methods inherited from class javax.servlet.jsp.tagext.TagSupport

doAfterBody, doEndTag, findAncestorWithClass, getId, getParent, getValue, getValues, removeValue, setId, setPageContext, setParent, setValue

## Methods inherited from class java.lang.Object

clon, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detail

public ConverterTag()

ConverterTag

public ConverterTag()

Deprecated.

Method Detail

public void setConverterId(String converterId)

Converter

canvas

setConverterId

public void setConverterId(String converterId)

Deprecated.

Set the identifier of the Converter instance to be created.

Parameters:

converterId - The identifier of the converter instance to be created.

public void setBinding(String binding) throws JspException
**ValueExpression**       **Converter** (backing) Bean

\[ binding \]

Throws \[ JspException \] \( \text{JSP} \)

---

**setBinding**

```java
public void setBinding(String binding)
    throws JspException
```

**Deprecated.**

Set the expression that will be used to create a `ValueExpression` that references a backing bean property of the `Converter` instance to be created.

**Parameters:**

- binding - The new expression

**Throws:**

- \[ JspException \] - if a JSP error occurs

---

**public int doStartTag() throws JspException**

```
Converter JSP  UICOMPONENT  UICOMPONENT  UICOMPONENT
      UICOMPONENT           UIComponentTag

localValue String
```

Throws \[ JspException \] \( \text{JSP} \)

---

**doStartTag**

```java
public int doStartTag()
    throws JspException
```

**Deprecated.**
Create a new instance of the specified Converter class, and register it with the UIComponent instance associated with our most immediately surrounding UICOMPONENTTag instance, if the UICOMPONENT instance was created by this execution of the containing JSP page. If the localValue of the UICOMPONENT is a String, attempt to convert it.

Specified by:  
    doStartTag in interface Tag  
Overrides:  
    doStartTag in class TagSupport  
Returns:  
    SKIP_BODY  
Throws:  
    JspException - if a JSP error occurs  
See Also:  
    Tag.doStartTag()  

public void release()

release

public void release()  

Deprecated.  

Release references to any acquired resources.

Specified by:  
    release in interface Tag  
Overrides:  
    release in class TagSupport  
See Also:  
    Tag.release()
protected **Converter** `createConverter()` throws **JspException**

**Converter** **UIComponent**

Throws **JspException**:

**createConverter**

protected **Converter** `createConverter()` throws **JspException**

**Deprecated.**

Create and return a new **Converter** to be registered on our surrounding **UIComponent**.

**Throws:**

**JspException** - if a new instance cannot be created

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to **license terms**.

PS:
javax.servlet.http  **Class Cookie**

java.lang.Object  
   \_ javax.servlet.http.Cookie

**All Implemented Interfaces:**  
  Cloneable

```java
public class Cookie
extends Object
implements Cloneable

Implements: Cloneable
```

cookie  servlet  Web  cookie  cookie

cookie  Web  bug  servlet

servlet  HttpServletResponse#addCookie  cookie  HTTP
cookie  Web  20  cookie  300  cookie  cookie
4 KB

HTTP  cookie  servlet  HttpServletResponse#getCookies
cookie  cookie

cookie  Web  HTTP 1.0  cookie  HTTP 1.1

0  Netscape  1  RFC 2109  cookie  cookie
0

Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server. A cookie's value can uniquely identify a client, so cookies are commonly used for session management.
A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number. Some Web browsers have bugs in how they handle the optional attributes, so use them sparingly to improve the interoperability of your servlets.

The servlet sends cookies to the browser by using the `HttpServletResponse.addCookie(javax.servlet.http.Cookie)` method, which adds fields to HTTP response headers to send cookies to the browser, one at a time. The browser is expected to support 20 cookies for each Web server, 300 cookies total, and may limit cookie size to 4 KB each.

The browser returns cookies to the servlet by adding fields to HTTP request headers. Cookies can be retrieved from a request by using the `HttpServletRequest.getCookies()` method. Several cookies might have the same name but different path attributes.

Cookies affect the caching of the Web pages that use them. HTTP 1.0 does not cache pages that use cookies created with this class. This class does not support the cache control defined with HTTP 1.1.

This class supports both the Version 0 (by Netscape) and Version 1 (by RFC 2109) cookie specifications. By default, cookies are created using Version 0 to ensure the best interoperability.

**Author:**
Various

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie(String name, String value)</td>
<td>Constructs a cookie with a specified name and value.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone()</td>
<td>Overrides the standard <code>java.lang.Object.clone</code> method to</td>
</tr>
</tbody>
</table>
return a copy of this cookie.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getComment()</code></td>
<td>Returns the comment describing the purpose of this cookie, or <code>null</code> if the cookie has no comment.</td>
</tr>
<tr>
<td><code>String getDomain()</code></td>
<td>Returns the domain name set for this cookie.</td>
</tr>
<tr>
<td><code>int getMaxAge()</code></td>
<td>Returns the maximum age of the cookie, specified in seconds. By default, <code>-1</code> indicating the cookie will persist until browser shutdown.</td>
</tr>
<tr>
<td><code>String getName()</code></td>
<td>Returns the name of the cookie.</td>
</tr>
<tr>
<td><code>String getPath()</code></td>
<td>Returns the path on the server to which the browser returns this cookie.</td>
</tr>
<tr>
<td><code>boolean getSecure()</code></td>
<td>Returns <code>true</code> if the browser is sending cookies only over a secure protocol, or <code>false</code> if the browser can send cookies using any protocol.</td>
</tr>
<tr>
<td><code>String getValue()</code></td>
<td>Returns the value of the cookie.</td>
</tr>
<tr>
<td><code>int getVersion()</code></td>
<td>Returns the version of the protocol this cookie complies with.</td>
</tr>
<tr>
<td><code>void setComment(String purpose)</code></td>
<td>Specifies a comment that describes a cookie's purpose.</td>
</tr>
<tr>
<td><code>void setDomain(String pattern)</code></td>
<td>Specifies the domain within which this cookie should be presented.</td>
</tr>
<tr>
<td><code>void setMaxAge(int expiry)</code></td>
<td>Sets the maximum age of the cookie in seconds.</td>
</tr>
<tr>
<td><code>void setPath(String uri)</code></td>
<td>Specifies a path for the cookie to which the client should return the cookie.</td>
</tr>
<tr>
<td><code>void setSecure(boolean flag)</code></td>
<td></td>
</tr>
</tbody>
</table>
### void
Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.

### void
setValue(String newValue)
Assigns a new value to a cookie after the cookie is created.

### void
setVersion(int v)
Sets the version of the cookie protocol this cookie complies with.

---

### Methods inherited from class java.lang.Object
equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

### Constructor Detail

**public Cookie(String name, String value)**

**RFC 2109 ASCII $ cookie**

**cookie setValue**

**Netscape cookie cookie**

**setVersion**

<table>
<thead>
<tr>
<th>name</th>
<th>cookie</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>cookie</td>
<td>String</td>
</tr>
</tbody>
</table>

**Throws**

IllegalArgumentException: cookie cookie

**See also**

setValue, setVersion

---

## Cookie

**public Cookie(String name, String value)**
Constructs a cookie with a specified name and value.

The name must conform to RFC 2109. That means it can contain only ASCII alphanumeric characters and cannot contain commas, semicolons, or white space or begin with a $ character. The cookie’s name cannot be changed after creation.

The value can be anything the server chooses to send. Its value is probably of interest only to the server. The cookie’s value can be changed after creation with the `setValue` method.

By default, cookies are created according to the Netscape cookie specification. The version can be changed with the `setVersion` method.

**Parameters:**
- `name` - a `String` specifying the name of the cookie
- `value` - a `String` specifying the value of the cookie

**Throws:**
- `IllegalArgumentException` - if the cookie name contains illegal characters (for example, a comma, space, or semicolon) or it is one of the tokens reserved for use by the cookie protocol

**See Also:**
- `setValue(java.lang.String)`, `setVersion(int)`

### Method Detail

**public void setComment(String purpose)**

```java
cookie  cookieNetscape Version 0 cookie

    purpose  String
See also  getComment
```

### setComment

**public void setComment(String purpose)**
Specifies a comment that describes a cookie's purpose. The comment is useful if the browser presents the cookie to the user. Comments are not supported by Netscape Version 0 cookies.

**Parameters:**
- *purpose* - a String specifying the comment to display to the user

**See Also:**
- `getComment()`

```java
public String getComment()

cookie
cookie

null

return

String
null

setComment
```

**getComment**

```java
public String getComment()

Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.

**Returns:**
- a String containing the comment, or null if none

**See Also:**
- `setComment(java.lang.String)`

```java
public void setDomain(String pattern)

RFC 2109 ( .foo.com) Domain Name System
DNS www.foo.com a.b.foo.com cookie cookie

pattern

cookie

String RFC 2109

See also
getDomain
```
setDomain

public void setDomain(String pattern)

Specifies the domain within which this cookie should be presented.

The form of the domain name is specified by RFC 2109. A domain name begins with a dot (\.foo.com) and means that the cookie is visible to servers in a specified Domain Name System (DNS) zone (for example, www.foo.com, but not a.b.foo.com). By default, cookies are only returned to the server that sent them.

Parameters:

pattern - a String containing the domain name within which this cookie is visible; form is according to RFC 2109

See Also:

getDomain()

---

getDomain

public String getDomain()

return String

See also setDomain

---

getDomain

public String getDomain()

Returns the domain name set for this cookie. The form of the domain name is set by RFC 2109.

Returns:

a String containing the domain name

See Also:

setDomain(java.lang.String)
public void setMaxAge(int expiry)

Sets the maximum age of the cookie in seconds.

A positive value indicates that the cookie will expire after that many seconds have passed. Note that the value is the *maximum* age when the cookie will expire, not the cookie's current age.

A negative value means that the cookie is not stored persistently and will be deleted when the Web browser exits. A zero value causes the cookie to be deleted.

**Parameters:**
expiry - an integer specifying the maximum age of the cookie in seconds; if negative, means the cookie is not stored; if zero, deletes the cookie

**See Also:**
getMaxAge()

---

public int getMaxAge()

return -1

See also setMaxAge
**getMaxAge**

```java
def public int getMaxAge()
```

Returns the maximum age of the cookie, specified in seconds. By default, -1 indicating the cookie will persist until browser shutdown.

**Returns:**
an integer specifying the maximum age of the cookie in seconds; if negative, means the cookie persists until browser shutdown

**See Also:**
`setMaxAge(int)`

---

**public void setPath(String uri)**

```java
def public void setPath(String uri)
```

Specifies a path for the cookie to which the client should return the cookie.

The cookie is visible to all the pages in the directory you specify, and all the pages in that directory's subdirectories. A cookie's path must include the servlet that set the cookie, for example, `/catalog`, which makes the cookie visible to all directories on the server under
Consult RFC 2109 (available on the Internet) for more information on setting path names for cookies.

**Parameters:**
- `uri` - a `String` specifying a path

**See Also:**
- `getPath()`

---

### getPath

```java
public String getPath() {
    return servletString;
}
```

**getPath**

public `String` `getPath()`

Returns the path on the server to which the browser returns this cookie. The cookie is visible to all subpaths on the server.

**Returns:**
- a `String` specifying a path that contains a servlet name, for example, `/catalog`

**See Also:**
- `setPath(java.lang.String)`

---

### setSecure(boolean flag)

**setSecure**

public `void` `setSecure(boolean flag)`

false

**See also**
- `getSecure()`
setSecure

```java
public void setSecure(boolean flag)
```

Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.

The default value is `false`.

**Parameters:**
- `flag` - if `true`, sends the cookie from the browser to the server only when using a secure protocol; if `false`, sent on any protocol

**See Also:**
- `getSecure()`

---

getSecure

```java
public boolean getSecure()
```

 Returns `true` if the browser is sending cookies only over a secure protocol, or `false` if the browser can send cookies using any protocol.

**Returns:**
- `true` if the browser uses a secure protocol; otherwise, `true`

**See Also:**
- `setSecure(boolean)`

---

getName

```java
public String getName()
```
### getName

```java
public String getName()
```

Returns the name of the cookie. The name cannot be changed after creation.

**Returns:**
- a `String` specifying the cookie's name

### setValue

```java
public void setValue(String newValue)
```

Assigns a new value to a cookie after the cookie is created. If you use a binary value, you may want to use BASE64 encoding.

With Version 0 cookies, values should not contain white space, brackets, parentheses, equals signs, commas, double quotes, slashes, question marks, at signs, colons, and semicolons. Empty values may not behave the same way on all browsers.

**Parameters:**
- `newValue` - a `String` specifying the new value
public String getValue()

    cookie

    return

See also

    setValue, java.servlet.http.Cookie

getValue

public String getValue()

    Returns the value of the cookie.

    Returns:
    
    a String containing the cookie's present value

    See Also:
    
    setValue(java.lang.String), Cookie

public int getVersion()

    cookie 1 RFC 2109 0 Netscape cookie cookie cookie

    return  cookie Netscape 0 cookie RFC 2109 1

See also

    setVersion

getVersion

public int getVersion()

    Returns the version of the protocol this cookie complies with. Version 1 complies with RFC 2109, and version 0 complies with the original cookie specification drafted by Netscape. Cookies provided by a browser use and identify the browser's cookie version.
Returns:
0 if the cookie complies with the original Netscape specification;
1 if the cookie complies with RFC 2109

See Also:
setVersion(int)

---

`public void setVersion(int v)`

cookie  cookie  0 Netscape cookie  1
RFC 2109

RFC 2109  1

v  cookie  Netscape  0 cookie  RFC 2109

See also  getVersion

---

**setVersion**

`public void setVersion(int v)`

Sets the version of the cookie protocol this cookie complies with. Version 0 complies with the original Netscape cookie specification. Version 1 complies with RFC 2109.

Since RFC 2109 is still somewhat new, consider version 1 as experimental; do not use it yet on production sites.

**Parameters:**

v - 0 if the cookie should comply with the original Netscape specification; 1 if the cookie should comply with RFC 2109

**See Also:**
getVersion()
public Object clone()

Overrides the standard java.lang.Object.clone method to return a copy of this cookie.

Overrides:

clone in class Object
javax.management.j2ee.statistics  

**Interface CountStatistic**

All Superinterfaces:  
  [Statistic](#)

```java
public interface CountStatistic
extends Statistic

Implements: Statistic
```

Specifies standard count measurements.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>long getCount()</strong></td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Statistic  
getDescription, getLastSampleTime, getName, getStartTime, getUnit

### Method Detail

`public long getCount()`

**getCount**
long `getCount()`

The count since the last reset.
javax.ejb  **Class CreateException**

**java.lang.Object**
- **java.lang.Throwable**
  - **java.lang.Exception**
    - **javax.ejb.CreateException**

**All Implemented Interfaces:**
- **Serializable**

**Direct Known Subclasses:**
- **DuplicateKeyException**

```java
public class CreateException
    extends Exception

Extends: Throwable > Exception
Extended by: DuplicateKeyException

Bean home throws CreateException

EJB
```

The CreateException exception must be included in the throws clauses of all create methods defined in an enterprise Bean's home interface.

This exception is used as a standard application-level exception to report a failure to create an EJB object.

**See Also:**
- **Serialized Form**

---

**Constructor Summary**
CreateException()
Constructs a CreateException with no detail message.

CreateException(String message)
Constructs a CreateException with the specified detail message.

**Method Summary**

**Methods inherited from class java.lang.Throwable**

- fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

**Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Constructor Detail**

public CreateException()
CreateException

CreateException

public CreateException()

    Constructs a CreateException with no detail message.
public CreateException(String message)

Constructs a CreateException with the specified detail message.
javax.activation Interface DataContentHandler

public interface DataContentHandler

DataContentHandler Transferable DataHandler
DataContentHandler

DataContentHandler DataHandler DataHandler
DataContentHandlerFactory MIME DataContentHandler
DataHandler DataContentHandler then

The DataContentHandler interface is implemented by objects that can be used to extend the capabilities of the DataHandler's implementation of the Transferable interface. Through DataContentHandlers the framework can be extended to convert streams in to objects, and to write objects to streams.

Applications don't generally call the methods in DataContentHandlers directly. Instead, an application calls the equivalent methods in DataHandler. The DataHandler will attempt to find an appropriate DataContentHandler that corresponds to its MIME type using the current DataContentHandlerFactory. The DataHandler then calls through to the methods in the DataContentHandler.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>getContentPane(DataSource ds)</strong></td>
</tr>
<tr>
<td>Return an object representing the data in its most preferred form.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>getTransferData(DataFlavor df, DataSource ds)</strong></td>
</tr>
<tr>
<td>Returns an object which represents the data to be transferred.</td>
</tr>
</tbody>
</table>
getTransferDataFlavors()  
Returns an array of DataFlavor objects indicating the flavors the data can be provided in.

writeTo(Object obj, String mimeType, OutputStream os) 
Convert the object to a byte stream of the specified MIME type and write it to the output stream.

Method Detail

public java.awt.datatransfer.DataFlavor[]
getTransferDataFlavors()

DataFlavor flavor

return DataFlavor

getTransferDataFlavors

DataFlavor[] getTransferDataFlavors()

Returns an array of DataFlavor objects indicating the flavors the data can be provided in. The array should be ordered according to preference for providing the data (from most richly descriptive to least descriptive).

Returns:
The DataFlavors.

public Object
getTransferData(java.awt.datatransfer.DataFlavor df, DataSource ds) throws java.awt.datatransfer.UnsupportedFlavorException, java.io.IOException

flavor

df DataFlavor
getTransferData

Object getTransferData(DataFlavor df, DataSource ds)
throws UnsupportedFlavorException,
IOException

Returns an object which represents the data to be transferred. The class of the object returned is defined by the representation class of the flavor.

Parameters:
  df - The DataFlavor representing the requested type.
  ds - The DataSource representing the data to be converted.

Returns:
The constructed Object.

Throws:
  UnsupportedFlavorException - if the handler doesn't support the requested flavor
  IOException - if the data can't be accessed

public Object getContent(DataSource ds) throws java.io.IOException

getContent
Object getContent(DataSource ds) throws IOException

Return an object representing the data in its most preferred form. Generally this will be the form described by the first DataFlavor returned by the getTransferDataFlavors method.

Parameters:
  ds - The DataSource representing the data to be converted.

Returns:
The constructed Object.

Throws:
  IOException - if the data can't be accessed

public void writeTo(Object obj, String mimeType, java.io.OutputStream os) throws java.io.IOException

Convert the object to a byte stream of the specified MIME type and write it to the output stream.

Parameters:
  obj - The object to be converted.
  mimeType - The requested MIME type of the resulting byte stream.
  os - The output stream into which to write the converted byte
stream.

**Throws:**

`IOException` - errors writing to the stream

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

---

PS:
javax.activation Interface DataContentHandlerFactory

public interface DataContentHandlerFactory

DataContentHandler MIME ContentHandler
java.net.URL ContentHandler

This interface defines a factory for DataContentHandlers. An implementation of this interface should map a MIME type into an instance of DataContentHandler. The design pattern for classes implementing this interface is the same as for the ContentHandler mechanism used in java.net.URL.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createDataContentHandler(String mimeType)</td>
<td>Creates a new DataContentHandler object for the MIME type.</td>
</tr>
</tbody>
</table>

Method Detail

public DataContentHandler
createDataContentHandler(String mimeType)

MIME DataContentHandler

mimeType
return

createDataContentHandler
DataContentHandler createDataContentHandler(String mimeType)

Creates a new DataContentHandler object for the MIME type.

Parameters:
   mimeType - the MIME type to create the DataContentHandler for.

Returns:
   The new DataContentHandler, or null if none are found.
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
java.activation Class DataHandler

java.lang.Object
   - javax.activation.DataHandler

All Implemented Interfaces:
   Transferable

public class DataHandler
extends Object
implements Transferable

Implements: java.awt.datatransfer.Transferable

DataHandler DataContentHandler CommandMap

DataHandler Transferable

DataHandler Transferable AWT Transferable
DataContentHandler DataContentHandler DataHandler
MIME

DataHandler CommandMap

DataHandler CommandMap
getCommand getAllCommands getPreferredCommands setCommandMap

URL DataHandler URLDataSource

See also
javax.activation.CommandMap,
javax.activation.DataContentHandler,
javax.activation.DataSource,
javax.activation.URLDataSource
The DataHandler class provides a consistent interface to data available in many different sources and formats. It manages simple stream to string conversions and related operations using DataContentHandlers. It provides access to commands that can operate on the data. The commands are found using a CommandMap.

**DataHandler and the Transferable Interface**

DataHandler implements the Transferable interface so that data can be used in AWT data transfer operations, such as cut and paste and drag and drop. The implementation of the Transferable interface relies on the availability of an installed DataContentHandler object corresponding to the MIME type of the data represented in the specific instance of the DataHandler.

**DataHandler and CommandMaps**

The DataHandler keeps track of the current CommandMap that it uses to service requests for commands (getCommand, getAllCommands, getPreferredCommands). Each instance of a DataHandler may have a CommandMap associated with it using the setCommandMap method. If a CommandMap was not set, DataHandler calls the getDefaultCommandMap method in CommandMap and uses the value it returns. See CommandMap for more information.

**DataHandler and URLs**

The current DataHandler implementation creates a private instance of URLDataSource when it is constructed with a URL.

**See Also:**

CommandMap, DataContentHandler, DataSource, URLDataSource

---

**Constructor Summary**

DataHandler(Source ds)

Create a DataHandler instance referencing the specified
**DataSource.**

**DataHandler**(*Object* obj, *String* mimeType)

Create a DataHandler instance representing an object of this MIME type.

**DataHandler**(*URL* url)

Create a DataHandler instance referencing a URL.

### Method Summary

<table>
<thead>
<tr>
<th>Parameter Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>CommandInfo</em>[ ]</td>
<td><code>getAllCommands()</code></td>
<td>Return all the commands for this type of data.</td>
</tr>
<tr>
<td><em>Object</em></td>
<td><code>getBean(CommandInfo cmdinfo)</code></td>
<td>A convenience method that takes a CommandInfo object and instantiates the corresponding command, usually a JavaBean component.</td>
</tr>
<tr>
<td><em>CommandInfo</em></td>
<td><code>getCommand(String cmdName)</code></td>
<td>Get the command <code>cmdName</code>.</td>
</tr>
<tr>
<td><em>Object</em></td>
<td><code>getContent()</code></td>
<td>Return the data in its preferred Object form.</td>
</tr>
<tr>
<td><em>String</em></td>
<td><code>getContentType()</code></td>
<td>Return the MIME type of this object as retrieved from the source object.</td>
</tr>
<tr>
<td><em>DataSource</em></td>
<td><code>getDataSource()</code></td>
<td>Return the DataSource associated with this instance of DataHandler.</td>
</tr>
<tr>
<td><em>InputStream</em></td>
<td><code>getInputStream()</code></td>
<td>Get the InputStream for this object.</td>
</tr>
<tr>
<td><em>String</em></td>
<td><code>getName()</code></td>
<td>Return the name of the data object.</td>
</tr>
<tr>
<td><em>OutputStream</em></td>
<td><code>getOutputStream()</code></td>
<td>Get an OutputStream for this DataHandler to allow overwriting the underlying data.</td>
</tr>
<tr>
<td><em>CommandInfo</em>[ ]</td>
<td><code>getPreferredCommands()</code></td>
<td>Return the preferred commands for this type of data.</td>
</tr>
<tr>
<td><em>Object</em></td>
<td><code>getTransferData(DataFlavor flavor)</code></td>
<td></td>
</tr>
</tbody>
</table>
Returns an object that represents the data to be transferred.

### Methods

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getTransferDataFlavors()</code></td>
<td>Return the DataFlavors in which this data is available.</td>
</tr>
<tr>
<td><code>isDataFlavorSupported(DataFlavor flavor)</code></td>
<td>Returns whether the specified data flavor is supported.</td>
</tr>
<tr>
<td><code>setCommandMap(CommandMap commandMap)</code></td>
<td>Set the CommandMap for use by this DataHandler.</td>
</tr>
<tr>
<td><code>setDataContentHandlerFactory(DataContentHandlerFactory newFactory)</code></td>
<td>Sets the DataContentHandlerFactory.</td>
</tr>
<tr>
<td><code>writeTo(OutputStream os)</code></td>
<td>Write the data to an OutputStream.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`  

### Constructor Detail

#### public DataHandler(DataSource ds)

**DataSource**

**DataHandler**

**InputStream**

Create a DataHandler instance referencing the specified DataSource. The data exists in a byte stream form. The DataSource will provide an InputStream to access the data.

**Parameters:**

- `ds` - the DataSource
public DataHandler(Object obj, String mimeType)

Create a DataHandler instance representing an object of this MIME type. This constructor is used when the application already has an in-memory representation of the data in the form of a Java Object.

Parameters:
- obj - the Java Object
- mimeType - the MIME type of the object

public DataHandler(java.net.URL url)

Create a DataHandler instance referencing a URL. The DataHandler internally creates a URLDataSource instance to represent the URL.

Parameters:
- url - a URL object
### Method Detail

**public DataSource getDataSource()**

```
DataHandler DataSource

DataSource DataHandler DataHandler
DataSource DataHandler DataHandler
DataSource DataSource DataHandler DataHandler

return DataHandler DataSource
```

**getDataSource**

```
public DataSource getDataSource()

Return the DataSource associated with this instance of DataHandler.

For DataHandlers that have been instantiated with a DataSource, this method returns the DataSource that was used to create the DataHandler object. In other cases the DataHandler constructs a DataSource from the data used to construct the DataHandler. DataSources created for DataHandlers not instantiated with a DataSource are cached for performance reasons.

Returns:

a valid DataSource object for this DataHandler
```

**public String getName()**

```
DataHandler DataSource
null

return
```
**getName**

```java
public String getName()
```

Return the name of the data object. If this DataHandler was created with a DataSource, this method calls through to the DataSource.getName method, otherwise it returns `null`.

**Returns:**
the name of the object

---

**public String getContentType()**

```java
MIME
return
```

**getContentType**

```java
public String getContentType()
```

Return the MIME type of this object as retrieved from the source object. Note that this is the full type with parameters.

**Returns:**
the MIME type

---

**public java.io.InputStream getInputStream() throws java.io.IOException**

```java
InputSteam
```

**DataSource DataHandler DataHandler**

```java
DataSource.getInputStream
```

**Object DataHandler DataHandler Object**

**DataContentHandler DataHandler MIME**
getInputStream

public <pre>InputStream</pre> getInputStream()<br>throws <pre>IOException</pre>

Get the InputStream for this object.

For DataHandlers instantiated with a DataSource, the DataHandler calls the DataSource.getInputStream method and returns the result to the caller.

For DataHandlers instantiated with an Object, the DataHandler first attempts to find a DataContentHandler for the Object. If the DataHandler can not find a DataContentHandler for this MIME type, it throws an UnsupportedDataTypeException. If it is successful, it creates a pipe and a thread. The thread uses the DataContentHandler's writeTo method to write the stream data into one end of the pipe. The other end of the pipe is returned to the caller. Because a thread is created to copy the data, IOExceptions that may occur during the copy can not be propagated back to the caller. The result is an empty stream.

Returns:
the InputStream representing this data

Throws:
  IOException - if an I/O error occurs

See Also:
  DataContentHandler.writeTo(java.lang.Object, java.lang.String, java.io.OutputStream)
public void writeTo(java.io.OutputStream os) throws java.io.IOException

OutputStream

DataHandler DataSource writeTo
InputStream Inputstream OutputStream

DataHandler Object writeTo
DataContentHandler DataContentHandler

DataContentHandler writeTo
OutputStream os

Throws java.io.IOException: I/O *

writeTo

public void writeTo(OutputStream os)

throws IOException

Write the data to an OutputStream.

If the DataHandler was created with a DataSource, writeTo retrieves the InputStream and copies the bytes from the InputStream to the OutputStream passed in.

If the DataHandler was created with an object, writeTo retrieves the DataContentHandler for the object's type. If the DataContentHandler was found, it calls the writeTo method on the DataContentHandler.

Parameters:

- os - the OutputStream to write to

Throws:

- IOException - if an I/O error occurs
public java.io.OutputStream getOutputStream() throws java.io.IOException

DataHandler  OutputStream DataHandler
DataSource  DataSource

null

return  OutputStream

See also  getOutputStream, javax.activation.URLDataSource

getOutputStream

public  OutputStream getOutputStream()

throws IOException

Get an OutputStream for this DataHandler to allow overwriting the underlying data. If the DataHandler was created with a DataSource, the DataSource's getOutputStream method is called. Otherwise, null is returned.

Returns:
the OutputStream

Throws:
IOException

See Also:
DataSource.getOutputStream(), URLDataSource

public java.awt.datatransfer.DataFlavor[]
getTransferDataFlavors()

DataFlavor

flavor  DataFlavor

DataHandler  MIME  DataContentHandler
DataHandler  DataContentHandler

getTransferDataFlavors
getTransferDataFlavors

public DataFlavor[] getTransferDataFlavors()

Return the DataFlavors in which this data is available.

Returns an array of DataFlavor objects indicating the flavors the data can be provided in. The array is usually ordered according to preference for providing the data, from most richly descriptive to least richly descriptive.

The DataHandler attempts to find a DataContentHandler that corresponds to the MIME type of the data. If one is located, the DataHandler calls the DataContentHandler's getTransferDataFlavors method.

If a DataContentHandler can not be located, and if the DataHandler was created with a DataSource (or URL), one DataFlavor is returned that represents this object's MIME type and the java.io.InputStream class. If the DataHandler was created with an object and a MIME type, getTransferDataFlavors returns one DataFlavor that represents this object's MIME type and the object's class.

Specified by:
getTransferDataFlavors in interface Transferable

Returns:
an array of data flavors in which this data can be transferred

See Also:
DataContentHandler.getTransferDataFlavors()
public boolean isDataFlavorSupported(DataFlavor flavor)

Returns whether the specified data flavor is supported for this object.

This method iterates through the DataFlavors returned from getTransferDataFlavors, comparing each with the specified flavor.

Specified by: isDataFlavorSupported in interface Transferable

Parameters:
flavor - the requested flavor for the data

Returns:
true if the data flavor is supported

See Also:
getTransferDataFlavors()
flavor

DataSource URL DataHandler

DataHandler MIME DataContentHandler
DataFlavor getTransferData DataHandler
DataContentHandler flavor MIME
java.io.InputStream InputStream
UnsupportedFlavorException

Object DataHandler

DataHandler MIME DataContentHandler
DataFlavor getTransferData DataHandler
DataContentHandler flavor MIME
DataHandler UnsupportedFlavorException

flavor

return

Throws java.awt.datatransfer.UnsupportedFlavorException: flavor
Throws java.io.IOException: I/O *
See also javax.activation.ActivationDataFlavor

getTransferData

public Object getTransferData(DataFlavor flavor) throws UnsupportedFlavorException, IOException

Returns an object that represents the data to be transferred. The class of the object returned is defined by the representation class of the data flavor.

For DataHandler's created with DataSources or URLs:

The DataHandler attempts to locate a DataContentHandler for this
MIME type. If one is found, the passed in DataFlavor and the type of the data are passed to its getTransferData method. If the DataHandler fails to locate a DataContentHandler and the flavor specifies this object's MIME type and the java.io.InputStream class, this object's InputStream is returned. Otherwise it throws an UnsupportedFlavorException.

For DataHandler's created with Objects:

The DataHandler attempts to locate a DataContentHandler for this MIME type. If one is found, the passed in DataFlavor and the type of the data are passed to its getTransferData method. If the DataHandler fails to locate a DataContentHandler and the flavor specifies this object's MIME type and its class, this DataHandler's referenced object is returned. Otherwise it throws an UnsupportedFlavorException.

Specified by:
   getTransferData in interface Transferable
Parameters:
   flavor - the requested flavor for the data
Returns:
   the object
Throws:
   UnsupportedFlavorException - if the data could not be converted to the requested flavor
   IOException - if an I/O error occurs
See Also:
   ActivationDataFlavor

```
public void setCommandMap(CommandMap commandMap)
   DataHandler CommandMap null
CommandMap CommandMap.getOrDefaultCommandMap
CommandMap CommandMap
```

**setCommandMap**

```java
public void setCommandMap(CommandMap commandMap)
```

Set the CommandMap for use by this DataHandler. Setting it to `null` causes the CommandMap to revert to the CommandMap returned by the `CommandMap.getDefaultCommandMap` method. Changing the CommandMap, or setting it to `null`, clears out any data cached from the previous CommandMap.

**Parameters:**
- `commandMap` - the CommandMap to use in this DataHandler

**See Also:**
- `CommandMap.setDefaultCommandMap(javax.activation.CommandMap)`

---

**getPreferredCommands**

```java
public CommandInfo[] getPreferredCommands()
```

```
CommandMap DataHandler
getPreferredCommands DataHandler MIME
CommandMap
```

Return the preferred commands for this type of data. This method calls the `getPreferredCommands` method in the CommandMap associated with this instance of DataHandler. This method returns an array that represents a subset of available commands. In cases where multiple commands for the MIME type represented by this DataHandler are present, the installed CommandMap chooses the appropriate commands.

**See also**
- `getPreferredCommands`
public CommandInfo[] getAllCommands()

Return all the commands for this type of data. This method returns an array containing all commands for the type of data represented by this DataHandler. The MIME type for the underlying data represented by this DataHandler is used to call through to the getAllCommands method of the CommandMap associated with this DataHandler.

Returns:
the CommandInfo objects representing all the commands
See Also:
CommandMap.getAllCommands(java.lang.String)

public CommandInfo getCommand(String cmdName)

cmdName DataHandler CommandMap
DataHandler MIME DataHandler CommandMap
getCommand

return CommandInfo
See also
getCommand
getCommand

public CommandInfo getCommand(String cmdName)

Get the command cmdName. Use the search semantics as defined by the CommandMap installed in this DataHandler. The MIME type for the underlying data represented by this DataHandler is used to call through to the getCommand method of the CommandMap associated with this DataHandler.

Parameters:
- cmdName - the command name

Returns:
- the CommandInfo corresponding to the command

See Also:
- CommandMap.getCommand(java.lang.String, java.lang.String)

---------------------------------------------------------------------

public Object getContent() throws java.io.IOException

Object

DataHandler

DataHandler DataSource DataContentHandler

DataSource

Inputstream

return

Throws java.io.IOException: IOException

g getContent

public Object getContent() throws IOException

Object
Return the data in its preferred Object form.

If the DataHandler was instantiated with an object, return the object.

If the DataHandler was instantiated with a DataSource, this method uses a DataContentHandler to return the content object for the data represented by this DataHandler. If no DataContentHandler can be found for the the type of this data, the DataHandler returns an InputStream for the data.

**Returns:**
the content.

**Throws:**
IOException - if an IOException occurs during this operation.

```java
public Object getBean(CommandInfo cmdinfo)
CommandInfo  JavaBean

CommandInfo getCommandObject
javax.activation.DataHandler ClassLoader
    cmdinfo CommandInfo
    return
```

getBean

```java
public Object getBean(CommandInfo cmdinfo)

    A convenience method that takes a CommandInfo object and instantiates the corresponding command, usually a JavaBean component.

    This method calls the CommandInfo's getCommandObject method with the ClassLoader used to load the javax.activation.DataHandler class itself.
```
**Parameters:**
cmdinfo - the CommandInfo corresponding to a command

**Returns:**
the instantiated command object

---

**public static void setDataContentHandlerFactory(DataContentHandlerFactory newFactory)**

Sets the DataContentHandlerFactory. The DataContentHandlerFactory is called first to find DataContentHandlers. The DataContentHandlerFactory can only be set once.

If the DataContentHandlerFactory has already been set, this method throws an Error.

**Parameters:**
newFactory - the DataContentHandlerFactory

**Throws:**
Error - if the factory has already been defined.

**See Also:**
DataContentHandlerFactory
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
Class **DataModel**

`javax.faces.model.DataModel` extends `java.lang.Object`

**Direct Known Subclasses:**
- ArrayDataModel, ListDataModel, ResultDataModel, ResultSetDataModel, ScalarDataModel

**Extended by:** ArrayDataModel, ListDataModel, ResultDataModel, ResultSetDataModel, ScalarDataModel

```java
public abstract class DataModel extends Object
```

**DataModel** is an abstraction around arbitrary data binding technologies that can be used to adapt a variety of data sources for use by JavaServer Faces components that support per-row processing for their child components (such as **UIData**).

The data collection underlying a **DataModel** instance is modeled as a collection of row objects that can be accessed by a zero-relative cursor (row index). The APIs provide mechanisms to position to a specified location.
zero-relative row index, and to retrieve an object that represents the data that corresponds to the current row index.

A concrete `DataModel` instance is attached to a particular collection of underlying data by calling the `setWrappedData()` method. It can be detached from that underlying data collection by passing a `null` parameter to this method.

Concrete `DataModel` implementations must provide a public zero-arguments constructor that calls `setWrappedData(null)`. A convenience constructor that takes a wrapped object of the appropriate type (and passes it on via a call to `setWrappedData()`), should also be provided.

Event listeners may be registered to receive notifications of when a new row index is selected.

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DataModel()</code></td>
<td>Constructor that must be public and have no arguments and call <code>setWrappedData(null)</code></td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addDataModelListener(DataModelListener listener)</code></td>
<td>Add a new <code>DataModelListener</code> to the set interested in notifications from this <code>DataModel</code></td>
</tr>
<tr>
<td><code>getDataModelListeners()</code></td>
<td>Return the set of <code>DataModelListener</code>s interested in notifications from this <code>DataModel</code></td>
</tr>
<tr>
<td><code>getRowCount()</code></td>
<td>Return the number of rows of data objects represented by this <code>DataModel</code></td>
</tr>
<tr>
<td><code>getRowData()</code></td>
<td>Return an object representing the data for the currently selected row index</td>
</tr>
<tr>
<td><code>getRowIndex()</code></td>
<td>Return the zero-relative index of the currently selected row index</td>
</tr>
</tbody>
</table>
abstract Object getWrappedData()  
Return the object representing the data wrapped by this DataModel, if any.

abstract boolean isRowAvailable()  
Return a flag indicating whether there is rowData available at the current rowIndex.

void removeDataModellListener(DataModellListener listener)  
Remove an existing DataModellListener from the set interested in notifications from this DataModel.

abstract void setRowIndex(int rowIndex)  
Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row.

abstract void setWrappedData(Object data)  
Set the object representing the data collection wrapped by this DataModel.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public DataModel()

DataModel

public DataModel()

Method Detail

abstract public boolean isRowAvailable()
Throws

### isRowAvailable

```java
public abstract boolean isRowAvailable()
```

Return a flag indicating whether there is rowData available at the current rowIndex. If no wrappedData is available, return false.

**Throws:**

- `FacesException` - if an error occurs getting the row availability

---

### abstract public int getRowCount()

```java
public abstract int getRowCount()
```

Return the number of rows of data objects represented by this DataModel. If the number of rows is unknown, or no wrappedData is available, return -1.

**Throws:**

- `FacesException` - if an error occurs getting the row count

---

### abstract public Object getRowData()

```java
public abstract Object getRowData()
```

Return the row data for the current rowIndex.

**Throws:**

- `FacesException` - if an error occurs getting the row data

---
Throws:  
FacesException:  
Throws:  
IllegalArgument Exception:  

getRowData

public abstract Object getRowData()

Return an object representing the data for the currently selected row index. If no wrappedData is available, return null.

Throws:
   FacesException - if an error occurs getting the row data
   IllegalArgumentException - if now row data is available at the currently specified row index

abstract public int getRowIndex()

return 0

wrappedData = -1

Throws:  
FacesException:

getRowIndex

public abstract int getRowIndex()

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no wrappedData is available, return -1.

Throws:
   FacesException - if an error occurs getting the row index

abstract public void setRowIndex(int rowIndex)
setRowIndex

public abstract void setRowIndex(int rowIndex)

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the isRowAvailable() method to detect whether row data will be available for use by the getRowData() method.

If there is no wrappedData available when this method is called, the specified rowIndex is stored (and may be retrieved by a subsequent call to getRowData()), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a DataModelEvent will be sent to the rowSelected() method of all registered DataModelListenerS.

Parameters:
rowIndex - The new zero-relative index (must be non-negative)

Throws:
FacesException - if an error occurs setting the row index
IllegalArgumentException - if rowIndex is less than -1

abstract public Object getWrappedData()
**DataModel**

**getWrappedData**

```java
public abstract Object getWrappedData()
```

Return the object representing the data wrapped by this `DataModel`, if any.

**abstract public void setWrappedData(Object data)**

```java
public abstract void setWrappedData(Object data)
```

Set the object representing the data collection wrapped by this `DataModel`. If the specified `data` is null, detach this `DataModel` from any previously wrapped data collection instead.

If `data` is non-null, the currently selected row index must be set to zero, and a `DataModelEvent` must be sent to the `rowSelected()` method of all registered `DataModelListener`s indicating that this row is now selected.

**Parameters:**
public void addDataModelListener(DataModelListener listener)

Add a new DataModelListener to the set interested in notifications from this DataModel.

Parameters:
listener - The new DataModelListener to be registered

Throws:
NullPointerException - if listener is null

public DataModelListener[] getDataModelListeners()
Return the set of DataModelListener interested in notifications from this DataModel. If there are no such listeners, an empty array is returned.

public void removeDataModelListener(DataModelListener listener)

removeDataModelListener

public void removeDataModelListener(DataModelListener listener)

Remove an existing DataModelListener from the set interested in notifications from this DataModel.

Parameters:

listener - The old DataModelListener to be deregistered

Throws:

NullPointerException - if listener is null
javax.faces.model Class DataModelEvent

java.lang.Object  
   java.util.EventObject  
   javax.faces.model.DataModelEvent

All Implemented Interfaces:
   Serializable

public class DataModelEvent
  extends EventObject

Extends: java.util.EventObject

DataModelEvent represents an event of interest to registered listeners that occurred on the specified DataModel.

See Also:
   Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
source

Constructor Summary

DataModelEvent(DataModel model, int index, Object data)
Construct an event object that is associated with the specified
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDataModel()</code></td>
<td>Return the <code>DataModel</code> that fired this event.</td>
</tr>
<tr>
<td><code>getRowData()</code></td>
<td>Return the object representing the data for the specified row index, or null for no associated row data.</td>
</tr>
<tr>
<td><code>getRowIndex()</code></td>
<td>Return the row index for this event, or -1 for no specific row.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.util.EventObject`
- `getSource`, `toString`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

```
public DataModelEvent(DataModel model, int index, Object data)
```

<table>
<thead>
<tr>
<th>model</th>
<th>DataModel</th>
</tr>
</thead>
<tbody>
<tr>
<td>index</td>
<td>0 -1</td>
</tr>
<tr>
<td>data</td>
<td>index null</td>
</tr>
</tbody>
</table>

DataModelEvent
public DataModelEvent(DataModel model, int index, Object data)

Construct an event object that is associated with the specified row index and associated data.

Parameters:
model - The DataModel on which this event occurred
index - The zero relative row index for which this event occurred, or -1 for no specific row association
data - Representation of the data for the row specified by index, or null for no particular row data

Method Detail

public DataModel getDataModel()

DataModel

dataGetModel

gDataModel

public Object getRowData()

null

getRowData

getRowData
Return the object representing the data for the specified row index, or null for no associated row data.

```java
public int getRowIndex()
-1
```

getRowIndex

```java
public int getRowIndex()
```

Return the row index for this event, or -1 for no specific row.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.model Interface `DataModelListener`

All Superinterfaces:
   `EventListener`

---

```java
public interface DataModelListener
extends EventListener
Implements: java.util.EventListener
```

`DataModelListener` represents an event listener that wishes to be notified of `DataModelEvent`s occurring on a particular `DataModel` instance.

---

**Method Summary**

```java
void rowSelected(DataModelEvent event)
```

Notification that a particular row index, with the associated row data, has been selected for processing.

**Method Detail**

```java
public void rowSelected(DataModelEvent event)
```

`event`  `DataModelEvent`
void rowSelected(DataModelEvent event)

Notification that a particular row index, with the associated row data, has been selected for processing.

Parameters:
  event - The DataModelEvent we are processing
All Known Subinterfaces:
   MultipartDataSource

All Known Implementing Classes:
   ByteArrayOutputStream, FileDataSource, MimePartDataSource, URLDataSource

The `DataSource` interface provides the JavaBeans Activation Framework with an abstraction of an arbitrary collection of data. It provides a type for that data as well as access to it in the form of InputStreams and OutputStreams where appropriate.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getContentType()</code></td>
<td>This method returns the MIME type of the data in the form of a string.</td>
</tr>
<tr>
<td><code>InputStream getInputStream()</code></td>
<td>This method returns an InputStream representing the data and throws the appropriate exception if it cannot do so.</td>
</tr>
<tr>
<td><code>String getName()</code></td>
<td>Return the name of this object where the name of the object is dependant on the nature of the underlying objects.</td>
</tr>
<tr>
<td><code>getOutputStream()</code></td>
<td></td>
</tr>
</tbody>
</table>
This method returns an OutputStream where the data can be written and throws the appropriate exception if it cannot do so.

**Method Detail**

```java
public java.io.InputStream getInputStream() throws java.io.IOException

InputStream return

getInputStream

InputStream InputStream

getInputStream

throws IOException

This method returns an InputStream representing the data and throws the appropriate exception if it cannot do so. Note that a new InputStream object must be returned each time this method is called, and the stream must be positioned at the beginning of the data.

Returns:
an InputStream

Throws:
IOException
```

```java
public java.io.OutputStream getOutputStream() throws java.io.IOException

OutputStream return

getOutputStream

OutputStream OutputStream

getOutputStream

throws IOException

This method returns an OutputStream where the data can be written and throws the appropriate exception if it cannot do so.
```
**OutputStream** getOutputStream()

This method returns an `OutputStream` where the data can be written and throws the appropriate exception if it can not do so. Note that a new `OutputStream` object must be returned each time this method is called, and the stream must be positioned at the location the data is to be written.

**Returns:**
- an `OutputStream`

**Throws:**
- `IOException`

---

```java
public String getContentType()
```

### MIME DataSource getContentType

"application/octet-stream"

```java
return MIME
```

**getContentType**

### String getContentType()

This method returns the MIME type of the data in the form of a string. It should always return a valid type. It is suggested that `getContentType` return "application/octet-stream" if the `DataSource` implementation can not determine the data type.

**Returns:**
- the MIME Type

---

```java
public String getName()
```

### DataSource getName()

```java
return
```

---
getName

String getName()

Return the name of this object where the name of the object is dependant on the nature of the underlying objects. DataSources encapsulating files may choose to return the filename of the object. (Typically this would be the last component of the filename, not an entire pathname.)

Returns:
the name of the object.
javax.xml.bind  **Class DatatypeConverter**

java.lang.Object
  ▼javax.xml.bind.DatatypeConverter

public final class DatatypeConverter
extends Object

**javaType**  XML  Java  XML  parse  print
parse  print  **XML Schema Part2: Datatypes**

parse  print  parse  print  JAXB  parse  print
parse  print  parse  print  xsd:dateTime  long

// Customized parse method
public long myParseCal( String dateTimeString ) {
    java.util.Calendar cal = DatatypeConverter.parseDateTime(dateTimeString);
    long longval = convert_calendar_to_long(cal);  //application specific
    return longval;
}

// Customized print method
public String myPrintCal( Long longval ) {
    java.util.Calendar cal = convert_long_to_calendar(longval);  //app
    String dateTimeString = DatatypeConverter.printDateTime(cal);
    return dateTimeString;
}

DatatypeConverterInterface  parse  print  parse  print

**javaType**  parse  print

JAXB  JAXBContext.newInstance
API

XML  print  XML  IllegalArgumentException
IllegalArgumentException
The javaType binding declaration can be used to customize the binding of an XML schema datatype to a Java datatype. Customizations can involve writing a parse and print method for parsing and printing lexical representations of a XML schema datatype respectively. However, writing parse and print methods requires knowledge of the lexical representations (XML Schema Part2: Datatypes specification) and hence may be difficult to write.

This class makes it easier to write parse and print methods. It defines static parse and print methods that provide access to a JAXB provider's implementation of parse and print methods. These methods are invoked by custom parse and print methods. For example, the binding of xsd:dateTime to a long can be customized using parse and print methods as follows:

```java
// Customized parse method
public long myParseCal( String dateTimeString ) {
    java.util.Calendar cal = DatatypeConverter.parseDateTime(dateTimeString);
    long longval = convert_calendar_to_long(cal); // application specific
    return longval;
}

// Customized print method
public String myPrintCal( Long longval ) {
    java.util.Calendar cal = convert_long_to_calendar(longval);
    String dateTimeString = DatatypeConverter.printDateTime(cal);
    return dateTimeString;
}
```

There is a static parse and print method corresponding to each parse and print method respectively in the DatatypeConverterInterface.

The static methods defined in the class can also be used to specify a
parse or a print method in a javaType binding declaration.

JAXB Providers are required to call the `setDatatypeConverter` api at some point before the first marshal or unmarshal operation (perhaps during the call to JAXBContext.newInstance). This step is necessary to configure the converter that should be used to perform the print and parse functionality.

A print method for a XML schema datatype can output any lexical representation that is valid with respect to the XML schema datatype. If an error is encountered during conversion, then an `IllegalArgumentException`, or a subclass of `IllegalArgumentException` must be thrown by the method.

**Since:**
JAXB1.0

**Version:**
$Revision: 1.3 $

**Author:**
- Sekhar Vajjhala, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Ryan Shoemaker, Sun Microsystems Inc.

**See Also:**
`DatatypeConverterInterface`, `ParseConversionEvent`, `PrintConversionEvent`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static String parseAnySimpleType(String lexicalXSDAnySimpleType)</code></td>
<td>Return a string containing the lexical representation of the simple type.</td>
</tr>
<tr>
<td><code>static byte[] parseBase64Binary(String lexicalXSDBase64Binary)</code></td>
<td>Converts the string argument into an array of bytes.</td>
</tr>
<tr>
<td><code>static boolean parseBoolean(String lexicalXSDBase64Binary)</code></td>
<td>Converts the string argument into a boolean value.</td>
</tr>
<tr>
<td><code>static byte parseByte(String lexicalXSDByte)</code></td>
<td></td>
</tr>
<tr>
<td>Static Method</td>
<td>Class</td>
</tr>
<tr>
<td>--------------</td>
<td>-------</td>
</tr>
<tr>
<td><code>parseDate</code></td>
<td><code>Calendar</code></td>
</tr>
<tr>
<td><code>parseDateTime</code></td>
<td><code>Calendar</code></td>
</tr>
<tr>
<td><code>parseDecimal</code></td>
<td><code>BigDecimal</code></td>
</tr>
<tr>
<td><code>parseDouble</code></td>
<td><code>double</code></td>
</tr>
<tr>
<td><code>parseFloat</code></td>
<td><code>float</code></td>
</tr>
<tr>
<td><code>parseHexBinary</code></td>
<td><code>byte[]</code></td>
</tr>
<tr>
<td><code>parseInt</code></td>
<td><code>int</code></td>
</tr>
<tr>
<td><code>parseInteger</code></td>
<td><code>BigInteger</code></td>
</tr>
<tr>
<td><code>parseLong</code></td>
<td><code>long</code></td>
</tr>
<tr>
<td><code>parseQName</code></td>
<td><code>QName</code></td>
</tr>
<tr>
<td><code>parseShort</code></td>
<td><code>short</code></td>
</tr>
<tr>
<td><code>parseString</code></td>
<td><code>String</code></td>
</tr>
<tr>
<td><code>parseTime</code></td>
<td><code>Calendar</code></td>
</tr>
<tr>
<td><code>parseUnsignedInt</code></td>
<td><code>long</code></td>
</tr>
<tr>
<td><code>parseUnsignedShort</code></td>
<td><code>int</code></td>
</tr>
<tr>
<td><code>printAnySimpleType</code></td>
<td><code>String</code></td>
</tr>
</tbody>
</table>
static String printBase64Binary(byte[] val)
    Converts an array of bytes into a string.
static String printBoolean(boolean val)
    Converts a boolean value into a string.
static String printByte(byte val)
    Converts a byte value into a string.
static String printDate(Calendar val)
    Converts a Calendar value into a string.
static String printDateTime(Calendar val)
    Converts a Calendar value into a string.
static String printDecimal(BigDecimal val)
    Converts a BigDecimal value into a string.
static String printDouble(double val)
    Converts a double value into a string.
static String printFloat(float val)
    Converts a float value into a string.
static String printHexBinary(byte[] val)
    Converts an array of bytes into a string.
static String printInt(int val)
    Converts an int value into a string.
static String printInteger(BigInteger val)
    Converts a BigInteger value into a string.
static String printLong(long val)
    Converts a long value into a string.
static String printQName(QName val, NamespaceContext nsc)
    Converts a QName instance into a string.
static String printShort(short val)
    Converts a short value into a string.
static String printString(String val)
    Converts the string argument into a string.
static String printTime(Calendar val)
    Converts a Calendar value into a string.
static String printUnsignedInt(long val)
    Converts a long value into a string.
**Method Detail**

**public static void setDatatypeConverter(DatatypeConverterInterface converter)**

This method is for JAXB provider use only.

JAXB Providers are required to call this method at some point before allowing any of the JAXB client marshal or unmarshal operations to occur. This is necessary to configure the datatype converter that should be used to perform the print and parse conversions.
Calling this api repeatedly will have no effect - the 
DatatypeConverterInterface instance passed into the first invocation 
is the one that will be used from then on.

Parameters:
  converter - an instance of a class that implements the 
  DatatypeConverterInterface class - this parameter must not be 
  null.

Throws:
  IllegalArgumentException - if the parameter is null

public static String parseString(String lexicalXSDString)

XSD String
  lexicalXSDString
  return

parseString

public static String parseString(String lexicalXSDString)

  Convert the lexical XSD string argument into a String value.

Parameters:
  lexicalXSDString - A string containing a lexical representation of 
  xsd:string.

Returns:
  A String value represented by the string argument.

public static java.math.BigInteger parseInteger(String 
lexicalXSDInteger)

BigInteger
  lexicalXSDInteger
  xsd:integer
parseInteger

public static BigInteger parseInteger(String lexicalXSDInteger)

Convert the string argument into a BigInteger value.

Parameters:
  lexicalXSDInteger - A string containing a lexical representation of xsd:integer.

Returns: A BigInteger value represented by the string argument.

Throws:NumberFormatException - lexicalXSDInteger is not a valid string representation of a BigInteger value.

parseInt

public static int parseInt(String lexicalXSDInt)

int lexicalXSDInt xsd:int
return int

throws NumberFormatException: lexicalXSDInt int

parseInt

public static int parseInt(String lexicalXSDInt)

Convert the string argument into an int value.

Parameters:
lexicalXSDInt - A string containing a lexical representation of xsd:int.

Returns:
A int value represented by the string argument.

Throws:
NumberFormatException - lexicalXSDInt is not a valid string representation of an int value.

public static long parseLong(String lexicalXSDLong)

long

lexicalXSDLong - A string containing lexical representation of xsd:long.

return long

Throws

NumberFormatException: lexicalXSDLong long

parseLong

public static long parseLong(String lexicalXSDLong)

Converts the string argument into a long value.

Parameters:

lexicalXSDLong - A string containing lexical representation of xsd:long.

Returns:
A long value represented by the string argument.

Throws:

NumberFormatException - lexicalXSDLong is not a valid string representation of a long value.

public static short parseShort(String lexicalXSDShort)

short
parseShort

public static short parseShort(String lexicalXSDShort)

Converts the string argument into a short value.

Parameters:
  lexicalXSDShort - A string containing lexical representation of xsd:short.

Returns:
  A short value represented by the string argument.

Throws:
  NumberFormatException - lexicalXSDShort is not a valid string representation of a short value.

public static java.math.BigDecimal parseDecimal(String lexicalXSDDecimal)

BigDecimal

lexicalXSDDecimal  xsd:decimal
return  BigDecimal
Throws  NumberFormatException: lexicalXSDDecimal

parseDecimal

public static BigDecimal parseDecimal(String lexicalXSDDecimal)
Converts the string argument into a BigDecimal value.

**Parameters:**

lexicalXSDDecimal - A string containing lexical representation of xsd:decimal.

**Returns:**

A BigDecimal value represented by the string argument.

**Throws:**

NumberFormatException - lexicalXSDDecimal is not a valid string representation of BigDecimal.

---

```java
public static float parseFloat(String lexicalXSDFloat)
```

**parseFloat**

Converts the string argument into a float value.

**Parameters:**

lexicalXSDFloat - A string containing lexical representation of xsd:float.

**Returns:**

A float value represented by the string argument.

**Throws:**

NumberFormatException - lexicalXSDFloat is not a valid string representation of a float value.
public static double parseDouble(String lexicalXSDDouble)

double

    lexicalXSDDouble    xsd:double
    return double

    Thows
    NumberFormatException: lexicalXSDDouble

parseDouble

public static double parseDouble(String lexicalXSDDouble)

    Converts the string argument into a double value.

    Parameters:
    lexicalXSDDouble - A string containing lexical representation of xsd:double.

    Returns:
    A double value represented by the string argument.

    Thows:
    NumberFormatException - lexicalXSDDouble is not a valid string representation of a double value.

public static boolean parseBoolean(String lexicalXSDBoolean)

boolean

    lexicalXSDBoolean    xsd:boolean
    return boolean

    Thows
    IllegalArgumentException: XML Schema Part 2: Datatypes xsd:boolean
parseBoolean

public static boolean parseBoolean(String lexicalXSDBoolean)

Converts the string argument into a boolean value.

Parameters:
lexicalXSDBoolean - A string containing lexical representation of xsd:boolean.

Returns:
A boolean value represented by the string argument.

Throws:
IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:boolean.

public static byte parseByte(String lexicalXSDByte)

byte

lexicalXSDByte xsd:byte
return byte

Throws
IllegalArgumentException: XML Schema Part 2: Datatypes xsd:byte

parseByte

public static byte parseByte(String lexicalXSDByte)

Converts the string argument into a byte value.

Parameters:
lexicalXSDByte - A string containing lexical representation of xsd:byte.

Returns:
A byte value represented by the string argument.

Throws:
IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:byte.

```java
public static javax.xml.namespace.QName parseQName(String lexicalXSDQName, javax.xml.namespace.NamespaceContext nsc)

byte

<table>
<thead>
<tr>
<th>lexicalXSDQName</th>
<th>XML Schema Part 2: Datatypes</th>
</tr>
</thead>
<tbody>
<tr>
<td>lexicalXSDQName</td>
<td>xsd:QName</td>
</tr>
<tr>
<td>nsc</td>
<td>QName</td>
</tr>
<tr>
<td>return</td>
<td>QName</td>
</tr>
</tbody>
</table>

Throws

IllegalArgumentException: XML Schema Part 2

parseQName

public static QName parseQName(String lexicalXSDQName, NamespaceContext nsc)

Converts the string argument into a byte value.

String parameter lexicalXSDQName must conform to lexical value space specified at XML Schema Part 2: Datatypes specification:QNames

Parameters:

- lexicalXSDQName - A string containing lexical representation of xsd:QName.
- nsc - A namespace context for interpreting a prefix within a QName.
Returns:
   A QName value represented by the string argument.

Throws:
   IllegalArgumentException - if string parameter does not conform
to XML Schema Part 2 specification or if namespace prefix of
lexicalXSDQname is not bound to a URI in NamespaceContext
nsc.

---

**public static java.util.Calendar parseDateTime(String lexicalXSDDateTime)**

Calendar

*lexicalXSDDateTime* xsd:datetime

*return* Calendar

*Throws*

IllegalArgumentException: XML Schema Part 2: Datatypes xsd:dateTime

---

**parseDateTime**

**public static Calendar parseDateTime(String lexicalXSDDateTime)**

Converts the string argument into a Calendar value.

**Parameters:**

*lexicalXSDDateTime* - A string containing lexical representation
of xsd:datetime.

**Returns:**

A Calendar object represented by the string argument.

**Throws:**

IllegalArgumentException - if string parameter does not conform
to lexical value space defined in XML Schema Part 2: Datatypes
for xsd:dateTime.

---

**public static byte[] parseBase64Binary(String**
parseBase64Binary

public static byte[] parseBase64Binary(String lexicalXSDBase64Binary)

Converts the string argument into an array of bytes.

Parameters:

- lexicalXSDBase64Binary - A string containing lexical representation of xsd:base64Binary.

Returns:

- An array of bytes represented by the string argument.

Throws:

- IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:base64Binary

parseHexBinary

public static byte[] parseHexBinary(String lexicalXSDHexBinary)

Converts the string argument into an array of bytes.

Parameters:

- lexicalXSDHexBinary - A string containing lexical representation of xsd:hexBinary.

Returns:

- An array of bytes represented by the string argument.

Throws:

- IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:hexBinary
parseHexBinary

public static byte[] parseHexBinary(String lexicalXSDHexBinary)

Converts the string argument into an array of bytes.

Parameters:
   lexicalXSDHexBinary - A string containing lexical representation of xsd:hexBinary.

Returns:
   An array of bytes represented by the string argument.

Throws:
   IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:hexBinary.

---

public static long parseUnsignedInt(String lexicalXSDUnsignedInt)

long

   lexicalXSDUnsignedInt     xsd:unsignedInt
   return                   long
   Throws                   NumberFormatException:

parseUnsignedInt

public static long parseUnsignedInt(String lexicalXSDUnsignedInt)

Converts the string argument into a long value.

Parameters:
   lexicalXSDUnsignedInt - A string containing lexical representation of xsd:unsignedInt.

Returns:
   A long value represented by the string argument.
public static int parseUnsignedShort(String lexicalXSDUnsignedShort)

int

lexicalXSDUnsignedShort  xsd:unsignedShort
return int

Throws: NumberFormatException - if string parameter can not be parsed into a long value.

parseUnsignedShort

public static int parseUnsignedShort(String lexicalXSDUnsignedShort)

Converts the string argument into an int value.

Parameters:

lexicalXSDUnsignedShort - A string containing lexical representation of xsd:unsignedShort.

Returns:

An int value represented by the string argument.

Throws:

NumberFormatException - if string parameter can not be parsed into an int value.

public static java.util.Calendar parseTime(String lexicalXSDTime)

Calendar

lexicalXSDTime  xsd:time
return Calendar
parseTime

public static Calendar parseTime(String lexicalXSDTime)

Converts the string argument into a Calendar value.

Parameters:

lexicalXSDTime - A string containing lexical representation of xsd:time.

Returns:

A Calendar value represented by the string argument.

Throws:

IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:Time.

parseDate

public static java.util.Calendar parseDate(String lexicalXSDDate)

Calendar

lexicalXSDDate     xsd:Date
return            Calendar

Throws

IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes xsd:Date.
Parameters:
  lexicalXSDDate - A string containing lexical representation of xsd:Date.

Returns:
  A Calendar value represented by the string argument.

Throws:
  IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:Date.

public static String parseAnySimpleType(String lexicalXSDAnySimpleType)

    lexicalXSDAnySimpleType
    return

parseAnySimpleType

public static String parseAnySimpleType(String lexicalXSDAnySimpleType)

    Return a string containing the lexical representation of the simple type.

Parameters:
  lexicalXSDAnySimpleType - A string containing lexical representation of the simple type.

Returns:
  A string containing the lexical representation of the simple type.

public static String printString(String val)

    val
printString

public static String printString(String val)

Converts the string argument into a string.

Parameters:
   val - A string value.

Returns:
   A string containing a lexical representation of xsd:string.

printInteger

public static String printInteger(java.math.BigInteger val)

BigInteger

val

return xsd:integer

Throws
   IllegalArgument exception: val null

printInteger

public static String printInteger(BigInteger val)

Converts a BigInteger value into a string.

Parameters:
   val - A BigInteger value

Returns:
   A string containing a lexical representation of xsd:integer

Throws:
   IllegalArgumentException - val is null.
public static String printInt(int val)

int val
return int

public static String printInt(int val)

Converts an int value into a string.

Parameters:
val - An int value

Returns:
A string containing a lexical representation of xsd:int

public static String printLong(long val)

long val
return long

public static String printLong(long val)

Converts a long value into a string.

Parameters:
val - A long value

Returns:
A string containing a lexical representation of xsd:long
public static String printShort(short val)

short val
return short
-xsd:short

printShort

public static String printShort(short val)

Converts a short value into a string.

**Parameters:**
val - A short value

**Returns:**
A string containing a lexical representation of xsd:short

public static String printDecimal(java.math.BigDecimal val)

BigDecimal val
BigDecimal return
-xsd:decimal
Throws IllegalArgumentException: val null

printDecimal

public static String printDecimal(BigDecimal val)

Converts a BigDecimal value into a string.

**Parameters:**
val - A BigDecimal value
public static String printFloat(float val)

float
val float

return xsd:float

printFloat

Parameters:
val - A float value

Returns:
A string containing a lexical representation of xsd:float

public static String printDouble(double val)

double
val double

return xsd:double

printDouble

Parameters:
val - A double value

Returns:
A string containing a lexical representation of xsd:double

Returns:
A string containing a lexical representation of xsd:decimal

Throws:
IllegalArgumentException - val is null.
Converts a double value into a string.

**Parameters:**
val - A double value

**Returns:**
A string containing a lexical representation of xsd:double

---

**public static String printBoolean(boolean val)**

```java
boolean val
return xsd:boolean
```

**printBoolean**

```java
public static String printBoolean(boolean val)

    Converts a boolean value into a string.

    **Parameters:**
    val - A boolean value

    **Returns:**
    A string containing a lexical representation of xsd:boolean
```

---

**public static String printByte(byte val)**

```java
byte val
return xsd:byte
```

**printByte**
public static String printByte(byte val)

Converts a byte value into a string.

**Parameters:**
val - A byte value

**Returns:**
A string containing a lexical representation of xsd:byte

public static String printQName(javax.xml.namespace.QName val, javax.xml.namespace.NamespaceContext nsc)

**QName**

<table>
<thead>
<tr>
<th>val</th>
<th>QName</th>
</tr>
</thead>
<tbody>
<tr>
<td>nsc</td>
<td>QName</td>
</tr>
<tr>
<td>return</td>
<td>QName</td>
</tr>
</tbody>
</table>

**Throws**

IllegalArgumentException: val null nsc null
nsc.getPrefix(nsPrefixFromVal) null

**printQName**

public static String printQName(QName val, NamespaceContext nsc)

Converts a QName instance into a string.

**Parameters:**
val - A QName value
nsc - A namespace context for interpreting a prefix within a QName.

**Returns:**
A string containing a lexical representation of QName

**Throws:**

IllegalArgumentException - if val is null or if nsc is non-null or
public static String printDateTime(java.util.Calendar val)

Converts a Calendar value into a string.

Parameters:
val - A Calendar value

Returns:
A string containing a lexical representation of xsd:dateTime

Throws:
IllegalArgumentException - if val is null.

public static String printBase64Binary(byte[] val)

Throws: IllegalArgumentException: val null

printBase64Binary
public static String printBase64Binary(byte[] val)

Converts an array of bytes into a string.

Parameters:
val - An array of bytes

Returns:
A string containing a lexical representation of xsd:base64Binary

Throws:
IllegalArgumentException - if val is null.

public static String printHexBinary(byte[] val)

val
return xsd:hexBinary

Throws IllegalArgumentException: val null

printHexBinary

public static String printHexBinary(byte[] val)

Converts an array of bytes into a string.

Parameters:
val - An array of bytes

Returns:
A string containing a lexical representation of xsd:hexBinary

Throws:
IllegalArgumentException - if val is null.

public static String printUnsignedInt(long val)

long
public static String printUnsignedInt(long val)

Converts a long value into a string.

Parameters:
  val - A long value

Returns:
  A string containing a lexical representation of xsd:unsignedInt

public static String printUnsignedShort(int val)

Converts an int value into a string.

Parameters:
  val - An int value

Returns:
  A string containing a lexical representation of xsd:unsignedShort

public static String printTime(java.util.Calendar val)
Calendar

    val Calendar
    return xsd:time
    Throws IllegalArgumentException: val null

printTime

public static String printTime(Calendar val)

    Converts a Calendar value into a string.

Parameters:
    val - A Calendar value

Returns:
    A string containing a lexical representation of xsd:time

Throws:
    IllegalArgumentException - if val is null.

public static String printDate(java.util.Calendar val)

Calendar

    val Calendar
    return xsd:date
    Throws IllegalArgumentException: val null

printDate

public static String printDate(Calendar val)

    Converts a Calendar value into a string.

Parameters:
    val - A Calendar value
public static String printAnySimpleType(String val)

val
return xsd:AnySimpleType

printAnySimpleType

public static String printAnySimpleType(String val)

Converts a string value into a string.

Parameters:
val - A string value

Returns:
A string containing a lexical representation of xsd:AnySimpleType
javax.xml.bind **Interface DatatypeConverterInterface**

```java
public interface DatatypeConverterInterface

DatatypeConverterInterface JAXB JAXB JAXB
JAXBContext.newInstance
DatatypeConverter.setDatatypeConverter api
DatatypeConverter

parse print JAXB Table 5-1 XML parse print

parse print DatatypeConverter parse print

XML parse XML
XML specification XML IllegalArgumentException
IllegalArgumentException

XML print XML IllegalArgumentException
IllegalArgumentException

xsd XML

XML Schema Part 2: Datatypes specification

See also

javax.xml.bind.DatatypeConverter,
javax.xml.bind.ParseConversionEvent,
javax.xml.bind.PrintConversionEvent

$Revision: 1.5 $ en

JAXB1.0

The DatatypeConverterInterface is for JAXB provider use only. A JAXB provider must supply a class that implements this interface. JAXB Providers are required to call the
DatatypeConverter.setDatatypeConverter api at some point before the
first marshal or unmarshal operation (perhaps during the call to JAXBContext.newInstance). This step is necessary to configure the converter that should be used to perform the print and parse functionality. Calling this api repeatedly will have no effect - the DatatypeConverter instance passed into the first invocation is the one that will be used from then on.

This interface defines the parse and print methods. There is one parse and print method for each XML schema datatype specified in the the default binding Table 5-1 in the JAXB specification.

The parse and print methods defined here are invoked by the static parse and print methods defined in the DatatypeConverter class.

A parse method for a XML schema datatype must be capable of converting any lexical representation of the XML schema datatype (specified by the XML Schema Part2: Datatypes specification) into a value in the value space of the XML schema datatype. If an error is encountered during conversion, then an IllegalArgumentException or a subclass of IllegalArgumentException must be thrown by the method.

A print method for a XML schema datatype can output any lexical representation that is valid with respect to the XML schema datatype. If an error is encountered during conversion, then an IllegalArgumentException, or a subclass of IllegalArgumentException must be thrown by the method.

The prefix xsd: is used to refer to XML schema datatypes XML Schema Part2: Datatypes specification.

Since: JAXB1.0
Version: $Revision: 1.5 $
Author:
  - Sekhar Vajjhala, Sun Microsystems, Inc.
  - Joe Fialli, Sun Microsystems Inc.
  - Kohsuke Kawaguchi, Sun Microsystems, Inc.
  - Ryan Shoemaker, Sun Microsystems Inc.
### Method Summary

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Method Name (Parameters)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><code>parseAnySimpleType</code> <em>(String lexicalXSDAnySimpleType)</em></td>
<td>Return a string containing the lexical representation of the simple type.</td>
</tr>
<tr>
<td>byte[]</td>
<td><code>parseBase64Binary</code> <em>(String lexicalXSDBase64Binary)</em></td>
<td>Converts the string argument into an array of bytes.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>parseBoolean</code> <em>(String lexicalXSDBoolean)</em></td>
<td>Converts the string argument into a boolean value.</td>
</tr>
<tr>
<td>byte</td>
<td><code>parseByte</code> <em>(String lexicalXSDByte)</em></td>
<td>Converts the string argument into a byte value.</td>
</tr>
<tr>
<td>Calendar</td>
<td><code>parseDate</code> <em>(String lexicalXSDDate)</em></td>
<td>Converts the string argument into a Calendar value.</td>
</tr>
<tr>
<td>Calendar</td>
<td><code>parseDateTime</code> <em>(String lexicalXSDDateTime)</em></td>
<td>Converts the string argument into a Calendar value.</td>
</tr>
<tr>
<td>BigDecimal</td>
<td><code>parseDecimal</code> <em>(String lexicalXSDDecimal)</em></td>
<td>Converts the string argument into a BigDecimal value.</td>
</tr>
<tr>
<td>double</td>
<td><code>parseDouble</code> <em>(String lexicalXSDDouble)</em></td>
<td>Converts the string argument into a double value.</td>
</tr>
<tr>
<td>float</td>
<td><code>parseFloat</code> <em>(String lexicalXSDFloat)</em></td>
<td>Converts the string argument into a float value.</td>
</tr>
<tr>
<td>byte[]</td>
<td><code>parseHexBinary</code> <em>(String lexicalXSDHexBinary)</em></td>
<td>Converts the string argument into an array of bytes.</td>
</tr>
<tr>
<td>int</td>
<td><code>parseInt</code> <em>(String lexicalXSDInt)</em></td>
<td>Convert the string argument into an int value.</td>
</tr>
<tr>
<td>BigInteger</td>
<td><code>parseInteger</code> <em>(String lexicalXSDInteger)</em></td>
<td>Convert the string argument into a BigInteger value.</td>
</tr>
<tr>
<td>long</td>
<td><code>parseLong</code> <em>(String lexicalXSDLong)</em></td>
<td>Converts the string argument into a long value.</td>
</tr>
<tr>
<td>QName</td>
<td><code>parseQName</code> <em>(String lexicalXSDQName, NamespaceContext nsc)</em></td>
<td>Converts the string argument into a QName value.</td>
</tr>
<tr>
<td>Type</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>--------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>short</td>
<td><code>parseShort(String lexicalXSDShort)</code></td>
<td>Converts the string argument into a short value.</td>
</tr>
<tr>
<td>String</td>
<td><code>parseString(String lexicalXSDString)</code></td>
<td>Convert the string argument into a string.</td>
</tr>
<tr>
<td>Calendar</td>
<td><code>parseTime(String lexicalXSDTime)</code></td>
<td>Converts the string argument into a Calendar value.</td>
</tr>
<tr>
<td>long</td>
<td><code>parseUnsignedInt(String lexicalXSDUnsignedInt)</code></td>
<td>Converts the string argument into a long value.</td>
</tr>
<tr>
<td>int</td>
<td><code>parseUnsignedShort(String lexicalXSDUnsignedShort)</code></td>
<td>Converts the string argument into an int value.</td>
</tr>
<tr>
<td>String</td>
<td><code>printAnySimpleType(String val)</code></td>
<td>Converts a string value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printBase64Binary(byte[] val)</code></td>
<td>Converts an array of bytes into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printBoolean(boolean val)</code></td>
<td>Converts a boolean value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printByte(byte val)</code></td>
<td>Converts a byte value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printDate(Calendar val)</code></td>
<td>Converts a Calendar value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printDateTime(Calendar val)</code></td>
<td>Converts a Calendar value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printDecimal(BigDecimal val)</code></td>
<td>Converts a BigDecimal value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printDouble(double val)</code></td>
<td>Converts a double value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printFloat(float val)</code></td>
<td>Converts a float value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printHexBinary(byte[] val)</code></td>
<td>Converts an array of bytes into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printInt(int val)</code></td>
<td>Converts an int value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printInteger(BigInteger val)</code></td>
<td>Converts a BigInteger value into a string.</td>
</tr>
<tr>
<td>String</td>
<td><code>printLong</code>(long val)</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Converts a long value into a string.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printQName</code>(QName val, NamespaceContext nsc)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts a QName instance into a string.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printShort</code>(short val)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts a short value into a string.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printString</code>(String val)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts the string argument into a string.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printTime</code>(Calendar val)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts a Calendar value into a string.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printUnsignedInt</code>(long val)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts a long value into a string.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>printUnsignedShort</code>(int val)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Converts an int value into a string.</td>
</tr>
</tbody>
</table>

**Method Detail**

**public String parseString(String lexicalXSDString)**

```java
lexicalXSDString
XML xsd:string
```

**parseString**

`String parseString(String lexicalXSDString)`

Convert the string argument into a string.

**Parameters:**

- `lexicalXSDString` - A lexical representation of the XML Schema datatype `xsd:string`
Returns:
A string that is the same as the input string.

public java.math.BigInteger parseInteger(String lexicalXSDInteger)

BigInteger
lexicalXSDInteger    xsd:integer
return
BigInteger

Throws
NumberFormatException: lexicalXSDInteger
java.math.BigInteger

parseInteger

BigInteger parseInteger(String lexicalXSDInt)

int
lexicalXSDInt    xsd:int
return
int

Throws
NumberFormatException: lexicalXSDInt
java.math.BigInteger

public int parseInt(String lexicalXSDInt)
**parseInt**

```java
int parseInt(String lexicalXSDInt)
```

Convert the string argument into an int value.

**Parameters:**
- `lexicalXSDInt` - A string containing a lexical representation of `xsd:int`.

**Returns:**
An int value represented byte the string argument.

**Throws:**
- `NumberFormatException` - `lexicalXSDInt` is not a valid string representation of an int value.

---

**public long parseLong(String lexicalXSDLong)**

```java
long parseLong(String lexicalXSDLong)
```

Converts the string argument into a long value.

**Parameters:**
- `lexicalXSDLong` - A string containing lexical representation of `xsd:long`.

**Returns:**

---

**parseLong**

```java
long parseLong(String lexicalXSDLong)
```

Converts the string argument into a long value.

**Parameters:**
- `lexicalXSDLong` - A string containing lexical representation of `xsd:long`.

**Returns:**
A long value represented by the string argument.

Throws:

- `NumberFormatException` - `lexicalXSDLong` is not a valid string representation of a long value.

public short parseShort(String lexicalXSDShort)

short

`lexicalXSDShort`  xsd:short

return  short

Throws  `NumberFormatException`:  `lexicalXSDShort`  short

parseShort

short  parseShort(String lexicalXSDShort)

Converts the string argument into a short value.

Parameters:

- `lexicalXSDShort` - A string containing lexical representation of xsd:short.

Returns:

A short value represented by the string argument.

Throws:

- `NumberFormatException` - `lexicalXSDShort` is not a valid string representation of a short value.

public java.math.BigDecimal parseDecimal(String lexicalXSDDecimal)

BigDecimal

`lexicalXSDDecimal`  xsd:decimal

return  BigDecimal
**parseDecimal**

**BigDecimal** `parseDecimal(String lexicalXSDDecimal)`

Converts the string argument into a BigDecimal value.

**Parameters:**
- `lexicalXSDDecimal` - A string containing lexical representation of `xsd:decimal`.

**Returns:**
A BigDecimal value represented by the string argument.

**Throws:**
- `NumberFormatException` - `lexicalXSDDecimal` is not a valid string representation of `BigDecimal`.

---

**public float parseFloat(String lexicalXSDFloat)**

**float**

`lexicalXSDFloat` - `xsd:float`

**return**
- float

**Throws**
- `NumberFormatException` - `lexicalXSDFloat` is not a valid string representation of `BigDecimal`.

---

**parseFloat**

**float** `parseFloat(String lexicalXSDFloat)`

Converts the string argument into a float value.

**Parameters:**
lexicalXSDFloat - A string containing lexical representation of xsd:float.

Returns:
A float value represented by the string argument.

Throws:
NumberFormatException - lexicalXSDFloat is not a valid string representation of a float value.

public double parseDouble(String lexicalXSDDouble)

double

lexicalXSDDouble - A string containing lexical representation of xsd:double.

Returns:
A double value represented by the string argument.

Throws:
NumberFormatException - lexicalXSDDouble is not a valid string representation of a double value.

public boolean parseBoolean(String lexicalXSDBoolean)

boolean

parseBoolean

double parseDouble(String lexicalXSDDouble)

Converts the string argument into a double value.

Parameters:
lexicalXSDDouble - A string containing lexical representation of xsd:double.

Returns:
A double value represented by the string argument.

Throws:
NumberFormatException - lexicalXSDDouble is not a valid string representation of a double value.
### parseBoolean

**java**

```java
public boolean parseBoolean(String lexicalXSDBoolean)
```

Converts the string argument into a boolean value.

**Parameters:**
- `lexicalXSDBoolean` - A string containing lexical representation of `xsd:boolean`.

**Returns:**
A boolean value represented by the string argument.

**Throws:**
- `IllegalArgumentException` - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for `xsd:boolean`.

---

### parseByte

**java**

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```

```java
public byte parseByte(String lexicalXSDByte)
```
Converts the string argument into a byte value.

**Parameters:**
- `lexicalXSDByte` - A string containing lexical representation of xsd:byte.

**Returns:**
A byte value represented by the string argument.

**Throws:**
- `NumberFormatException` - `lexicalXSDByte` does not contain a parseable byte.
- `IllegalArgumentException` - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:byte.

```java
public javax.xml.namespace.QName parseQName(String lexicalXSDQName, javax.xml.namespace.NamespaceContext nsc)
```

**QName**

- `lexicalXSDQname`  
  - **XML Schema Part 2: Datatypes specification:** QNames
  - `lexicalXSDQName`  
    - xsd:QName
  - `ns`  
    - QName
  - `return`  
    - QName
  - `Throws`  
    - `IllegalArgumentException`  
      - XML Schema Part 2

**parseQName**

```java
QName parseQName(String lexicalXSDQName, NamespaceContext nsc)
```

Converts the string argument into a QName value.
String parameter `lexicalXSDQName` must conform to lexical value space specified at [XML Schema Part 2: Datatypes specification: QNames](https://www.w3.org/2001/XMLSchema#)

**Parameters:**
- `lexicalXSDQName` - A string containing lexical representation of `xsd:QName`.
- `nsc` - A namespace context for interpreting a prefix within a QName.

**Returns:**
A QName value represented by the string argument.

**Throws:**
- `IllegalArgumentException` - if string parameter does not conform to XML Schema Part 2 specification or if namespace prefix of `lexicalXSDQName` is not bound to a URI in NamespaceContext `nsc`.

```java
public java.util.Calendar parseDateTime(String lexicalXSDDateTime) {
    // Convert the string argument into a Calendar value.

    lexicalXSDDateTime
    xsd:datetime
    return
    Calendar
    Throws
    IllegalArgumentException: XML Schema Part 2: Datatypes xsd:dateTime
}
```

**parseDateTime**

```java
Calendar parseDateTime(String lexicalXSDDateTime) {
    // Converts the string argument into a Calendar value.

    lexicalXSDDateTime
    xsd:datetime
    return
    Calendar
    Throws
    IllegalArgumentException: XML Schema Part 2: Datatypes xsd:dateTime
}
```
Returns:
A Calendar object represented by the string argument.

Throws:
IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:dateTime.

public byte[] parseBase64Binary(String lexicalXSDBase64Binary)

lexicalXSDBase64Binary    xsd:base64Binary

return

Throws
IllegalArgumentException: XML Schema Part 2: Datatypes xsd:base64Binary

parseBase64Binary

byte[] parseBase64Binary(String lexicalXSDBase64Binary)

Converts the string argument into an array of bytes.

Parameters:
lexicalXSDBase64Binary - A string containing lexical representation of xsd:base64Binary.

Returns:
An array of bytes represented by the string argument.

Throws:
IllegalArgumentException - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:base64Binary

public byte[] parseHexBinary(String lexicalXSDHexBinary)
**parseHexBinary**

```java
byte[] parseHexBinary(String lexicalXSDHexBinary)
```

Converts the string argument into an array of bytes.

**Parameters:**
- `lexicalXSDHexBinary` - A string containing lexical representation of `xsd:hexBinary`.

**Returns:**
- An array of bytes represented by the string argument.

**Throws:**
- `IllegalArgumentException` - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for `xsd:hexBinary`.

---

**public long parseUnsignedInt(String lexicalXSDUnsignedInt)**

```java
long
```

**Parameters:**
- `lexicalXSDUnsignedInt` - A string containing lexical representation of `xsd:unsignedInt`.

**Returns:**
- `long` value represented by the string argument.

**Throws:**
- `NumberFormatException` - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for `xsd:unsignedInt`.
long parseUnsignedInt(String lexicalXSDUnsignedInt)

Converts the string argument into a long value.

**Parameters:**
- lexicalXSDUnsignedInt - A string containing lexical representation of xsd:unsignedInt.

**Returns:**
- A long value represented by the string argument.

**Throws:**
- `NumberFormatException` - if string parameter can not be parsed into a long value.

---

public int parseUnsignedShort(String lexicalXSDUnsignedShort)

**Parameters:**
- lexicalXSDUnsignedShort - A string containing lexical representation of xsd:unsignedShort.

**Returns:**
- An int value represented by the string argument.

**Throws:**
- `NumberFormatException` - if string parameter can not be parsed into an int value.

---

int parseUnsignedShort(String lexicalXSDUnsignedShort)

Converts the string argument into an int value.

**Parameters:**
- lexicalXSDUnsignedShort - A string containing lexical representation of xsd:unsignedShort.

**Returns:**
- An int value represented by the string argument.

**Throws:**
- `NumberFormatException` - if string parameter can not be parsed into an int value.
public java.util.Calendar parseTime(String lexicalXSDTime)

Calendar

    lexicalXSDTime    xsd:Time
    return            Calendar
    Throws             IllegalArgumentException: XML Schema Part 2:
                        Datatypes xsd:Time

parseTime

Calendar parseTime(String lexicalXSDTime)

Converts the string argument into a Calendar value.

Parameters:
    lexicalXSDTime - A string containing lexical representation of
                    xsd:Time.

Returns:
    A Calendar value represented by the string argument.

Throws:
    IllegalArgumentException - if string parameter does not conform
to lexical value space defined in XML Schema Part 2: Datatypes
for xsd:Time.

public java.util.Calendar parseDate(String lexicalXSDDate)

Calendar

    lexicalXSDDate    xsd:Date
    return            Calendar
    Throws             IllegalArgumentException: XML Schema Part 2:
                        Datatypes xsd:Date
parseDate

`Calendar parseDate(String lexicalXSDDate)`

Converts the string argument into a Calendar value.

**Parameters:**
- `lexicalXSDDate` - A string containing lexical representation of xsd:Date.

**Returns:**
- A Calendar value represented by the string argument.

**Throws:**
- `IllegalArgumentException` - if string parameter does not conform to lexical value space defined in XML Schema Part 2: Datatypes for xsd:Date.

---

public String parseAnySimpleType(String lexicalXSDAnySimpleType)

`lexicalXSDAnySimpleType`

`return`

parseAnySimpleType

`String parseAnySimpleType(String lexicalXSDAnySimpleType)`

Return a string containing the lexical representation of the simple type.

**Parameters:**
- `lexicalXSDAnySimpleType` - A string containing lexical representation of the simple type.

**Returns:**
- A string containing the lexical representation of the simple type.
public String printString(String val)

    val
    return xsd:string

printString

String printString(String val)

    Converts the string argument into a string.

Parameters:
    val - A string value.

Returns:
    A string containing a lexical representation of xsd:string

public String printInteger(java.math.BigInteger val)

BigInteger

    val
    return xsd:integer
    Throws IllegalArgumentException: val null

printInteger

String printInteger(BigInteger val)

    Converts a BigInteger value into a string.

Parameters:
    val - A BigInteger value
Returns:
A string containing a lexical representation of xsd:integer

Throws:
IllegalArgumentException - val is null.

---

**public String printInt(int val)**

```
int
    val int
    return xsd:int
```

**printInt**

*String printInt(int val)*

Converts an int value into a string.

**Parameters:**
val - An int value

**Returns:**
A string containing a lexical representation of xsd:int

---

**public String printLong(long val)**

```
long
    val long
    return xsd:long
```

**printLong**

*String printLong(long val)*
Converts a long value into a string.

**Parameters:**
val - A long value

**Returns:**
A string containing a lexical representation of xsd:long

```
public String printShort(short val)
```

short
val short
return xsd:short

**printShort**

`String printShort(short val)`

Converts a short value into a string.

**Parameters:**
val - A short value

**Returns:**
A string containing a lexical representation of xsd:short

```
public String printDecimal(java.math.BigDecimal val)
```

BigDecimal
val BigDecimal
return xsd:decimal
Throws IllegalArgumentException: val null
printDecimal

String printDecimal(BigDecimal val)

Converts a BigDecimal value into a string.

**Parameters:**

val - A BigDecimal value

**Returns:**

A string containing a lexical representation of xsd:decimal

**Throws:**

IllegalArgumentException - val is null.

---

public String printFloat(float val)

float

val float

return xsd:float

---

printFloat

String printFloat(float val)

Converts a float value into a string.

**Parameters:**

val - A float value

**Returns:**

A string containing a lexical representation of xsd:float

---

public String printDouble(double val)

double

val double
printDouble

String printDouble(double val)

Converts a double value into a string.

Parameters:
val - A double value

Returns:
A string containing a lexical representation of xsd:double

public String printBoolean(boolean val)

boolean val

return xsd:boolean

printBoolean

String printBoolean(boolean val)

Converts a boolean value into a string.

Parameters:
val - A boolean value

Returns:
A string containing a lexical representation of xsd:boolean

public String printByte(byte val)
printByte

String printByte(byte val)

Converts a byte value into a string.

Parameters:
- val - A byte value

Returns:
A string containing a lexical representation of xsd:byte

public String printQName(javax.xml.namespace.QName val, javax.xml.namespace.NamespaceContext nsc)

QName

String printQName(QName val, NamespaceContext nsc)

Converts a QName instance into a string.

Parameters:
val - A QName value
nsc - A namespace context for interpreting a prefix within a QName.

Returns:
A string containing a lexical representation of QName

Throws:
IllegalArgumentException - if val is null or if nsc is non-null or
nsc.getPrefix(nsPrefixFromVal) is null.

public String printDateTime(java.util.Calendar val)

Calendar
  val Calendar
  return xsd:dateTime
  Throws IllegalArgument Exception: val null

printDateTime

String printDateTime(Calendar val)

Converts a Calendar value into a string.

Parameters:
val - A Calendar value

Returns:
A string containing a lexical representation of xsd:dateTime

Throws:
IllegalArgumentException - if val is null.

public String printBase64Binary(byte[] val)

val byte[]
return xsd:base64Binary
printBase64Binary

String printBase64Binary(byte[] val)

Converts an array of bytes into a string.

Parameters:
val - an array of bytes

Returns:
A string containing a lexical representation of xsd:base64Binary

Throws:
IllegalArgumentException - if val is null.

printHexBinary

String printHexBinary(byte[] val)

val
return xsd:hexBinary
Throws IllegalArgumentException: val null

printHexBinary

String printHexBinary(byte[] val)

Converts an array of bytes into a string.

Parameters:
val - an array of bytes

Returns:
A string containing a lexical representation of xsd:hexBinary

Throws:
IllegalArgumentException - if val is null.

public String printUnsignedInt(long val)

long
val
return

printUnsignedInt

String printUnsignedInt(long val)

Converts a long value into a string.

Parameters:
val - A long value

Returns:
A string containing a lexical representation of xsd:unsignedInt

public String printUnsignedShort(int val)

int
val
return

printUnsignedShort

String printUnsignedShort(int val)

Converts an int value into a string.

Parameters:
val - An int value

**Returns:**
A string containing a lexical representation of xsd:unsignedShort

```java
public String printTime(java.util.Calendar val)
```

**Calendar**

```java
val Calendar
return xsd:time
Throws IllegalArgumentException: val null
```

**printTime**

```java
String printTime(Calendar val)
```

Converts a Calendar value into a string.

**Parameters:**
val - A Calendar value

**Returns:**
A string containing a lexical representation of xsd:time

**Throws:**
`IllegalArgumentException` - if val is null.

---

```java
public String printDate(java.util.Calendar val)
```

**Calendar**

```java
val Calendar
return xsd:date
Throws IllegalArgumentException: val null
```
printDate

String printDate(Calendar val)

Converts a Calendar value into a string.

Parameters:
val - A Calendar value

Returns:
A string containing a lexical representation of xsd:date

Throws:
IllegalArgumentException - if val is null.

public String printAnySimpleType(String val)

val
return xsd:AnySimpleType

printAnySimpleType

String printAnySimpleType(String val)

Converts a string value into a string.

Parameters:
val - A string value

Returns:
A string containing a lexical representation of xsd:AnySimpleType
javax.mail.search Class DateTerm

java.lang.Object
   | javax.mail.search.SearchTerm
      | javax.mail.search.ComparisonTerm
      | javax.mail.search.DateTerm

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   ReceivedDateTerm, SentDateTerm

public abstract class DateTerm extends ComparisonTerm

Extends: SearchTerm > ComparisonTerm
Extended by: ReceivedDateTerm, SentDateTerm

Date

This class implements comparisons for Dates

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected</th>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>The date.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.mail.search.ComparisonTerm
comparison, EQ, GE, GT, LE, LT, NE
### Constructor Summary

| protected DateTerm(int comparison, Date date) | Constructor. |

### Method Summary

<table>
<thead>
<tr>
<th>boolean equals(Object obj)</th>
<th>Equality comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getComparison()</td>
<td>Return the type of comparison.</td>
</tr>
<tr>
<td>Date getDate()</td>
<td>Return the Date to compare with.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>protected boolean match(Date d)</td>
<td>The date comparison method.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.search.SearchTerm

| match |

Methods inherited from class java.lang.Object

| clone, finalize, getClass, notify, notifyAll, toString, wait, wait |

### Field Detail

<table>
<thead>
<tr>
<th>date</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected Date date</td>
</tr>
</tbody>
</table>
The date.

**Constructor Detail**

protected `DateTerm(int comparison, java.util.Date date)`

```
comparison  
date        Date
```

**DateTerm**

protected `DateTerm(int comparison, Date date)`

Constructor.

**Parameters:**
- `comparison` - the comparison type
- `date` - The Date to be compared against

**Method Detail**

public `java.util.Date getDate()`

`Date`

**getDATE**

public `Date getDate()`

Return the Date to compare with.

public `int getComparison()`
getComparison

public int getComparison()

    Return the type of comparison.

protected boolean match(java.util.Date d)

    d
    return true false

match

protected boolean match(Date d)

    The date comparison method.

    Parameters:
    d - the date in the constructor is compared with this date

    Returns:
    true if the dates match, otherwise false

equals

equals

public boolean equals(Object obj)

    Equality comparison.

    Overrides:
public int hashCode()
hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

Overrides:
hashCode in class ComparisonTerm
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED  | FIELD | CONSTR | METHOD |

FRAMES  NO FRAMES
DETAIL: FIELD  | CONSTR | METHOD |
javax.faces.convert Class DateTimeConverter

java.lang.Object
   \ javax.faces.convert.DateTimeConverter

All Implemented Interfaces:
   StateHolder, Converter

public class DateTimeConverter
   extends Object
   implements Converter, StateHolder

Implements: Converter, StateHolder

java.util.Date Converter

getAsObject() String java.util.Date

   • String null null
   • String 0 null
   • locale null Locale UIViewRoot Locale
   • pattern java.text.SimpleDateFormat typedateStyle
timeStyle
   • pattern type dateStyle timeStyle
   • timezone DateFormat "GMT"

getAsString() java.util.Date String

   • null 0 String
   • String
   • locale null Locale UIViewRoot Locale
   • timezone DateFormat "GMT"
   • pattern java.text.SimpleDateFormat typedateStyle
timeStyle
   • pattern type String dateStyle timeStyle
Converter implementation for java.util.Date values.

The getAsObject() method parses a String into a java.util.Date, according to the following algorithm:

- If the specified String is null, return a null. Otherwise, trim leading and trailing whitespace before proceeding.
- If the specified String - after trimming - has a zero length, return null.
- If the locale property is not null, use that Locale for managing parsing. Otherwise, use the Locale from the UIViewRoot.
- If a pattern has been specified, its syntax must conform the rules specified by java.text.SimpleDateFormat. Such a pattern will be used to parse, and the type, dateStyle, and timeStyle properties will be ignored.
- If a pattern has not been specified, parsing will be based on the type property, which expects a date value, a time value, or both. Any date and time values included will be parsed in accordance to the styles specified by dateStyle and timeStyle, respectively.
- If a timezone has been specified, it must be passed to the underlying DateFormat instance. Otherwise the "GMT" timezone is used.
- In all cases, parsing must be non-lenient; the given string must strictly adhere to the parsing format.

The getAsString() method expects a value of type java.util.Date (or a subclass), and creates a formatted String according to the following algorithm:

- If the specified value is null, return a zero-length String.
- If the specified value is a String, return it unmodified.
- If the locale property is not null, use that Locale for managing formatting. Otherwise, use the Locale from the UIViewRoot.
- If a timezone has been specified, it must be passed to the underlying DateFormat instance. Otherwise the "GMT" timezone is used.
- If a pattern has been specified, its syntax must conform the rules
Specified by java.text.SimpleDateFormat. Such a pattern will be used to format, and the type, dateStyle, and timeStyle properties will be ignored.

- If a pattern has not been specified, formatting will be based on the type property, which includes a date value, a time value, or both into the formatted String. Any date and time values included will be formatted in accordance to the styles specified by dateStyle and timeStyle, respectively.

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td><strong>CONVERTER_ID</strong> The standard converter id for this converter.</td>
</tr>
<tr>
<td>static String</td>
<td><strong>DATE_ID</strong> The message identifier of the FacesMessage to be created if the conversion to Date fails.</td>
</tr>
<tr>
<td>static String</td>
<td><strong>DATETIME_ID</strong> The message identifier of the FacesMessage to be created if the conversion to DateTime fails.</td>
</tr>
<tr>
<td>static String</td>
<td><strong>STRING_ID</strong> The message identifier of the FacesMessage to be created if the conversion of the DateTime value to String fails.</td>
</tr>
<tr>
<td>static String</td>
<td><strong>TIME_ID</strong> The message identifier of the FacesMessage to be created if the conversion to Time fails.</td>
</tr>
</tbody>
</table>

### Constructor Summary

**DateTimeConverter()**

### Method Summary

**getAsObject(FacesContext context, UICOMPONENT component, String value)**
Convert the specified string value, which is associated with the specified `UIComponent`, into a model data object that is appropriate for being stored during the *Apply Request Values* phase of the request processing lifecycle.

```java
String getAsString(FacesContext context, UIComponent component, Object value)
```

Convert the specified model object value, which is associated with the specified `UIComponent`, into a String that is suitable for being included in the response generated during the *Render Response* phase of the request processing lifecycle.

```java
String getDateStyle()
```

Return the style to be used to format or parse dates.

```java
Locale getLocale()
```

Return the `Locale` to be used when parsing or formatting dates and times.

```java
String getPattern()
```

Return the format pattern to be used when formatting and parsing dates and times.

```java
String getTimeStyle()
```

Return the style to be used to format or parse times.

```java
TimeZone getTimeZone()
```

Return the `TimeZone` used to interpret a time value.

```java
String getType()
```

Return the type of value to be formatted or parsed.

```java
boolean isTransient()
```

If true, the Object implementing this interface must not participate in state saving or restoring.

```java
void restoreState(FacesContext context, Object state)
```

Perform any processing required to restore the state from the entries in the state Object.

```java
Object saveState(FacesContext context)
```

Gets the state of the instance as a `Serializable` Object.

```java
void setDateStyle(String dateStyle)
```

Set the style to be used to format or parse dates.

```java
void setLocale(Locale locale)
```

Set the `Locale` to be used when parsing or formatting
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void setPattern(String pattern)</code></td>
<td>Set the format pattern to be used when formatting and parsing dates and times.</td>
</tr>
<tr>
<td><code>void setTimeZone(TimeZone timeZone)</code></td>
<td>Set the TimeZone used to interpret a time value.</td>
</tr>
<tr>
<td><code>void setStyle(String timeStyle)</code></td>
<td>Set the style to be used to format or parse times.</td>
</tr>
<tr>
<td><code>void setTransient(boolean transientFlag)</code></td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
<tr>
<td><code>void setType(String type)</code></td>
<td>Set the type of value to be formatted or parsed.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

**Field Detail**

**CONVERTER_ID**

public static final String **CONVERTER_ID**

The standard converter id for this converter.

**See Also:**

[Constant Field Values](#)
DATE_ID

public static final String DATE_ID

The message identifier of the FacesMessage to be created if the conversion to Date fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

---

TIME_ID

public static final String TIME_ID

The message identifier of the FacesMessage to be created if the conversion to Time fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

---

DATETIME_ID
public static final String DATETIME_ID

The message identifier of the FacesMessage to be created if the conversion to DateTime fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the DateTime value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail

public DateTimeConverter()
DateTimeConverter

```java
public DateTimeConverter()
```

### Method Detail

**public String getDateStyle()**

default

**getDateTimeStyle**

```java
public String getDateStyle()
```

Return the style to be used to format or parse dates. If not set, the default value, default, is returned.

**public void setDateStyle(String dateStyle)**

```java
public void setDateStyle(String dateStyle)
```

default short medium long full getAsObject() getAsString() ConverterException

dateStyle

**setDateTimeStyle**

```java
public void setDateStyle(String dateStyle)
```

Set the style to be used to format or parse dates. Valid values are default, short, medium, long, and full. An invalid value will cause a `ConverterException` when `getAsObject()` or `getAsString()` is called.

**Parameters:**
public java.util.Locale getLocale()

Locale javax.faces.component.UUIViewRoot Locale

getLocale

public Locale getLocale()

Return the Locale to be used when parsing or formatting dates and times. If not explicitly set, the Locale stored in the UUIViewRoot for the current request is returned.

public void setLocale(java.util.Locale locale)

Locale null javax.faces.component.UUIViewRoot Locale

locale Locale null

setLocale

public void setLocale(Locale locale)

Set the Locale to be used when parsing or formatting dates and times. If set to null, the Locale stored in the UUIViewRoot for the current request will be utilized.

Parameters:
locale - The new Locale (or null)

public String getPattern()
getPattern

```java
public String getPattern()

    Return the format pattern to be used when formatting and parsing dates and times.
```

setPattern

```java
public void setPattern(String pattern)

    java.text.SimpleDateFormat getAsObject()
getAsString()    ConverterException

    pattern
```

public void setPattern(String pattern)

Set the format pattern to be used when formatting and parsing dates and times. Valid values are those supported by java.text.SimpleDateFormat. An invalid value will cause a ConverterException when getAsObject() or getAsString() is called.

Parameters:

    pattern - The new format pattern
```

public String getTimeStyle()

    default
getTimeStyle

public String getTimeStyle()

    Return the style to be used to format or parse times. If not set, the default value, default, is returned.

public void setTimeStyle(String timeStyle)

    default short medium long full getAsObject()
getAsString() ConverterException

timeStyle

setTimeStyle

public void setStyle(String timeStyle)

    Set the style to be used to format or parse times. Valid values are default, short, medium, long, and full. An invalid value will cause a ConverterException when getAsObject() or getAsString() is called.

    Parameters:
        timeStyle - The new style code

public java.util.TimeZone getTimeZone()

    TimeZone GMT

g getTimeZone

public TimeZone getTimeZone()
Return the TimeZone used to interpret a time value. If not explicitly set, the default time zone of GMT returned.

public void setTimeZone(java.util.TimeZone timeZone)

TimeZone

timeZone

setTimeZone

public void setTimeZone(TimeZone timeZone)

Set the TimeZone used to interpret a time value.

Parameters:

timeZone - The new time zone

public String getType()

date

g getType

public String getType()  

Return the type of value to be formatted or parsed. If not explicitly set, the default type, date is returned.

public void setType(String type)

bothdate time getAsObject() getAsString()
ConverterException

:type

**setType**

public void **set**Type(String type)

Set the type of value to be formatted or parsed. Valid values are both, date, or time. An invalid value will cause a ConverterException when getAsObject() or getAsString() is called.

**Parameters:**

type - The new date style

---

**getAsObject**

public Object **getAs**Object(FacesContext context, UIComponent component, String value)

**Throws** ConverterException: NullPointerException

Throws NullPointerException: NullPointerException

---

**getAsObject**

public Object **getAs**Object(FacesContext context, UIComponent component, String value)

**Description copied from interface:** Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Specified by:** getAsObject in interface Converter
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws ConverterException: NullPointerException
Throws NullPointerException: NullPointerException

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - String value to be converted (may be null)
associated value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null

public Object saveState(FacesContext context)

Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
public void restoreState(FacesContext context, Object state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: 
restoreState in interface StateHolder

public boolean isTransient()

isTransient

public boolean isTransient()

Description copied from interface: StateHolder

If true, the Object implementing this interface must not participate in state saving or restoring.
Specified by:

`isTransient` in interface `StateHolder`

public void `setTransient(boolean transientFlag)`

**setTransient**

public void `setTransient(boolean transientFlag)`

Description copied from interface: `StateHolder`

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

Specified by:

`setTransient` in interface `StateHolder`

Parameters:

- `transientFlag` - boolean pass `true` if this Object will participate in state saving or restoring, otherwise pass `false`.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/legal/license-agreements.html).
The DConfigBean is a deployment configuration bean (DConfigBean) that is associated with one or more deployment descriptor beans, (DDBean). A DConfigBean represents a logical grouping of deployment configuration data to be presented to the Deploier. A DConfigBean provides zero or more XPaths that identifies the XML information it requires. A DConfigBean may contain other DConfigBeans and regular JavaBeans. The top most DConfigBean is a DConfigBeanRoot object which represents a single XML instance document.

A DConfigBean is created by calling DConfigBean.getDConfigBean(DDBean) method, where DConfigBean is the object that provided the XPath which the DDBean represents.
A DConfigBean is a JavaBean component that presents the dynamic deployment configuration information for a J2EE plugin to the deployer. It is a JavaBean. The JavaBean architecture was chosen because of its versatility in providing both simple and complex components. JavaBeans also enable the development of property sheets and property editors, as well as sophisticated customization wizards.

It is expected that a plugin vendor will provide a Property Editor for any complex datatype in a DConfigBean that a deployer needs to edit through a property sheet. The Property Editor should be implemented and made available to a tool according to the guidelines defined in the JavaBeans API Specification version 1.01.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><strong>addPropertyChangeListener</strong>(PropertyChangeListener pcl)</td>
</tr>
<tr>
<td></td>
<td>Register a property listener for this bean.</td>
</tr>
<tr>
<td><strong>DConfigBean</strong></td>
<td><strong>getDConfigBean</strong>(DDBean bean)</td>
</tr>
<tr>
<td></td>
<td>Return the JavaBean containing the server-specific deployment configuration information based upon the XML data provided by the DDBean.</td>
</tr>
<tr>
<td><strong>DDBean</strong></td>
<td><strong>getDDBean</strong>( )</td>
</tr>
<tr>
<td></td>
<td>Return the JavaBean containing the deployment descriptor XML text associated with this DConfigBean.</td>
</tr>
<tr>
<td><strong>String[ ]</strong></td>
<td><strong>getXpaths</strong>( )</td>
</tr>
<tr>
<td></td>
<td>Return a list of XPaths designating the deployment descriptor information this DConfigBean requires.</td>
</tr>
<tr>
<td>void</td>
<td><strong>notifyDDChange</strong>(XpathEvent event)</td>
</tr>
<tr>
<td></td>
<td>A notification that the DDBean provided in the event has changed and this bean or its child beans need to reevaluate themselves.</td>
</tr>
<tr>
<td>void</td>
<td><strong>removeDConfigBean</strong>(DConfigBean bean)</td>
</tr>
<tr>
<td></td>
<td>Remove a child DConfigBean from this bean.</td>
</tr>
<tr>
<td>void</td>
<td><strong>removePropertyChangeListener</strong>(PropertyChangeListener pcl)</td>
</tr>
<tr>
<td></td>
<td>Unregister a property listener for this bean.</td>
</tr>
</tbody>
</table>
public **DDBean** getDDBean()

DConfigBean XML JavaBean

return DConfigBean XML Bean

getDDBean

**DDBean** getDDBean()

Return the JavaBean containing the deployment descriptor XML text associated with this DConfigBean.

**Returns:**

The bean class containing the XML text for this DConfigBean.

---

public **String[]** getXpaths()

DConfigBean XPath

String X

return XML XPath String 'null'

g getXpaths

**String[]** getXpaths()

Return a list of XPaths designating the deployment descriptor information this DConfigBean requires. A given server vendor will need to specify some server-specific information. Each String returned by this method is an XPath describing a certain portion of the standard deployment descriptor for which there is corresponding server-specific configuration.

**Returns:**

a list of XPath Strings representing XML data to be retrieved or 'null' if there are none.
public **DConfigBean** getDConfigBean(**DDBean** bean) throws **ConfigurationException**

JavaBean DDBean XML

    return Bean DConfigBean
    bean XML DDBean

Throws **ConfigurationException**: Bean DDBean

### getDConfigBean

**DConfigBean** getDConfigBean(**DDBean** bean) throws **ConfigurationException**

Return the JavaBean containing the server-specific deployment configuration information based upon the XML data provided by the DDBean.

**Parameters:**

- **bean** - The DDBean containing the XML data to be evaluated.

**Returns:**

The DConfigBean to display the server-specific properties for the standard bean.

**Throws:**

- **ConfigurationException** - reports errors in generating a configuration bean. This DDBean is considered undeployable to this server until this exception is resolved. A suitably descriptive message is required so the user can diagnose the error.

---

public void removeDConfigBean(**DConfigBean** bean) throws **BeanNotFoundException**

Bean DConfigBean

    bean DConfigBean

Throws **BeanNotFoundException**: Bean Bean
removeDConfigBean

void removeDConfigBean(DConfigBean bean)
    throws BeanNotFoundException

Remove a child DConfigBean from this bean.

Parameters:
    bean - The child DConfigBean to be removed.

Throws:
    BeanNotFoundException - the bean provided is not in the child list of this bean.

public void notifyDDChange(XpathEvent event)
DDBean Bean Bean
    event DDBean

notifyDDChange

void notifyDDChange(XpathEvent event)

A notification that the DDBean provided in the event has changed and this bean or its child beans need to reevaluate themselves.

Parameters:
    event - an event containing a reference to the DDBean which has changed.

public void
addPropertyChangeListener(java.beans.PropertyChangeListener pcl)
Bean Bean
    pcl PropertyChangeListener

addPropertyChangeListener
addPropertyChangeListener

void addPropertyChangeListener(PropertyChangeListener pcl)

Register a property listener for this bean.

Parameters:

 pcl - PropertyChangeListener to add

public void removePropertyChangeListener(java.beans.PropertyChangeListener pcl)

Unregister a property listener for this bean.

Parameters:

 pcl - Listener to remove.
javax.enterprise.deploy.spi  Interface DConfigBeanRoot

All Superinterfaces:
DConfigBean

public interface DConfigBeanRoot
extends DConfigBean

Implements: DConfigBean

J2EE 0 XML 0 XML
Web

DConfigBeanRoot  Bean (DConfigBean)
DeploymentConfiguration.getDConfigBean(DDBeanRoot)
DDBeanRoot

DConfigBean  DConfigBeanRoot.getDConfigBean(DDBeanRoot)
DConfigBean  DConfigBeanRoot DDBeanRoot
webservice.xml  webserviceclient.xml

DDBeanRoot  XML
DeployableObject.getDDBeanRoot(String)  String WSDL
XML

A J2EE component module consists of one or more deployment descriptor files and zero or more non-deployment descriptor XML instance documents. A module must contain a component-specific deployment descriptor file (see the component specification for details). It may contain one or more secondary deployment descriptor files that define extra functionality on the component and zero or more non-deployment descriptor XML instance documents (see the Web Services specification).

The DConfigBeanRoot object is a deployment configuration bean
A DConfigBean object is associated with a deployment descriptor that extends a component's functionality. It must be created by calling the DConfigBeanRoot.getDConfigBean(DDBeanRoot) method. This DConfigBean object is a child of the component's DConfigBeanRoot object. The DDBeanRoot argument represents the secondary deployment descriptor. Deployment descriptor files such as webservice.xml and webserviceclient.xml are examples of secondary deployment descriptor files.

The server plugin must request a DDBeanRoot object for any non-deployment descriptor XML instance document data it requires. The plugin must call method DeployableObject.getDDBeanRoot(String) where String is the full path name from the root of the module to the file to be represented. A WSDL file is an example of a non-deployment descriptor XML instance document.

Author:
gfink

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getDConfigBean</strong>(DDBeanRoot ddBeanRoot)</td>
</tr>
<tr>
<td>DConfigBean</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.enterprise.deploy.spi.DConfigBean</td>
</tr>
<tr>
<td>addPropertyChangeListener, getDConfigBean, getDDBean, getXpaths, notifyDDChange, removeDConfigBean, removePropertyChangeListener</td>
</tr>
</tbody>
</table>

Method Detail
public DConfigBean getDConfigBean(DDBeanRoot ddBeanRoot)
DConfigBeanWeb DDBeanRoot
Web Web Web
    ddBeanRoot
    return DConfigBean DConfigBean null

getDConfigBean

DConfigBean getDConfigBean(DDBeanRoot ddBeanRoot)

Return a DConfigBean for a deployment descriptor that is not the module's primary deployment descriptor. Web services provides a deployment descriptor in addition to the module's primary deployment descriptor. Only the DDBeanRoot for this category of secondary deployment descriptors are to be passed as arguments through this method. Web service has two deployment descriptor files, one that defines the web service and one that defines a client of a web service. See the Web Service specificiation for the details.

Parameters:
    ddBeanRoot - represents the root element of a deployment descriptor file.

Returns:
    a DConfigBean to be used for processing this deployment descriptor data. Null may be returned if no DConfigBean is required for this deployment descriptor.
PS:
javax.enterprise.deploy.shared Class DConfigBeanVersionType

java.lang.Object

javax.enterprise.deploy.shared.DConfigBeanVersionType

public class DConfigBeanVersionType

extends Object

DConfigBeanVersionTypes J2EE

Class DConfigBeanVersionTypes defines enumeration values for the J2EE Platform version number.

Author:

rsearls

Field Summary

<table>
<thead>
<tr>
<th>static DConfigBeanVersionType</th>
<th>V1_3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>J2EE Platform version 1.3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static DConfigBeanVersionType</th>
<th>V1_3_1</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>J2EE Platform version 1.3.1 THIS CONSTANT SHOULD NEVER BE USED.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static DConfigBeanVersionType</th>
<th>V1_4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>J2EE Platform version 1.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static DConfigBeanVersionType</th>
<th>V5</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Java EE Platform version 5</td>
</tr>
</tbody>
</table>

Constructor Summary

DConfigBeanVersionType(int value)
Construct a new enumeration value with the given integer value.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static DConfigBeanVersionType getDConfigBeanVersionType(int value)</code></td>
<td>Return an object of the specified value.</td>
</tr>
<tr>
<td><code>protected DConfigBeanVersionType[] getEnumValueTable()</code></td>
<td>Returns the enumeration value table for class DConfigBeanVersionType.</td>
</tr>
<tr>
<td><code>protected int getOffset()</code></td>
<td>Returns the lowest integer value used by this enumeration value's enumeration class.</td>
</tr>
<tr>
<td><code>protected String[] getStringTable()</code></td>
<td>Returns the string table for class DConfigBeanVersionType.</td>
</tr>
<tr>
<td><code>int getValue()</code></td>
<td>Returns this enumeration value's integer value.</td>
</tr>
<tr>
<td><code>String toString()</code></td>
<td>Return the string name of this DConfigBeanVersionType or the integer value if outside the bounds of the table.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

**Field Detail**

V1_3

public static final DConfigBeanVersionType V1_3
J2EE Platform version 1.3

V1_3_1

public static final DConfigBeanVersionType V1_3_1

J2EE Platform version 1.3.1 THIS CONSTANT SHOULD NEVER BE USED. Use V1_3 instead.

V1_4

public static final DConfigBeanVersionType V1_4

J2EE Platform version 1.4

V5

public static final DConfigBeanVersionType V5

Java EE Platform version 5

Constructor Detail

protected DConfigBeanVersionType(int value)

value

DConfigBeanVersionType
protected `DConfigBeanVersionType(int value)`

Construct a new enumeration value with the given integer value.

**Parameters:**
value - Integer value.

**Method Detail**

```java
public int getValue()
```

```java
    return getValue()
```

**getValue**

```java
public int getValue()
```

Returns this enumeration value's integer value.

**Returns:**
the value

```java
protected String[] getStringTable()
```

`DConfigBeanVersionType`

**getStringTable**

```java
protected String[] getStringTable()
```

Returns the string table for class `DConfigBeanVersionType`

```java
protected `DConfigBeanVersionType[]` getEnumValueTable()
```

`DConfigBeanVersionType`
getEnumValueTable

protected DConfigBeanVersionType[] getEnumValueTable()

Returns the enumeration value table for class DConfigBeanVersionType

public static DConfigBeanVersionType getDConfigBeanVersionType(int value)

value

getDConfigBeanVersionType

public static DConfigBeanVersionType getDConfigBeanVersionType(int value)

Return an object of the specified value.

Parameters:
value - a designator for the object.

public String toString()
DConfigBeanVersionType

toString

public String toString()

Return the string name of this DConfigBeanVersionType or the integer value if outside the bounds of the table

Overrides:
protected int getOffset()

0

return

getOffset

protected int getOffset()

Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0.

Returns:
the offset of the lowest enumeration value.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | const | METHOD

FRAMES  NO FRAMES
DETAIL: FIELD | const | METHOD
javax.enterprise.deploy.spi.exceptions Class
DConfigBeanVersionUnsupportedException

java.lang.Object
    ^ java.lang.Throwable
    | java.lang.Exception
    | java.enterprise.deploy.spi.exceptions.DConfigBeanVersionUnsupportedException

All Implemented Interfaces:
    Serializable

public class DConfigBeanVersionUnsupportedException
extends Exception

Extends: Throwable > Exception

DConfigBean

This exception is to report that there is no support for the DConfigBean version requested.

See Also:
    Serialized Form

Constructor Summary

DConfigBeanVersionUnsupportedException(String s)
    Creates a new DConfigBeanVersionUnsupportedException object.

Method Summary

Methods inherited from class java.lang.Throwable
Constructor Detail

public DConfigBeanVersionUnsupportedException(String s)
DConfigBeanVersionUnsupportedException
   s

DConfigBeanVersionUnsupportedException

public DConfigBeanVersionUnsupportedException(String s)

   Creates an new DConfigBeanVersionUnsupportedException object.

   Parameters:
   s - a string indicating what was wrong with the version request.
javax.enterprise.deploy.model

**Interface DDBean**

All Known Subinterfaces:
- DDBeanRoot

```java
public interface DDBean
```

**Implemented by:** DDBeanRoot

Bean Bean J2EE

An interface for beans that represent a fragment of a standard deployment descriptor. A link is provided to the J2EE application that includes this bean.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addXpathListener</strong></td>
<td>Register a listener for a specific XPath.</td>
</tr>
<tr>
<td><strong>getAttributeNames</strong></td>
<td>Returns the list of attribute names associated with the XML element.</td>
</tr>
<tr>
<td><strong>getAttributeValue</strong></td>
<td>Returns the string value of the named attribute.</td>
</tr>
<tr>
<td><strong>getChildBean</strong></td>
<td>Return a list of DDBeans based upon the XPath.</td>
</tr>
<tr>
<td><strong>getId</strong></td>
<td>Returns a tool-specific reference for attribute ID on an element in the deployment descriptor.</td>
</tr>
<tr>
<td><strong>getRoot</strong></td>
<td>Return the root element for this DDBean.</td>
</tr>
<tr>
<td><strong>getText</strong></td>
<td>Returns the XML text for by this bean.</td>
</tr>
</tbody>
</table>
**Method Detail**

**public String getXpath()**

DConfigBean xpath

    return Bean XPath

**getXpath()**

String get xpath()

    Returns the original xpath string provided by the DConfigBean.

**Returns:**
The XPath of this Bean.

**public String getText()**

Bean XML

    return Bean XML

**getText()**

String get Text()
Returns the XML text for by this bean.

**Returns:**

The XML text for this Bean.

---

```java
public String getId()
```

**ID J2EE 1.2  J2EE 1.3**

```java
return Bean XML 'null'
```

**getId**

```java
String getId()
```

Returns a tool-specific reference for attribute ID on an element in the deployment descriptor. This attribute is defined for J2EE 1.2 and 1.3 components.

**Returns:**

The XML text for this Bean or 'null' if no attribute was specified with the tag.

---

```java
public DDBeanRoot getRoot()
```

**DDBean**

```java
return DDBean DDBeanRoot
```

**getRoot**

```java
DDBeanRoot getRoot()
```

Return the root element for this DDBean.

**Returns:**

The DDBeanRoot at the root of this DDBean tree.
public DDBean[] getChildBean(String xpath)

XPath DDBeans

    xpath    Bean XPath
    return    DDBeans XML 'null'

getChildBean

DDBean[] getChildBean(String xpath)

Return a list of DDBeans based upon the XPath.

Parameters:
    xpath - An XPath string referring to a location in the same
            deployment descriptor as this standard bean.

Returns:
    a list of DDBeans or 'null' if no matching XML data is found.

public String[] getText(String xpath)

XPath

    xpath    XPath
    return    XPath XML 'null'

getText

String[] getText(String xpath)

Return a list of text values for a given XPath in the deployment
descriptor.

Parameters:
    xpath - An XPath.

Returns:
    The list text values for this XPath or 'null' if no matching XML
data is found.
public void addXpathListener(String xpath, XpathListener xpl)

```
addXpathListener
```

Parameters:
- `xpath` - The XPath this listener is to be registered for.
- `xpl` - The listener object.

public void removeXpathListener(String xpath, XpathListener xpl)

```
removeXpathListener
```

Parameters:
- `xpath` - The XPath from which this listener is to be unregistered.
- `xpl` - The listener object.
public String[] getAttributeNames()

XML

    return null

getAttributeNames

String[] getAttributeNames()

    Returns the list of attribute names associated with the XML element.

    Returns:
    a list of attribute names on this element. Null is returned if there are no attributes.

public String getAttributeValue(String attrName)

    return null

getAttributeValue

String getAttributeValue(String attrName)

    Returns the string value of the named attribute.

    Returns:
    a the value of the attribute. Null is returned if there is no such attribute.
to **license terms.**

PS:
javaxenterprise.deploy.model.exceptions Class

DDBeanCreateException

java.lang.Object
  java.lang.Throwable
    java.lang.Exception
      javaxenterprise.deploy.model.exceptions.DDBeanCreateException

All Implemented Interfaces:
  Serializable

public class DDBeanCreateException extends Exception

Extends: Throwable > Exception

DDBean

This exception reports errors in generating a DDBean.

Author: gfink
See Also: Serialized Form

Constructor Summary

DDBeanCreateException()
Creates new DDBeanCreateException without detail message.

DDBeanCreateException(String msg)
Constructs an DDBeanCreateException with the specified detail message.
Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public DDBeanCreateException()

DDBeanCreateException

public DDBeanCreateException()

Creates new DDBeanCreateException without detail message.

public DDBeanCreateException(String msg)

DDBeanCreateException

msg

DDBeanCreateException

public DDBeanCreateException(String msg)

Constructs an DDBeanCreateException with the specified detail message.
Parameters:
msg - the detail message.
javax.enterprise.deploy.model Interface DDBeanRoot

All Superinterfaces:
   DDBean

public interface DDBeanRoot
extends DDBean

Implements: DDBean

DDBeanRoot  DDBean

An interface that represents the root of a standard deployment descriptor. A DDBeanRoot is a type of DDBean.

Author:
gfink

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
</tr>
<tr>
<td>getDDBeanRootVersion()</td>
</tr>
<tr>
<td>Returns the version number of an XML instance document.</td>
</tr>
<tr>
<td>DeployableObject</td>
</tr>
<tr>
<td>getDeployableObject()</td>
</tr>
<tr>
<td>Return the containing DeployableObject</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>getFilename()</td>
</tr>
<tr>
<td>Returns the filename relative to the root of the module of the XML instance document this DDBeanRoot represents.</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>getModuleDTDVersion()</td>
</tr>
<tr>
<td>Deprecated. As of version 1.1 replaced by DDBeanRoot.getDDBeanRootVersion()</td>
</tr>
<tr>
<td>getType()</td>
</tr>
</tbody>
</table>
ModuleType getType()
ModuleType getType()

Return the ModuleType of deployment descriptor.

Returns:
The ModuleType of deployment descriptor

String getXpath()

Return the XPath for this standard bean.

Methods inherited from interface javax.enterprise.deploy.model.DDBean
addXpathListener, getAttributeNames, getAttributeValue, getChildBean, getId, getRoot, getText, getText, removeXpathListener

Method Detail

deployableObject

public DeployableObject getDeployableObject()
DeployableObject

getDeployableObject

Returns:

Return the containing DeployableObject

**Returns:**
The DeployableObject that contains this deployment descriptor

```java
public String getModuleDTDVersion()

DTD DeployableObject

    @return DTD
    DD DTD XML Schema
    DDBeanRoot.getDDBeanRootVersion
    
    @deprecated 1.1 DDBeanRoot.getDDBeanRootVersion()
```

### getModuleDTDVersion

```java
String getModuleDTDVersion()
```

**Deprecated. As of version 1.1 replaced by**

`DDBeanRoot.getDDBeanRootVersion()`

A convenience method to return the DTD version number. The DeployableObject has this information.

**Returns:**
a string containing the DTD version number This method is being deprecated. Two DD data formats are being used, DTD and XML Schema. DDBeanRoot.getDDBeanRootVersion should be used in its place.

```java
public String getDDBeanRootVersion()

XML DDBeanRoot.getModuleDTDVersion

DeployableObject.getModuleDTDVersion J2EE XML

    @return XML null
```
getDDBeanRootVersion

`String getDDBeanRootVersion()`

Returns the version number of an XML instance document. This method is replacing the methods `DDBeanRoot.getModuleDTDVersion` and `DeployableObject.getModuleDTDVersion`. This method returns the version number of any J2EE XML instance document.

**Returns:**
a string that is the version number of the XML instance document. Null is returned if no version number can be found.

---

public `String getXpath()`

**Bean** XPath XPath "/"

`return"/"` Bean

getxpath

`String getXpath()`

Return the XPath for this standard bean. The root XPath is "/".

**Specified by:**
`getXpath` in interface `DDBean`

**Returns:**
"/" this is the root standard bean.

---

public `String getFilename()`

**DDBeanRoot** XML

`return(String)`

getFilename
String  getFilename()

Returns the filename relative to the root of the module of the XML instance document this DDBeanRoot represents.

**Returns:**

String the filename relative to the root of the module
javax.xml.registry  Interface DeclarativeQueryManager

All Superinterfaces:  
QueryManager

public interface DeclarativeQueryManager
extends QueryManager

Implements: QueryManager

SQL

This interface provides the ability to execute declarative queries (e.g. SQL)

Author: 
Farrukh S. Najmi

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query</td>
<td><code>createQuery(int queryType, String queryString)</code></td>
</tr>
<tr>
<td></td>
<td>Creates a Query object given a queryType (for example, QUERY_TYPE_SQL) and a String that represents a query in the syntax appropriate for queryType.</td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>executeQuery(Query query)</code></td>
</tr>
<tr>
<td></td>
<td>Executes a query as specified by query parameter.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.registry.QueryManager
getRegistryObject, getRegistryObject, getRegistryObjects, getRegistryObjects, getRegistryObjects, getRegistryService
public Query createQuery(int queryType, String queryString) throws InvalidRequestException, JAXRException

queryType QUERY_TYPE_SQL queryType
String Query

Throws:

throws JAXRException: JAXR

Throws InvalidRequestException: JAXR sqlQuery

See also:
QUERY_TYPE_SQL, QUERY_TYPE_XQUERY, QUERY_TYPE_EBXML_FILTER_QUERY

createQuery

Query createQuery(int queryType,
String queryString)
throws InvalidRequestException,
JAXRException

Creates a Query object given a queryType (for example, QUERY_TYPE_SQL) and a String that represents a query in the syntax appropriate for queryType.

Capability Level: 1

Parameters:

queryType - the type of query
queryString - the query in its string representation

Throws:

JAXRException - If the JAXR provider encounters an internal error
InvalidRequestException - If the JAXR provider validates query syntax (optional) and the sqlQuery is not valid

See Also:
Query.QUERY_TYPE_SQL, Query.QUERY_TYPE_XQUERY,
Query.QUERY_TYPE_EBXML_FILTER_QUERY

```java
default public BulkResponse executeQuery(Query query) throws JAXRException
```

query

```java
query
return BulkResponse
Throws JAXRException: JAXR
```

executeQuery

```java
BulkResponse executeQuery(Query query) throws JAXRException
```

Executes a query as specified by query parameter.

**Capability Level: 1**

**Parameters:**
query - the query to be executed

**Returns:**
the BulkResponse that is the result of the query

**Throws:**
JAXRException - If the JAXR provider encounters an internal error
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.annotation.security  Annotation Type DeclareRoles

@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface DeclareRoles

**Implements:** Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)

### since

1.0

Used by application to declare roles. It can be specified on a class.

**Since:**

1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>String[]</th>
<th>value</th>
</tr>
</thead>
</table>

### Element Detail

abstract public String[] value()

**value**

public abstract String[] value
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.helpers Class DefaultValidationEventHandler

java.lang.Object
   ↓ javax.xml.bind.helpers.DefaultValidationEventHandler

All Implemented Interfaces:
   ValidationEventHandler

public class DefaultValidationEventHandler
extends Object
implements ValidationEventHandler

Implements: ValidationEventHandler

JAXB 1.0  JAXB 1.0  JAXBContext

JAXB 2.0  JAXB
javax.xml.bind.Marshaller

version
since
See
also
$Revision: 1.5 $
JAXB1.0
javax.xml.bind.Unmarshaller, javax.xml.bind.Validator,
javax.xml.bind.ValidationEventHandler

JAXB 1.0 only default validation event handler. This is the default handler for all objects created from a JAXBContext that is managing schema-derived code generated by a JAXB 1.0 binding compiler.

This handler causes the unmarshal and validate operations to fail on the first error or fatal error.
This handler is not the default handler for JAXB mapped classes following JAXB 2.0 or later versions. Default validation event handling has changed and is specified in Unmarshaller and Marshaller.

Since:
   JAXB1.0
Version:
   $Revision: 1.5$
Author:
   - Ryan Shoemaker, Sun Microsystems, Inc.
See Also:
   Unmarshaller, Validator, ValidationEventHandler

Constructor Summary

DefaultValidationEventHandler()

Method Summary

| boolean | handleEvent(ValidationEvent event) |
Receive notification of a validation warning or error.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public DefaultValidationEventHandler()
public boolean handleEvent(ValidationEvent event)

Description copied from interface: ValidationEventHandler
Receive notification of a validation warning or error. The ValidationEvent will have a ValidationEventLocator embedded in it that indicates where the error or warning occurred.

If an unchecked runtime exception is thrown from this method, the JAXB provider will treat it as if the method returned false and interrupt the current unmarshal, validate, or marshal operation.

Specified by:
handleEvent in interface ValidationEventHandler

Parameters:
event - the encapsulated validation event information. It is a provider error if this parameter is null.

Returns:
true if the JAXB Provider should attempt to continue the current unmarshal, validate, or marshal operation after handling this warning/error, false if the provider should terminate the current operation with the appropriate UnmarshalException, ValidationException, Or MarshalException.
PS:
javax.xml.registry  Class DeleteException


All Implemented Interfaces: Serializable, JAXRResponse

public class DeleteException
extends RegistryException

Extends: Throwable > Exception > JAXRException > RegistryException

delete  RegistryException

A RegistryException that occurs during a delete action.

Author:
   Farrukh S. Najmi
See Also:
   Serialized Form

Field Summary

| Fields inherited from class javax.xml.registry.JAXRException |
| cause |

| Fields inherited from interface javax.xml.registry.JAXRResponse |
| STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING |
## Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeleteException()</strong></td>
<td>Constructs a JAXRException object with no reason or embedded Throwable.</td>
</tr>
<tr>
<td><strong>DeleteException(String reason)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><strong>DeleteException(String reason, Throwable cause)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.</td>
</tr>
<tr>
<td><strong>DeleteException(Throwable cause)</strong></td>
<td>Constructs a JAXRException object initialized with the given Throwable object.</td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class javax.xml.registry.RegistryException
- `getErrorObjectKey`, `setErrorObjectKey`

### Methods inherited from class javax.xml.registry.JAXRException
- `getCause`, `getMessage`, `getRequestId`, `getStatus`, `initCause`, `isAvailable`

### Methods inherited from class java.lang.Throwable
- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public DeleteException()
    Throwable  JAXRException

DeleteException

public DeleteException()
    Constructs a JAXRException object with no reason or embedded Throwable.

public DeleteException(String reason)
    JAXRException  String
    reason

DeleteException

public DeleteException(String reason)
    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

Parameters:
    reason - a description of what caused the exception

public DeleteException(String reason, Throwable cause)
    JAXRException  String  Throwable
    reason
    cause
DeleteException

public DeleteException(String reason, Throwable cause)

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
reason - a description of what caused the exception
cause - a Throwable object that is to be embedded in this JAXRException object

public DeleteException(Throwable cause)

Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
cause - the Throwable that caused this Exception
PS:
The delivery modes supported by the JMS API are **PERSISTENT** and **NON_PERSISTENT**.

A client marks a message as persistent if it feels that the application will have problems if the message is lost in transit. A client marks a message as non-persistent if an occasional lost message is tolerable. Clients use delivery mode to tell a JMS provider how to balance message transport reliability with throughput.

Delivery mode covers only the transport of the message to its destination. Retention of a message at the destination until its receipt is acknowledged is not guaranteed by a **PERSISTENT** delivery mode. Clients should assume that message retention policies are set administratively. Message retention policy governs the reliability of message delivery from destination to message consumer. For example, if a client's message storage space is exhausted, some messages may be dropped in accordance with a site-specific message retention policy.

A message is guaranteed to be delivered once and only once by a JMS provider if the delivery mode of the message is **PERSISTENT** and if the destination has a sufficient message retention policy.
Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>NON_PERSISTENT</td>
<td>This is the lowest-overhead delivery mode because it does not require that the message be logged to stable storage.</td>
</tr>
<tr>
<td>static int</td>
<td>PERSISTENT</td>
<td>This delivery mode instructs the JMS provider to log the message to stable storage as part of the client's send operation.</td>
</tr>
</tbody>
</table>

Field Detail

NON_PERSISTENT

static final int NON_PERSISTENT

This is the lowest-overhead delivery mode because it does not require that the message be logged to stable storage. The level of JMS provider failure that causes a NON_PERSISTENT message to be lost is not defined.

A JMS provider must deliver a NON_PERSISTENT message with an at-most-once guarantee. This means that it may lose the message, but it must not deliver it twice.

See Also:

Constant Field Values
static final int PERSISTENT

This delivery mode instructs the JMS provider to log the message to stable storage as part of the client's send operation. Only a hard media failure should cause a PERSISTENT message to be lost.

See Also:
Constant Field Values
@Documented
@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface DenyAll

**Implements:** Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=METHOD)

J2EE

**since** 1.0

**See** [javax.annotation.security.RolesAllowed](javax.annotation.security.RolesAllowed), [javax.annotation.security.PermitAll](javax.annotation.security.PermitAll)

Specifies that no security roles are allowed to invoke the specified method(s) - i.e that the methods are to be excluded from execution in the J2EE container.

**Since:**

1.0

**See Also:**

RolesAllowed, PermitAll

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.oracle.com/javase/7/docs/api/index.html).
PS:
javax.enterprise.deploy.model Interface DeployableObject

All Known Subinterfaces:
   J2eeApplicationObject

public interface DeployableObject

Implemented by: J2eeApplicationObject

DeployableObject J2EE JARWAR RAR EAR
DeployableObject

version 0.1

The DeployableObject interface is an abstract representation of a J2EE deployable module (JAR, WAR, RAR, EAR). A DeployableObject provides access to the module's deployment descriptor and class files.

Version:
   0.1
Author:
   gfink

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enumeration</strong> entries()</td>
</tr>
<tr>
<td><strong>DDBean[]</strong> getChildBean(String xpath)</td>
</tr>
<tr>
<td><strong>Class</strong> getClassFromScope(String className)</td>
</tr>
<tr>
<td><strong>DDBeanRoot</strong> getDDBeanRoot()</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td><code>getDDBeanRoot(String filename)</code></td>
</tr>
<tr>
<td><code>getEntry(String name)</code></td>
</tr>
<tr>
<td><code>getModuleDTDVersion()</code></td>
</tr>
<tr>
<td><code>getText(String xpath)</code></td>
</tr>
<tr>
<td><code>getType()</code></td>
</tr>
</tbody>
</table>

**Method Detail**

```java
definition
public ModuleType getType()

ModuleType EARJARWARRAR

DeploymentManager

```

### getType

```java
definition
public ModuleType getType()

```

Return the ModuleType of deployment descriptor (i.e., EAR, JAR, WAR, RAR) this deployable object represents. Values are found in DeploymentManager.

**Returns:**

The ModuleType of deployable object
public DDBeanRoot getDDBeanRoot()

return Bean

getDDBeanRoot

DDBeanRoot getDDBeanRoot()

Return the top level standard bean representing the root of the deployment descriptor.

Returns:
A standard bean representing the deployment descriptor.

gublic DDBean[] getChildBean(String xpath)

Bean Bean XPath XML

xpath XPath

return DDBean 'null'

getChildBean

DDBean[] getChildBean(String xpath)

Return an array of standard beans representing the XML content returned based upon the XPath.

Parameters:
xpath - An XPath string identifying the data to be extracted from the deployment descriptor.

Returns:
a array of DDBeans or 'null' if no matching data found.
public String[] getText(String xpath)

XPath XML
xpath xpath
return XML 'null'

ggetText

String[] getText(String xpath)

Return the XML content associated with the XPath from a deployment descriptor.

Parameters:
 xpath - An xpath string referring to a location in the deployment descriptor

Returns:
 a list XML content or 'null' if no matching data found.

public Class<T> getClassFromScope(String className)

EJB cross-path

className
return (Class)

gGetClassFromScope

Class getClassFromScope(String className)

Retrieve the specified class from this deployable module.

One use: to get all finder methods from an EJB If the tool is attempting to package an module and retrieve a class from the package, the class request may fail. The class may not yet be
available. The tool should respect the manifest cross-path entries.

**Parameters:**
- className - Class to retrieve.

**Returns:**
- Class representation of the class

```java
public String getModuleDTDVersion()
XML DOCTYPE DTD J2EE XML DOCTYPE

DTD

DOCTYPE

DOCTYPE

<!DOCTYPE root_element PUBLIC "-//organization//label//language" "location">

root_element - DTD
organization - DTD

return

label -

language - ISO 639 id
DTD
location - DTD URL

J2EE DOCTYPE

<!DOCTYPE application-client PUBLIC "-//Sun Microsystems, Inc.//DTD J2EE Application Client 1.3/,
"http://java.sun.com/dtd/application-client_1_3.dtd">

"DTD J2EE Application Client 1.3"
DTD 1.3 getModuleDTDVersion
"1.3"

J2EE 1.4 DDBeanRoot.getDDBeanRootVersion```
getModuleDTDVersion

String getModuleDTDVersion()

Deprecated. As of version 1.1 replaced by DDBeanRoot.getDDBeanRootVersion()

Returns the DTD version number given in the XML DOCTYPE text provided in every standard J2EE module's deployment descriptor file.

Returns:

a string containing the DTD version number

A module's deployment descriptor file always contains a document type identifier, DOCTYPE. The DOCTYPE statement contains the module DTD version number in the label of the statement.

The format of the DOCTYPE statement is:

```xml
<!DOCTYPE root_element PUBLIC "-//organization//label//language" "location">
```

root_element - is the name of the root document in the DTD.
organization - is the name of the organization responsible for the creation and maintenance of the DTD being referenced.
label - is a unique descriptive name for the public text being referenced.
language - is the ISO 639 language id representing the natural language encoding of the DTD.
location - is the URL of the DTD.

An example J2EE deployment descriptor DOCTYPE statement is:

```xml
<!DOCTYPE application-client PUBLIC "-//Sun Microsystems, Inc./DTD J2EE Application Client 1_3.dtd" "http://java.sun.com/dtd/application-client_1_3.dtd">
```

In this example the label is, "DTD J2EE Application Client" and the DTD version number is 1.3. A call to getModuleDTDVersion...
would return a string containing, "1.3".

This method is being deprecated. With the addition of multiple deployment descriptors in components for J2EE 1.4 this method is being replaced by DDBeanRoot.getDDBeanRootVersion.

```java
public DDBeanRoot getDDBeanRoot(String filename)
throws java.io.FileNotFoundException,
DDBeanCreateException
XML DDBeanRoot XML WSDL
DDBeanRoot
  return XML DDBeanRoot
  Throws java.io.FileNotFoundException,:
  Throws DDBeanCreateException: DDBeanRoot
```

getDDBeanRoot

```java
DDBeanRoot getDDBeanRoot(String filename)
throws FileNotFoundException,
DDBeanCreateException
```

Returns a DDBeanRoot object for the XML instance document named. This method should be used to return DDBeanRoot objects for non deployment descriptor XML instance documents such as WSDL files.

**Returns:**
- a DDBeanRoot object for the XML data.

**Throws:**
- java.io.FileNotFoundException, - if the named file can not be found
- DDBeanCreateException - if an error is encountered creating the DDBeanRoot object.
- FileNotFoundException
public java.util.Enumeration\langle E \rangle \ entries() \\
String \ String \\
\hspace{1cm} \text{return} \\

\textbf{entries} \\

\hspace{1cm} \textit{Enumeration entries()} \\

Returns an enumeration of the module file entries. All elements in the enumeration are of type String. Each String represents a file name relative to the root of the module.

\textbf{Returns:} \\
\hspace{1cm} an enumeration of the archive file entries.

\textbf{public java.io.InputStream \ getEntry(String name)} \\
\hspace{1cm} \textit{InputStream \ name} \\
\hspace{1cm} \text{return} \hspace{1cm} \text{InputStream null} \\

\textbf{getEntry} \\

\hspace{1cm} \textit{InputStream \ getEntry(String name)} \\

Returns the InputStream for the given entry name. The file name must be relative to the root of the module.

\textbf{Parameters:} \\
\hspace{1cm} \textit{name} - the file name relative to the root of the module. \\

\textbf{Returns:} \\
\hspace{1cm} the InputStream for the given entry name or null if not found.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface DeploymentConfiguration

J2EE DeploymentConfiguration  EAR
    version  0.1

An interface that defines a container for all the server-specific configuration information for a single top-level J2EE module. The DeploymentConfiguration object could represent a single stand alone module or an EAR file that contains several sub-modules.

**Version:**
  0.1

**Author:**
gfink

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>DConfigBeanRoot</td>
</tr>
<tr>
<td>DeployableObject</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td>InputStream</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>DConfigBeanRoot</td>
</tr>
</tbody>
</table>
| restoreDConfigBean( InputStream inputArchive,  
| DDBeanRoot bean) |
| Restore from disk to instantiated objects all the  
| DConfigBeans associated with a specific deployment descriptor. |
| void  | Save to disk the current set configuration beans created for this deployable module. |
| save( OutputStream outputArchive) |
| void  | Save to disk all the configuration beans associated with a particular deployment descriptor file. |
| saveDConfigBean( OutputStream outputArchive,  
| DConfigBeanRoot bean) |

**Method Detail**

```java
public DeployableObject getDeployableObject() {
    return DeployableObject
}
```

**getDeployableObject**

```java
DeployableObject getDeployableObject() {
    return DeployableObject
}
```

Returns an object that provides access to the deployment descriptor data and classes of a J2EE module.

**Returns:**

DeployableObject

```java
public DConfigBeanRoot getDConfigBeanRoot(DDBeanRoot bean) throws
```

```java
DConfigBeanRoot getDConfigBeanRoot(DDBeanRoot bean) throws
```
ConfigurationException
Bean DConfigBeanRoot DDBeanRoot Bean

```java
return

Throws

DConfigBeanRoot getDConfigBeanRoot(DDBeanRoot bean)
throws ConfigurationException
```

getDConfigBeanRoot

Returns the top level configuration bean, DConfigBeanRoot, associated with the deployment descriptor represented by the designated DDBeanRoot bean.

**Parameters:**
- `bean` - The top level bean that represents the associated deployment descriptor.

**Returns:**
- the DConfigBeanRoot for editing the server-specific properties required by the module.

**Throws:**
- `ConfigurationException` - reports errors in generating a configuration bean

---

```java
public void removeDConfigBean(DConfigBeanRoot bean)
throws BeanNotFoundException
DConfigBean Bean

Throws

BeanNotFoundException: Bean Bean
```

removeDConfigBean

```java
void removeDConfigBean(DConfigBeanRoot bean)
throws BeanNotFoundException
```

Throws
Remove the root DConfigBean and all its children.

**Parameters:**
bean - the top level DConfigBean to remove.

**Throws:**
BeanNotFoundException - the bean provided is not in this bean's child list.

```java
public DConfigBeanRoot restoreDConfigBean(java.io.InputStream inputArchive, DDBeanRoot bean) throwsConfigurationException
```

**DConfigBean Bean**

- `inputArchive` - DConfigBean
- `bean` - DDBeanRoot Bean
- `return` - Bean DConfigBeanRoot

**Throws**
ConfigurationException: Bean

**restoreDConfigBean**

```java
DConfigBeanRoot restoreDConfigBean(InputStream inputArchive, DDBeanRoot bean)
```

**Parameters:**
- `inputArchive` - The input stream for the file from which the DConfigBeans should be restored.
- `bean` - The DDBeanRoot bean associated with the deployment descriptor file.

**Returns:**
The top most parent configuration bean, DConfigBeanRoot

**Throws:**
ConfigurationException - reports errors in generating a
public void saveDConfigBean(java.io.OutputStream outputArchive, DConfigBeanRoot bean) throws ConfigurationException

Bean DConfigBean XML

outputArchive DConfigBean
bean Bean DConfigBeanRoot Bean
Throws ConfigurationException: Bean

saveDConfigBean

void saveDConfigBean(OutputStream outputArchive, DConfigBeanRoot bean)
throws ConfigurationException

Save to disk all the configuration beans associated with a particular deployment descriptor file. The saved data may be fully or partially configured DConfigBeans. The output file format is recommended to be XML.

Parameters:
outputArchive - The output stream to which the DConfigBeans should be saved.
bean - The top level bean, DConfigBeanRoot, from which to be save.

Throws:
ConfigurationException - reports errors in generating a configuration bean

public void restore(java.io.InputStream inputArchive)
throws ConfigurationException

Bean

inputArchive
Throws ConfigurationException: Bean
restore

```java
void restore(InputStream inputArchive)
    throws ConfigurationException
```

Restore from disk to a full set of configuration beans previously stored.

**Parameters:**
- `inputArchive` - The input stream from which to restore the Configuration.

**Throws:**
- `ConfigurationException` - reports errors in generating a configuration bean

---

save

```java
public void save(java.io.OutputStream outputArchive)
    throws ConfigurationException
```

Save to disk the current set configuration beans created for this deployable module. It is recommended the file format be XML.

**Parameters:**
- `outputArchive` - The output stream to which to save the Configuration.

**Throws:**
- `ConfigurationException`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
The DeploymentFactory interface is a deployment driver for a J2EE platform product. It returns a DeploymentManager object which represents a connection to a specific J2EE platform product.

Each application server vendor must provide an implementation of this class in order for the J2EE Deployment API to work with their product.

The class implementing this interface should have a public no-argument constructor, and it should be stateless (two instances of the class should always behave the same). It is suggested but not required that the class have a static initializer that registers an instance of the class with the DeploymentFactoryManager class.

A connected or disconnected DeploymentManager can be requested. A DeploymentManager that runs connected to the platform can provide access to J2EE resources. A DeploymentManager that runs disconnected only provides module deployment configuration support.
See Also:
DeploymentFactoryManager

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DeploymentManager getDeploymentManager(String uri, String username, String password)</code></td>
<td>Return a connected <code>DeploymentManager</code> instance.</td>
</tr>
<tr>
<td><code>DeploymentManager getDisconnectedDeploymentManager(String uri)</code></td>
<td>Return a disconnected <code>DeploymentManager</code> instance.</td>
</tr>
<tr>
<td><code>String getDisplayName()</code></td>
<td>Provide a string with the name of this vendor's <code>DeploymentManager</code>.</td>
</tr>
<tr>
<td><code>String getProductVersion()</code></td>
<td>Provide a string identifying version of this vendor's <code>DeploymentManager</code>.</td>
</tr>
<tr>
<td><code>boolean handlesURI(String uri)</code></td>
<td>Tests whether this factory can create a <code>DeploymentManager</code> object based on the specified URI.</td>
</tr>
</tbody>
</table>

## Method Detail

```java
default public boolean handlesURI(String uri) {
    // Implementation details...
    return true;
}
```
Tests whether this factory can create a DeploymentManager object based on the specified URI. This does not indicate whether such an attempt will be successful, only whether the factory can handle the uri.

**Parameters:**
- **uri** - The uri to check

**Returns:**
- true if the factory can handle the uri.

```java
public DeploymentManager getDeploymentManager(String uri, String username, String password) throws DeploymentManagerCreationException
```

```
uri
username
password
return
```

**Throws**
- DeploymentManagerCreationException

**getDeploymentManager**

```java
DeploymentManager getDeploymentManager(String uri,
String username,
String password)
```

**Returns:**

- Connected DeploymentManager instance.

**Parameters:**
- **uri** - The URI that specifies the connection parameters
- **username** - An optional username (may be null if no authentication is required for this platform).
- **password** - An optional password (may be null if no authentication is required for this platform).

**Returns:**
A ready DeploymentManager instance.

**Throws:**
- `DeploymentManagerCreationException` - occurs when a DeploymentManager could not be returned (server down, unable to authenticate, etc).

```java
public DeploymentManager getDisconnectedDeploymentManager(String uri) throws DeploymentManagerCreationException
    return DeploymentManager
Throws DeploymentManagerCreationException: DeploymentManager
```

**getDisconnectedDeploymentManager**

```java
DeploymentManager getDisconnectedDeploymentManager(String uri) throws DeploymentManagerCreationException
```

Return a disconnected DeploymentManager instance.

**Parameters:**
- `uri` - the uri of the DeploymentManager to return.

**Returns:**
- A DeploymentManager disconnected instance.

**Throws:**
- `DeploymentManagerCreationException` - occurs if the DeploymentManager could not be created.

```java
public String getDisplayName()
    return DeploymentManager
```

```java
DeploymentManager
```
getDisplayName

String getDisplayName()

Provide a string with the name of this vendor's DeploymentManager.

Returns:
the name of the vendor's DeploymentManager.

public String getProductVersion()

DeploymentManager return DeploymentManager

getProductVersion

String getProductVersion()

Provide a string identifying version of this vendor's DeploymentManager.

Returns:
the name of the vendor's DeploymentManager.
The DeploymentFactoryManager class is a central registry for J2EE DeploymentFactory objects. The DeploymentFactoryManager retains references to DeploymentFactory objects loaded by a tool. A DeploymentFactory object provides a reference to a DeploymentManager. The DeploymentFactoryManager has been implemented as a singleton. A tool gets a reference to the DeploymentFactoryManager via the getInstance method. The DeploymentFactoryManager can return two types of DeploymentManagers, a connected DeploymentManager and a disconnected DeploymentManager. The connected DeploymentManager provides access to any product resources that may be required for configurations and deployment. The method to retrieve a connected DeploymentManager is getDeploymentManager. This method provides parameters for user name and password that the product may require for
user authentication. A disconnected DeploymentManager does not provide access to a running J2EE product. The method to retrieve a disconnected DeploymentManager is getDisconnectedDeploymentManager. A disconnected DeploymentManager does not need user authentication information.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeploymentFactory[]</td>
<td>getDeploymentFactories()</td>
<td>Retrieve the lists of currently registered DeploymentFactories.</td>
</tr>
<tr>
<td>DeploymentManager</td>
<td>getDeploymentManager(String uri, String username, String password)</td>
<td>Retrieves a DeploymentManager instance for deployment.</td>
</tr>
<tr>
<td>DeploymentManager</td>
<td>getDisconnectedDeploymentManager(String uri)</td>
<td>Return a disconnected DeploymentManager instance.</td>
</tr>
<tr>
<td>static DeploymentFactoryManager</td>
<td>getInstance()</td>
<td>Retrieve the Singleton DeploymentFactoryManager</td>
</tr>
<tr>
<td>void</td>
<td>registerDeploymentFactory(DeploymentFactory factory)</td>
<td>Registers a DeploymentFactory so it will handle requests.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Method Detail

public static DeploymentFactoryManager getInstance()

DeploymentFactoryManager

return DeploymentFactoryManager
getInstance

public static DeploymentFactoryManager getInstance()

Retrieve the Singleton DeploymentFactoryManager

Returns:
DeploymentFactoryManager instance

getDeploymentFactories

DeploymentFactory[] getDeploymentFactories()

return DeploymentFactory

getDeploymentFactories

public DeploymentFactory[] getDeploymentFactories()

Retrieve the lists of currently registered DeploymentFactories.

Returns:
the list of DeploymentFactory objects or an empty array if there are none.

getDeploymentManager

DeploymentManager getDeploymentManager(String uri, String username, String password) throws DeploymentManagerCreationException

DeploymentManager URI
DeploymentFactory URI DeploymentFactory

uri
username
password

uri
null

uri
null
getDeploymentManager

```java
public DeploymentManager getDeploymentManager(String uri,
                                             String username,
                                             String password)
throws DeploymentManagerCreationException
```

Retrieves a DeploymentManager instance to use for deployment. The caller provides a URI and optional username and password, and all registered DeploymentFactories will be checked. The first one to understand the URI provided will attempt to initiate a server connection and return a ready DeploymentManager instance.

**Parameters:**
- **uri** - The uri to check
- **username** - An optional username (may be null if no authentication is required for this platform).
- **password** - An optional password (may be null if no authentication is required for this platform).

**Returns:**
A ready DeploymentManager instance.

**Throws:**
- **DeploymentManagerCreationException** - Occurs when the factory appropriate to the specified URI was unable to initialize a DeploymentManager instance (server down, unable to authenticate, etc.).

public void registerDeploymentFactory(DeploymentFactory factory)

registerDeploymentFactory
public void registerDeploymentFactory(DeploymentFactory factory)

Registers a DeploymentFactory so it will be able to handle requests.

public DeploymentManager getDisconnectedDeploymentManager(String uri) throws DeploymentManagerCreationException

    DeploymentManager uri DeploymentManager
    return DeploymentManager
    Throws DeploymentDriverException: DeploymentManager

getDisconnectedDeploymentManager

public DeploymentManager getDisconnectedDeploymentManager(String uri) throws DeploymentManagerCreationException

    Return a disconnected DeploymentManager instance.

    Parameters:
    uri - identifier of the disconnected DeploymentManager to return.

    Returns:
    A DeploymentManager instance.

    Throws:
    DeploymentDriverException - occurs if the DeploymentManager could not be created.
    DeploymentManagerCreationException

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
The DeploymentManager object provides the core set of functions a J2EE platform must provide for J2EE application deployment. It provides server related information, such as, a list of deployment targets, and vendor unique runtime configuration information.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DeploymentConfiguration createConfiguration(DeployableObject dObj)</code></td>
<td>Retrieve the object that provides server-specific configuration information for the J2EE deployable component.</td>
</tr>
<tr>
<td><code>ProgressObject distribute(Target[] targetList, File moduleArchive, File deploymentPlan)</code></td>
<td>The distribute method performs three tasks; it validates the deployment configuration data, generates all container classes and interfaces, and moves the fully baked archive to designated deployment targets.</td>
</tr>
<tr>
<td><code>ProgressObject distribute(Target[] targetList, InputStream moduleArchive, InputStream deploymentPlan)</code></td>
<td><strong>Deprecated.</strong> as of Java EE 5, replaced with <code>distribute(Target[], ModuleType, InputStream, InputStream)</code>.</td>
</tr>
<tr>
<td><code>ProgressObject distribute(Target[] targetList, ModuleType type, InputStream moduleArchive, InputStream deploymentPlan)</code></td>
<td>The distribute method performs three tasks; it validates the deployment configuration data, generates all container classes and interfaces, and moves the fully baked archive to designated deployment targets.</td>
</tr>
<tr>
<td><code>getAvailableModules(ModuleType moduleType, Target[] targetList)</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getNonRunningModules()</code></td>
<td>Retrieve the list of J2EE application modules distributed to the identified targets and that are currently not running on the associated server or servers.</td>
</tr>
<tr>
<td><code>getCurrentLocale()</code></td>
<td>Returns the active locale this implementation of <code>javax.enterprise.deploy.spi</code> subpackages is running.</td>
</tr>
<tr>
<td><code>getDConfigBeanVersion()</code></td>
<td>Returns the J2EE platform version number for which the configuration beans are provided.</td>
</tr>
<tr>
<td><code>getDefaultLocale()</code></td>
<td>Returns the default locale supported by this implementation of <code>javax.enterprise.deploy.spi</code> subpackages.</td>
</tr>
<tr>
<td><code>getRunningModules()</code></td>
<td>Retrieve the list of J2EE application modules distributed to the identified targets and that are currently running on the associated server or servers.</td>
</tr>
<tr>
<td><code>getSupportedLocales()</code></td>
<td>Returns an array of supported locales for this implementation.</td>
</tr>
<tr>
<td><code>getTargets()</code></td>
<td>Retrieve the list of deployment targets supported by DeploymentManager.</td>
</tr>
<tr>
<td><code>isDConfigBeanVersionSupported()</code></td>
<td>Returns 'true' if the configuration beans support the J2EE platform version specified.</td>
</tr>
<tr>
<td><code>isLocaleSupported()</code></td>
<td>Reports if this implementation supports the designated locale.</td>
</tr>
<tr>
<td><code>isRedeploySupported()</code></td>
<td>This method designates whether this platform vendor provides application redeployment functionality.</td>
</tr>
<tr>
<td><code>redeploy()</code></td>
<td></td>
</tr>
</tbody>
</table>
The redeploy method provides a means for updating currently deployed J2EE applications.

```java
ProgressObject redeploy(TargetModuleID[] moduleIDList, InputStream moduleArchive, InputStream deploymentPlan)
```

The release method is the mechanism by which signals to the DeploymentManager that the tool does to continue running connected to the platform.

```java
void release()
```

Set the configuration beans to be used to the J2EE platform version specified.

```java
void setDConfigBeanVersion(DConfigBeanVersionType version)
```

Set the active locale for this implementation of javax.enterprise.deploy.spi subpackages to run.

```java
void setLocale(Locale locale)
```

Start the application running.

```java
ProgressObject start(TargetModuleID[] moduleIDList)
```

Stop the application running.

```java
ProgressObject stop(TargetModuleID[] moduleIDList)
```

Remove the application from the target server.

```java
ProgressObject undeploy(TargetModuleID[] moduleIDList)
```

Method Detail

```java
public Target[] getTargets() throws IllegalStateException
```

Throws IllegalArgumentException:

```java
return Target 'null'
```

getTargets
getTargets() throws IllegalStateException

Retrieve the list of deployment targets supported by this DeploymentManager.

Returns:
A list of deployment Target designators the user may select for application deployment or 'null' if there are none.

Throws:
IllegalStateException - is thrown when the method is called when running in disconnected mode.

public TargetModuleID[] getRunningModules(ModuleType moduleType, Target[] targetList) throws TargetException, IllegalStateException

J2EE

moduleType J2EE

targetList Target

return TargetModuleID 'null'

Throws IllegalStateException:

Throws TargetException: Target

getRunningModules

TargetModuleID[] getRunningModules(ModuleType moduleType, Target[] targetList) throws TargetException, IllegalStateException

Retrieve the list of J2EE application modules distributed to the identified targets and that are currently running on the associated server or servers.

Parameters:
moduleType - A predefined designator for a J2EE module type.
targetList - A list of deployment Target designators the user
wants checked for module run status.

**Returns:**
An array of TargetModuleID objects representing the running modules or 'null' if there are none.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.
- `TargetException` - An invalid Target designator encountered.

```java
public TargetModuleID[] getNonRunningModules(ModuleType moduleType, Target[] targetList) throws TargetException, IllegalStateException
```

**J2EE**

- **moduleType**
  - J2EE
- **targetList**
  - Target
- **return**
  - TargetModuleID 'null'
- **Throws**
  - TargetException: Target

```
getNonRunningModules
```

Retrieve the list of J2EE application modules distributed to the identified targets and that are currently not running on the associated server or servers.

**Parameters:**
- `moduleType` - A predefined designator for a J2EE module type.
- `targetList` - A list of deployment Target designators the user wants checked for module not running status.

**Returns:**
An array of TargetModuleID objects representing the non-
running modules or 'null' if there are none.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.
- `TargetException` - An invalid Target designator encountered.

```java
public TargetModuleID[] getAvailableModules(ModuleType moduleType, Target[] targetList) throws TargetException, IllegalStateException
```

**J2EE**

moduleType - A predefined designator for a J2EE module type.
targetList - A list of deployment Target designators the user wants checked for module not running status.

**Returns:**
An array of TargetModuleID objects representing all deployed modules running or not or 'null' if there are no deployed modules.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.
when running in disconnected mode.  
TargetException - An invalid Target designator encountered.

```
public DeploymentConfiguration createConfiguration(DeployableObject dObj) throws InvalidModuleException

J2EE dObj J2EE

Throws InvalidModuleException: DeployableObject

createConfiguration

DeploymentConfiguration createConfiguration(DeployableObject dObj) throws InvalidModuleException

Retrieve the object that provides server-specific deployment configuration information for the J2EE deployable component.

Parameters:

dObj - An object representing a J2EE deployable component.

Throws:

InvalidModuleException - The DeployableObject is an unknown or unsupport component for this configuration tool.

public ProgressObject distribute(Target[] targetList, java.io.File moduleArchive, java.io.File deploymentPlan) throws IllegalStateException

(targetList moduleArchive deploymentPlan

Throws IllegalStateException: ProgressObject

return
The distribute method performs three tasks; it validates the deployment configuration data, generates all container specific classes and interfaces, and moves the fully baked archive to the designated deployment targets.

**Parameters:**
- `targetList` - A list of server targets the user is specifying this application be deployed to.
- `moduleArchive` - The file name of the application archive to be distributed.
- `deploymentPlan` - The XML file containing the runtime configuration information associated with this application archive.

**Returns:**
- ProgressObject an object that tracks and reports the status of the distribution process.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.

```java
public ProgressObject distribute(Target[] targetList,
                                 File moduleArchive,
                                 File deploymentPlan)
                                 throws IllegalStateException
```

deprecated

Java EE 5

```java
#distribute(Target[], ModuleType,
            InputStream, InputStream)
```
The distribute method performs three tasks; it validates the deployment configuration data, generates all container specific classes and interfaces, and moves the fully baked archive to the designated deployment targets.

**Parameters:**
- **targetList** - A list of server targets the user is specifying this application be deployed to.
- **moduleArchive** - The input stream containing the application archive to be distributed.
- **deploymentPlan** - The input stream containing the deployment configuration information associated with this application archive.

**Returns:**
ProgressObject an object that tracks and reports the status of the distribution process.

**Throws:**
- **IllegalStateException** - is thrown when the method is called when running in disconnected mode.
The distribute method performs three tasks; it validates the deployment configuration data, generates all container specific classes and interfaces, and moves the fully baked archive to the designated deployment targets.

**Parameters:**
- `targetList` - A list of server targets the user is specifying this application be deployed to.
- `moduleType` - The module type of this application archive.
- `moduleArchive` - The input stream containing the application archive to be distributed.
- `deploymentPlan` - The input stream containing the deployment configuration information associated with this application archive.

**Returns:**
- `ProgressObject` an object that tracks and reports the status of the distribution process.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.
public ProgressObject stop(TargetModuleID[] moduleIDList) throws IllegalStateException

Stop the application running.

Only the TargetModuleIDs which represent a root module are valid for being stopped. A root TargetModuleID has no parent. A TargetModuleID with a parent can not be individually stopped. A root TargetModuleID module and all its child modules will be stopped.

Parameters:
moduleIDList - A array of TargetModuleID objects representing the modules to be stopped.

Returns:
ProgressObject an object that tracks and reports the status of the stop operation.

Throws:
IllegalStateException - is thrown when the method is called when running in disconnected mode.
stop

**ProgressObject** stop(**TargetModuleID[]** moduleIDList) throws **IllegalStateException**

Stop the application running.

Only the **TargetModuleID**s which represent a root module are valid for being stopped. A root **TargetModuleID** has no parent. A **TargetModuleID** with a parent can not be individually stopped. A root **TargetModuleID** module and all its child modules will be stopped.

**Parameters:**
- **moduleIDList** - A array of **TargetModuleID** objects representing the modules to be stopped.

**Returns:**
- **ProgressObject** an object that tracks and reports the status of the stop operation.

**Throws:**
- **IllegalStateException** - is thrown when the method is called when running in disconnected mode.

---

public **ProgressObject** undeploy(**TargetModuleID[]** moduleIDList) throws **IllegalStateException**
Throws
IllegalStateException:
return ProgressObject

**undeploy**

**ProgressObject** undeploy(TargetModuleID[] moduleIDList)
throws **IllegalStateException**

Remove the application from the target server.

Only the TargetModuleIDs which represent a root module are valid for undeployment. A root TargetModuleID has no parent. A TargetModuleID with a parent can not be undeployed. A root TargetModuleID module and all its child modules will be undeployed. The root TargetModuleID module and all its child modules must stopped before they can be undeployed.

**Parameters:**
moduleIDList - An array of TargetModuleID objects representing the root modules to be stopped.

**Returns:**
ProgressObject an object that tracks and reports the status of the stop operation.

**Throws:**
**IllegalStateException** - is thrown when the method is called when running in disconnected mode.

---

**public boolean isRedeploySupported()**

true false

return true DeploymentManager false

isRedeploySupported

boolean **isRedeploySupported()**
This method designates whether this platform vendor provides application redeployment functionality. A value of true means it is supported. False means it is not.

**Returns:**
A value of true means redeployment is supported by this vendor’s DeploymentManager. False means it is not.

```java
public ProgressObject redeploy(TargetModuleID[] moduleIdList, java.io.File moduleArchive, java.io.File deploymentPlan) throws UnsupportedOperationException, IllegalStateException
```

The redeploy method provides a means for updating currently deployed J2EE applications. This is an optional method for vendor implementation. Redeploy replaces a currently deployed application with an updated version. The runtime configuration information for the updated application must remain identical to the application it is updating. When an application update is redeployed,
all existing client connections to the original running application must not be disrupted; new clients will connect to the application update. This operation is valid for TargetModuleIDs that represent a root module. A root TargetModuleID has no parent. A root TargetModuleID module and all its child modules will be redeployed. A child TargetModuleID module cannot be individually redeployed. The redeploy operation is complete only when this action for all the modules has completed.

**Parameters:**
- `moduleIDList` - An array of designators of the applications to be updated.
- `moduleArchive` - The file name of the application archive to be distributed.
- `deploymentPlan` - The deployment configuration information associated with this application archive.

**Returns:**
- `ProgressObject` an object that tracks and reports the status of the redeploy operation.

**Throws:**
- `IllegalStateException` - is thrown when the method is called when running in disconnected mode.
- `UnsupportedOperationException` - this optional command is not supported by this implementation.

```java
public ProgressObject redeploy(TargetModuleID[] moduleIDList, java.io.InputStream moduleArchive, java.io.InputStream deploymentPlan) throws UnsupportedOperationException, IllegalStateException
J2EE
TargetModuleID  TargetModuleID
TargetModuleID  TargetModuleID

moduleIDList
moduleArchive
deploymentPlan
return ProgressObject
```
redeploy

`ProgressObject` `redeploy`(`TargetModuleID[]` `moduleIDList`, `InputStream` `moduleArchive`, `InputStream` `deploymentPlan`) throws `UnsupportedOperationException`, `IllegalStateException` (optional)

The redeploy method provides a means for updating currently deployed J2EE applications. This is an optional method for vendor implementation. Redeploy replaces a currently deployed application with an updated version. The runtime configuration information for the updated application must remain identical to the application it is updating. When an application update is redeployed, all existing client connections to the original running application must not be disrupted; new clients will connect to the application update. This operation is valid for TargetModuleIDs that represent a root module. A root TargetModuleID has no parent. A root TargetModuleID module and all its child modules will be redeployed. A child TargetModuleID module cannot be individually redeployed. The redeploy operation is complete only when this action for all the modules has completed.

**Parameters:**
- `moduleIDList` - An array of designators of the applications to be updated.
- `moduleArchive` - The input stream containing the application archive to be distributed.
- `deploymentPlan` - The input stream containing the runtime configuration information associated with this application archive.

**Returns:**
- `ProgressObject` an object that tracks and reports the status of the redeploy operation.

**Throws:**
- `IllegalStateException` - is thrown when the method is called
when running in disconnected mode.

UnsupportedOperationException - this optional command is not supported by this implementation.

```java
public void release()
release DeploymentManager
DeploymentManager J2EE
ProgressObject ProgressObject
```

The release method is the mechanism by which the tool signals to the DeploymentManager that the tool does not need it to continue running connected to the platform. The tool may be signaling it wants to run in a disconnected mode or it is planning to shutdown. When release is called the DeploymentManager may close any J2EE resource connections it had for deployment configuration and perform other related resource cleanup. It should not accept any new operation requests (i.e., distribute, start stop, undeploy, redeploy. It should finish any operations that are currently in process. Each ProgressObject associated with a running operation should be marked as released (see the ProgressObject).

```java
public java.util.Locale getDefaultLocale()
javax.enterprise.deploy.spi
return (Locale)
```

defaultLocale

```java
Locale getDefaultLocale()
```
Returns the default locale supported by this implementation of javax.enterprise.deploy.spi subpackages.

**Returns:**
Locale the default locale for this implementation.

```java
public java.util.Locale getCurrentLocale()
javax.enterprise.deploy.spi
return (Locale)
```

**getCurrentLocale**

Locale getCurrentLocale()

Returns the active locale this implementation of javax.enterprise.deploy.spi subpackages is running.

**Returns:**
Locale the active locale of this implementation.

```java
public void setLocale(java.util.Locale locale) throws UnsupportedOperationException
javax.enterprise.deploy.spi
Throws UnsupportedOperationException:
```

**setLocale**

void setLocale(Locale locale)
throws UnsupportedOperationException

Set the active locale for this implementation of javax.enterprise.deploy.spi subpackages to run.

**Throws:**
UnsupportedOperationException - the provide locale is not supported.

public java.util.Locale[] getSupportedLocales()

    return (Locale[])

getSupportedLocales

Locale[] getSupportedLocales()

    Returns an array of supported locales for this implementation.

    Returns:
    Locale[] the list of supported locales.

public boolean isLocaleSupported(java.util.Locale locale)

    return 'true' 'false'

isLocaleSupported

boolean isLocaleSupported(Locale locale)

    Reports if this implementation supports the designated locale.

    Returns:
    A value of 'true' means it is support and 'false' it is not.

public DConfigBeanVersionType getDConfigBeanVersion()

Bean J2EE Bean J2EE J2SE

    return Bean DConfigBeanVersionType
getDConfigBeanVersion

DConfigBeanVersionType getDConfigBeanVersion()

Returns the J2EE platform version number for which the configuration beans are provided. The beans must have been compiled with the J2SE version required by the J2EE platform.

Returns:
a DConfigBeanVersionType object representing the platform version number for which these beans are provided.

public boolean isDConfigBeanVersionSupported(DConfigBeanVersionType version)

Bean J2EE 'true' 'false'

version J2EE DConfigBeanVersionType
return 'true' 'false'

isDConfigBeanVersionSupported

boolean isDConfigBeanVersionSupported(DConfigBeanVersionType version)

Returns 'true' if the configuration beans support the J2EE platform version specified. It returns 'false' if the version is not supported.

Parameters:
version - a DConfigBeanVersionType object representing the J2EE platform version for which support is requested.

Returns:
'true' if the version is supported and 'false if not.
setDConfigBeanVersion(DConfigBeanVersionType version) throws DConfigBeanVersionUnsupportedException

J2EE Bean

    version    J2EE DConfigBeanVersionType
Throws DConfigBeanVersionUnsupportedException: Bean

setDConfigBeanVersion

void setDConfigBeanVersion(DConfigBeanVersionType version) throws DConfigBeanVersionUnsupportedException

Set the configuration beans to be used to the J2EE platform version specified.

Parameters:
    version - a DConfigBeanVersionType object representing the J2EE platform version for which support is requested.

Throws:
    DConfigBeanVersionUnsupportedException - when the requested bean version is not supported.
javax.enterprise.deploy.spi.exceptions Class DeploymentManagerCreationException

java.lang.Object
  ↓ java.lang.Throwable
  ↓ java.lang.Exception
  ↓ javax.enterprise.deploy.spi.exceptions.DeploymentManagerCreationException

All Implemented Interfaces:
  Serializable

public class DeploymentManagerCreationException
  extends Exception

Extends: Throwable > Exception

DeploymentManager

This exception is to report problems in returning a DeploymentManager object cause by such things as server down, unable to authenticate and the like.

See Also:
 Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DeploymentManagerCreationException(String s)</code></td>
<td>Creates an new DeploymentManagerCreationException object.</td>
</tr>
</tbody>
</table>

---

**Method Summary**

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
<th></th>
</tr>
</thead>
</table>
Constructor Detail

public DeploymentManagerCreationException(String s)
DeploymentManagerCreationException

s

DeploymentManagerCreationException

public DeploymentManagerCreationException(String s)

Creates an new DeploymentManagerCreationException object.

Parameters:

s - a string providing more information about the problem.
public interface DeploymentStatus

DeploymentStatus

The DeploymentStatus interface provides information about the progress status of a deployment action.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionType</td>
<td><strong>getAction()</strong> Retrieve the deployment ActionType for this event.</td>
</tr>
<tr>
<td>CommandType</td>
<td><strong>getCommand()</strong> Retrieve the deployment CommandType of this event.</td>
</tr>
<tr>
<td>String</td>
<td><strong>getMessage()</strong> Retrieve any additional information about the status of this event.</td>
</tr>
<tr>
<td>StateType</td>
<td><strong>getState()</strong> Retrieve the StateType value.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isCompleted()</strong> A convenience method to report if the operation is in the completed state.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isFailed()</strong> A convenience method to report if the operation is in the failed state.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isRunning()</strong> A convenience method to report if the operation is in the running state.</td>
</tr>
</tbody>
</table>
Method Detail

public **StateType** getState()

```java
StateType
```

return StateType

**getState**

**StateType** getState()

Retrieve the StateType value.

**Returns:**
the StateType object

---

public **CommandType** getCommand()

```java
CommandType
```

return CommandType

**getCommand**

**CommandType** getCommand()

Retrieve the deployment CommandType of this event.

**Returns:**
the CommandType Object

---

public **ActionType** getAction()

```java
ActionType
```

return ActionType

**getAction**

**ActionType** getAction()
**getAction**

*ActionType* `getAction()`

Retrieve the deployment *ActionType* for this event.

**Returns:**
- the *ActionType* Object

---

**public String getMessage()**

`return`

---

**getmessage**

*String* `getmessage()`

Retrieve any additional information about the status of this event.

**Returns:**
- message text

---

**public boolean isCompleted()**

`return true`

---

**isCompleted**

*boolean* `isCompleted()`

A convenience method to report if the operation is in the completed state.

**Returns:**
true if this command has completed successfully

public boolean isFailed()

    return true

isFailed

boolean isFailed()

    A convenience method to report if the operation is in the failed state.

    Returns:
    true if this command has failed

public boolean isRunning()

    return true

isRunning

boolean isRunning()

    A convenience method to report if the operation is in the running state.

    Returns:
    true if this command is still running

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface DeserializationContext

javax.xml.rpc.encoding.DeserializationContext JAX-RPC XML Deserializer

version 1.0

The javax.xml.rpc.encoding.DeserializationContext interface is implemented by the JAX-RPC runtime system in an XML processing mechanism specific manner. A deserializer uses this interface to access and maintain context information during the deserialization..

Version:
1.0
Author:
Rahul Sharma

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.rpc.encoding Interface Deserializer

All Superinterfaces:
Serializable

public interface Deserializer
extends Serializable

Implements: java.io.Serializable

javax.xml.rpc.encoding.Deserializer  Deserializer  Deserializer
XML  XML  Java

version 1.0

The javax.xml.rpc.encoding.Deserializer interface defines a base interface for deserializers. A Deserializer converts an XML representation to a Java object using a specific XML processing mechanism and based on the specified type mapping and encoding style.

Version:
1.0
Author:
Rahul Sharma

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getMechanismType()</td>
<td>Gets the type of the XML processing mechanism and representation used by this Deserializer.</td>
</tr>
</tbody>
</table>

Method Detail
public String getMechanismType()
    Deserializer  XML

    return XML

getMechanismType

String getMechanismType()

Gets the type of the XML processing mechanism and representation used by this Deserializer.

Returns:
    XML processing mechanism type
javax.xml.rpc.encoding Interface DeserializerFactory

All Superinterfaces:
Serializable

public interface DeserializerFactory
extends Serializable

Implements: java.io.Serializable

javax.xml.rpc.encoding.DeserializerFactory Deserializer
DeserializerFactory TypeMapping TypeMappingRegistry

version 1.0

See also javax.xml.rpc.encoding.Serializer

The javax.xml.rpc.encoding.DeserializerFactory is a factory of deserializers. A DeserializerFactory is registered with a TypeMapping instance as part of the TypeMappingRegistry.

Version:
1.0

Author:
Rahul Sharma

See Also:
Serializer

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getDeserializerAs</strong>(String mechanismType)</td>
</tr>
<tr>
<td>Returns a Deserializer for the specified XML processing mechanism type.</td>
</tr>
<tr>
<td><strong>getSupportedMechanismTypes</strong>()</td>
</tr>
<tr>
<td>Returns a list of all XML processing mechanism types supported by this DeserializerFactory.</td>
</tr>
</tbody>
</table>
Method Detail

public Deserializer getDeserializerAs(String mechanismType) throws JAXRPCException:

XML Deserializer

getDeserializerAs

Deserializer getDeserializerAs(String mechanismType)

Returns a Deserializer for the specified XML processing mechanism type.

Parameters:
mechanismType - XML processing mechanism type [TBD: definition of valid constants]

Throws:
JAXRPCException - If DeserializerFactory does not support the specified XML processing mechanism

public java.util.Iterator<E> getSupportedMechanismTypes() throws JAXRPCException:

DeserializerFactory XML

getSupportedMechanismTypes

Iterator getSupportedMechanismTypes()

Returns a list of all XML processing mechanism types supported by
this DeserializerFactory.

**Returns:**
List of unique identifiers for the supported XML processing mechanism types
javax.jms **Interface Destination**

**All Known Subinterfaces:**

- Queue, TemporaryQueue, TemporaryTopic, Topic

---

```java
public interface Destination
```

**Implemented by:** Queue, Topic

Destination JMS API (MOM)

- Destination

JMS API

Destination

Destination JMS

JMS JMS JMS API

- Java Naming and Directory Interface (JNDI) API JMS API JMS JNDI

JMS API

JMS JNDI JMS javax.naming.Referenceable

java.io.Serializable JNDI JavaBeans

---

- JMS
- JMS Java "Java"
- JNDI JMS

JNDI API

Java
A Destination object encapsulates a provider-specific address. The JMS API does not define a standard address syntax. Although a standard address syntax was considered, it was decided that the differences in address semantics between existing message-oriented middleware (MOM) products were too wide to bridge with a single syntax.

Since Destination is an administered object, it may contain provider-specific configuration information in addition to its address.

The JMS API also supports a client's use of provider-specific address names.

Destination objects support concurrent use.

A Destination object is a JMS administered object.

JMS administered objects are objects containing configuration information that are created by an administrator and later used by JMS clients. They make it practical to administer the JMS API in the enterprise.

Although the interfaces for administered objects do not explicitly depend on the Java Naming and Directory Interface (JNDI) API, the JMS API establishes the convention that JMS clients find administered objects by looking them up in a JNDI namespace.

An administrator can place an administered object anywhere in a namespace. The JMS API does not define a naming policy.

It is expected that JMS providers will provide the tools an administrator needs to create and configure administered objects in a JNDI namespace. JMS provider implementations of administered objects should implement the javax.naming.Referenceable and java.io.Serializable interfaces so that they can be stored in all JNDI naming contexts. In addition, it is recommended that these
implementations follow the JavaBeans™ design patterns.

This strategy provides several benefits:

- It hides provider-specific details from JMS clients.
- It abstracts JMS administrative information into objects in the Java programming language ("Java objects") that are easily organized and administered from a common management console.
- Since there will be JNDI providers for all popular naming services, JMS providers can deliver one implementation of administered objects that will run everywhere.

An administered object should not hold on to any remote resources. Its lookup should not use remote resources other than those used by the JNDI API itself.

Clients should think of administered objects as local Java objects. Looking them up should not have any hidden side effects or use surprising amounts of local resources.

Version:
   1.0 - 3 August 1998

Author:
   Mark Hapner, Rich Burridge

See Also:
   Queue, Topic
javax.xml.soap  Interface Detail

**All Superinterfaces:**
- Element, Node, SOAPElement, SOAPFaultElement

```
public interface Detail
extends SOAPFaultElement

Implements: SOAPFaultElement
```

```
DetailEntry   DetailEntry   SOAPBody
            SOAPFault.getDetail  Detail  SOAPFault  Detail
DetailEntry   Detail   Detail  DetailEntry

sf    SOAPFault    Detail ( d)    DetailEntry  d  
DetailEntry addDetailEntry    Name    Se    Name
SOAPEnvelope

Detail d = sf.getDetail();
Name name = se.createName("GetLastTradePrice", "WOMBAT",
"http://www.wombat.org/trader");
d.addDetailEntry(name);
Iterator it = d.getDetailEntries();
```

A container for DetailEntry objects. DetailEntry objects give detailed error information that is application-specific and related to the SOAPBody object that contains it.

A Detail object, which is part of a SOAPFault object, can be retrieved using the method SOAPFault.getDetail. The Detail interface provides two methods. One creates a new DetailEntry object and also automatically adds it to the Detail object. The second method gets a list of the DetailEntry objects contained in a Detail object.
The following code fragment, in which \texttt{sf} is a \texttt{SOAPFault} object, gets its \texttt{Detail} object (\texttt{d}), adds a new \texttt{DetailEntry} object to \texttt{d}, and then gets a list of all the \texttt{DetailEntry} objects in \texttt{d}. The code also creates a \texttt{Name} object to pass to the method \texttt{addDetailEntry}. The variable \texttt{se}, used to create the \texttt{Name} object, is a \texttt{SOAPEnvelope} object.

```java
Detail d = sf.getDetail();
Name name = se.createName("GetLastTradePrice", "WOMBAT", "http://www.wombat.org/trader");
d.addDetailEntry(name);
Iterator it = d.getDetailEntries();
```

### Field Summary

<table>
<thead>
<tr>
<th>Details inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{addDetailEntry(\texttt{Name} name)}</td>
</tr>
</tbody>
</table>

\texttt{DetailEntry} \texttt{addDetailEntry(\texttt{Name} name)}

Creates a new \texttt{DetailEntry} object with the given name and adds it to this \texttt{Detail} object.
<table>
<thead>
<tr>
<th><strong>DetailEntry</strong></th>
<th><code>addDetailEntry(QName qname)</code> Creates a new <code>DetailEntry</code> object with the given QName and adds it to this <code>Detail</code> object.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Iterator</strong></td>
<td><code>getDetailEntries()</code> Gets an Iterator over all of the <code>DetailEntry</code>S in this <code>Detail</code> object.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface `javax.xml.soap.SOAPElement`

- `addAttribute`, `addAttribute`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addNamespaceDeclaration`, `addTextNode`, `createQName`, `getAllAttributes`, `getAllAttributesAsQNames`, `getAttributeValue`, `getAttributeValue`, `getChildElements`, `get_childElements`, `get_childElements`, `getElementName`, `getElementQName`, `getEncodingStyle`, `getNamespacePrefixes`, `getNamespaceURI`, `getVisibleNamespacePrefixes`, `removeAttribute`, `removeContents`, `removeNamespaceDeclaration`, `setElementQName`, `setEncodingStyle`

### Methods inherited from interface `javax.xml.soap.Node`

- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

### Methods inherited from interface `org.w3c.dom.Node`

- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildNodes`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getParentNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`, `replaceChild`, `setNameValue`, `setPrefix`, `setTextContent`, `setUserData`

### Methods inherited from interface `org.w3c.dom.Element`

- `getAttribute`, `getAttributeNode`, `getAttributeNodeNS`, `getAttributeNS`, `getElementsByTagName`, `getElementsByTagNameNS`, `getSchemaTypeInfo`, `getTagName`, `hasAttribute`, `hasAttributeNS`, `removeAttribute`, `removeAttributeNode`, `removeAttributeNS`, `setAttribute`, `setAttributeNode`, `setAttributeNodeNS`, `setAttributeNS`, `setIdAttribute`, `setIdAttributeNode`, `setIdAttributeNS`
Methods inherited from interface org.w3c.dom.Node

appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getPrefix, getPreviousSibling, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, hasAttributes, hasChildNodes, insertBefore, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData

Method Detail

class addDetailEntry(\texttt{Name} name) throws SOAPException

\begin{itemize}
\item \texttt{DetailEntry} addDetailEntry(\texttt{Name} name) throws SOAPException
\end{itemize}

Create a new \texttt{DetailEntry} object with the given name and adds it to this \texttt{Detail} object.

\textbf{Parameters:}

- name - a \texttt{Name} object identifying the new \texttt{DetailEntry} object

\textbf{Throws:}

- \texttt{SOAPException} - thrown when there is a problem in adding a \texttt{DetailEntry} object to this \texttt{Detail} object.

\textbf{See Also:}

- addDetailEntry(QName qname)
public **DetailEntry**
addDetailEntry(javax.xml.namespace.QName qname) throws **SOAPException**

<table>
<thead>
<tr>
<th>QName</th>
<th>DetailEntry</th>
<th>Detail Name</th>
</tr>
</thead>
</table>

**Throws**  
**SOAPException**: DetailEntry Detail  

**since**  
SAAJ 1.3  

**See also**  
addDetailEntry(Name name)

---

**addDetailEntry**

**DetailEntry** **addDetailEntry**(QName qname)  
throws **SOAPException**

Creates a new `DetailEntry` object with the given QName and adds it to this `Detail` object. This method is the preferred over the one using `Name`.

**Parameters:**

qname - a QName object identifying the new `DetailEntry` object

**Throws:**

**SOAPException** - thrown when there is a problem in adding a `DetailEntry` object to this `Detail` object.

**Since:**

SAAJ 1.3

**See Also:**

addDetailEntry(Name name)

---

public **java.util.Iterator<E>** getDetailEntries()

<table>
<thead>
<tr>
<th>return</th>
<th>Iterator</th>
<th>Detail</th>
<th>DetailEntry</th>
</tr>
</thead>
</table>

**getDetailEntries**
Iterator getDetailEntries()

Gets an Iterator over all of the DetailEntry objects in this Detail object.

Returns:
        an Iterator object over the DetailEntry objects in this Detail object
javax.xml.soap  Interface DetailEntry

All Superinterfaces:
Element, Node, SOAPElement

public interface DetailEntry
extends SOAPElement

Implements: SOAPElement

Detail      SOAPFault      DetailEntry      SOAPBody

The content for a Detail object, giving details for a SOAPFault object. A DetailEntry object, which carries information about errors related to the SOAPBody object that contains it, is application-specific.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,</td>
</tr>
<tr>
<td>ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,</td>
</tr>
<tr>
<td>ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>
## Method Summary

### Methods inherited from interface `javax.xml.soap.SOAPElement`
- `addAttribute`, `addAttribute`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addNamespaceDeclaration`, `addTextNode`, `createQName`, `getAllAttributes`, `getAllAttributesAsQNames`, `getAttributeValue`, `getAttributeValue`, `getChildrenElements`, `getChildrenElements`, `getChildrenElements`, `getChildrenElements`, `getElementName`, `getElementQName`, `getElementQName`, `getEncodingStyle`, `getNamespacePrefixes`, `getNamespaceURI`, `getVisibleNamespacePrefixes`, `removeAttribute`, `removeContents`, `removeNamespaceDeclaration`, `setElementQName`, `setEncodingStyle`

### Methods inherited from interface `javax.xml.soap.Node`
- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

### Methods inherited from interface `org.w3c.dom.Node`
- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildNodes`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getOwnerNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`, `replaceChild`, `setNodeValue`, `setPrefix`, `setTextContent`, `setUserData`

### Methods inherited from interface `org.w3c.dom.Element`
- `getAttribute`, `getAttributeNode`, `getAttributeNodeNS`, `getAttributeNS`, `getElementsByTagName`, `getElementsByTagNameNS`, `getSchemaTypeInfo`, `getTagName`, `hasAttribute`, `hasAttributeNS`, `removeAttribute`, `removeAttributeNode`, `removeAttributeNS`, `setAttribute`, `setAttributeNode`, `setAttributeNodeNS`, `setIdAttribute`, `setIdAttributeNode`, `setIdAttributeNS`, `setIdAttributeNS`
Methods inherited from interface org.w3c.dom.Node

appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData
javax.persistence Annotation Type DiscriminatorColumn

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface DiscriminatorColumn

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

@link InheritanceType#SINGLE_TABLE SINGLE_TABLE} JOINED

DiscriminatorColumn "DTYPE" DiscriminatorType.STRING

@Entity
@Table(name="CUST")
@Inheritance(strategy=SINGLE_TABLE)
@DiscriminatorColumn(name="DISC", discriminatorType=STRING,length=20)
public class Customer {
... }

@Entity
public class ValuedCustomer extends Customer {
... }

since Java Persistence 1.0 en

Is used to define the discriminator column for the SINGLE_TABLE and
JOINED inheritance mapping strategies.

The strategy and the discriminator column are only specified in the root of
an entity class hierarchy or subhierarchy in which a different inheritance
strategy is applied.

If the DiscriminatorColumn annotation is missing, and a discriminator
column is required, the name of the discriminator column defaults to
"DTYPE" and the discriminator type to `DiscriminatorType.STRING`.

Example:
```java
@Entity
@Table(name="CUST")
@Inheritance(strategy=SINGLE_TABLE)
@DiscriminatorColumn(name="DISC", discriminatorType=STRING, length=20)
public class Customer {
    ...
}

@Entity
public class ValuedCustomer extends Customer {
    ...
}
```

Since:
Java Persistence 1.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String</code></td>
<td><code>columnDefinition</code></td>
<td>(Optional) The SQL fragment that is used when generating the DDL for the discriminator column.</td>
</tr>
<tr>
<td><code>DiscriminatorType</code></td>
<td><code>discriminatorType</code></td>
<td>(Optional) The type of object/column to use as a class discriminator.</td>
</tr>
<tr>
<td><code>int</code></td>
<td><code>length</code></td>
<td>(Optional) The column length for String-based discriminator types.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>name</code></td>
<td>(Optional) The name of column to be used for the discriminator.</td>
</tr>
</tbody>
</table>

abstract public String name()

name

public abstract String name
(Optional) The name of column to be used for the discriminator.

**Default:**

"DTYPE"

---

```java
abstract public DiscriminatorType discriminatorType()
/
   DiscriminatorType.STRING
```

**discriminatorType**

**public abstract**  

```java
DiscriminatorType discriminatorType
```

(Optional) The type of object/column to use as a class discriminator.  
Defaults to `DiscriminatorType.STRING`.

**Default:**

`STRING`

---

```java
abstract public String columnDefinition()
```

**SQL**

**SQL**

**columnDefinition**

**public abstract**  

```java
String columnDefinition
```

(Optional) The SQL fragment that is used when generating the DDL  
for the discriminator column.

Defaults to the provider-generated SQL to create a column of the  
specified discriminator type.
abstract public int length()
String

length

public abstract int length

(Optional) The column length for String-based discriminator types. Ignored for other discriminator types.

Default:
31
javax.persistence  **Enum DiscriminatorType**

java.lang.Object  
   - java.lang.Enum<DiscriminatorType>  
   - javax.persistence.DiscriminatorType

**All Implemented Interfaces:**  
Serializable, Comparable<DiscriminatorType>

```java
public enum DiscriminatorType
extends Enum<DiscriminatorType>

Extends: Enum<E>
```

**since**  
Java Persistence 1.0  
en

Defines supported types of the discriminator column.

**Since:**  
Java Persistence 1.0

---

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>Single character as the discriminator type</td>
</tr>
<tr>
<td>INTEGER</td>
<td>Integer as the discriminator type</td>
</tr>
<tr>
<td>STRING</td>
<td>String as the discriminator type</td>
</tr>
</tbody>
</table>

### Method Summary

---
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Enum**
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`  

**Methods inherited from class java.lang.Object**
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`  

## Enum Constant Detail

**STRING**

- `public static final DiscriminatorType STRING`  
  - String as the discriminator type

---

**CHAR**

- `public static final DiscriminatorType CHAR`  
  - Single character as the discriminator type

---
public static final DiscriminatorType INTEGER

Integer as the discriminator type

### Method Detail

#### final public static DiscriminatorType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(DiscriminatorType c : DiscriminatorType.values())
    System.out.println(c);
```

**Returns:**
- an array containing the constants of this enum type, in the order they're declared

#### public static DiscriminatorType[] values()

**Returns:**
- an array containing the constants of this enum type, in the order they're declared

#### public static DiscriminatorType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)
Parameters:
  name - the name of the enum constant to be returned.

Returns:
  the enum constant with the specified name

Throws:
  IllegalArgumentException - if this enum type has no constant with the specified name
javax.persistence  Annotation Type DiscriminatorValue

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface DiscriminatorValue

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

DiscriminatorValue

Discriminator

DiscriminatorType  STRING

@Entity
@Table(name="CUST")
@Inheritance(strategy=SINGLE_TABLE)
@DiscriminatorColumn(name="DISC", discriminatorType=STRING, length=20)
@DiscriminatorValue("CUSTOMER")
public class Customer {
  ...
}

@Entity
@DiscriminatorValue("VCUSTOMER")
public class ValuedCustomer extends Customer {
  ...
}

since  Java Persistence 1.0  en

Is used to specify the value of the discriminator column for entities of the given type. The DiscriminatorValue annotation can only be specified on a concrete entity class. If the DiscriminatorValue annotation is not specified and a discriminator column is used, a provider-specific function will be used to generate a value representing the entity type. If the DiscriminatorType is STRING, the discriminator value default is the entity name.
The inheritance strategy and the discriminator column are only specified in the root of an entity class hierarchy or subhierarchy in which a different inheritance strategy is applied. The discriminator value, if not defaulted, should be specified for each entity class in the hierarchy.

Example:

```java
@Entity
@Table(name="CUST")
@Inheritance(strategy=SINGLE_TABLE)
@DiscriminatorColumn(name="DISC", discriminatorType=STRING,length=20)
@DiscriminatorValue("CUSTOMER")
public class Customer {
    ...
}

@Entity
@DiscriminatorValue("VCUSTOMER")
public class ValuedCustomer extends Customer {
    ...
}
```

Since: Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>value</code></td>
<td>(Optional) The value that indicates that the row is an entity of the annotated entity type.</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String value()
public abstract String value

(Optional) The value that indicates that the row is an entity of the annotated entity type.

If the DiscriminatorValue annotation is not specified and a discriminator column is used, a provider-specific function will be used to generate a value representing the entity type. If the DiscriminatorType is String, the discriminator value default is the entity name.
<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
<th>DETAIL:</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREFIX</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.ws Interface Dispatch\<T\>

All Superinterfaces:

BindingProvider

---

public interface Dispatch\<T\>

extends BindingProvider

Implements: BindingProvider

Dispatch javax.xml.ws.Service Dispatch

since JAX-WS 2.0

The Dispatch interface provides support for the dynamic invocation of a service endpoint operations. The javax.xml.ws.Service interface acts as a factory for the creation of Dispatch instances.

Since:

JAX-WS 2.0

---

**Field Summary**

---

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.xml.ws.BindingProvider</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDPOINT_ADDRESS_PROPERTY, PASSWORD_PROPERTY,</td>
</tr>
<tr>
<td>SESSION_MAINTAIN_PROPERTY, SOAPACTION_URI_PROPERTY,</td>
</tr>
<tr>
<td>SOAPACTION_USE_PROPERTY, USERNAME_PROPERTY</td>
</tr>
</tbody>
</table>

---

**Method Summary**

---

\begin{Verbatim}
I invoke(I msg)
\end{Verbatim}

Invoke a service operation synchronously.

\begin{Verbatim}
Response\<I\> invokeAsync(I msg)
\end{Verbatim}
Invoke a service operation asynchronously.

```java
Future<?> invokeAsync(T msg, AsyncHandler<T> handler)
```

Invoke a service operation asynchronously.

```java
void invokeOneWay(T msg)
```

Invokes a service operation using the one-way interaction mode.

Methods inherited from interface javax.xml.ws.BindingProvider

```java
getBinding, getRequestContext, getResponseContext
```

Method Detail

```java
T invoke(T msg)
```

Invoke a service operation synchronously. The client is responsible for ensuring that the `msg` object when marshalled is formed according to the requirements of the protocol binding in use.

**Parameters:**

- `msg` - An object that will form the message or payload of the message used to invoke the operation.

**Returns:**

The response message or message payload to the operation invocation.

**Throws:**

- `WebServiceException` - If a fault occurs during communication with the service
- `WebServiceException` - If there is any error in the configuration of the Dispatch instance
invokeAsync

Response<T> invokeAsync(T msg)

Invoke a service operation asynchronously. The method returns without waiting for the response to the operation invocation, the results of the operation are obtained by polling the returned Response. The client is responsible for ensuring that the msg object when marshalled is formed according to the requirements of the protocol binding in use.

Parameters:
msg - An object that will form the message or payload of the message used to invoke the operation.

Returns:
The response message or message payload to the operation invocation.

Throws:
WebServiceException - If there is any error in the configuration of the Dispatch instance

invokeAsync

Future<?> invokeAsync(T msg, AsyncHandler<T> handler)

Invoke a service operation asynchronously. The method returns without waiting for the response to the operation invocation, the results of the operation are communicated to the client via the passed in handler. The client is responsible for ensuring that the msg object when marshalled is formed according to the requirements of the protocol binding in use.

Parameters:
msg - An object that will form the message or payload of the message used to invoke the operation.
handler - The handler object that will receive the response to the
invokeOneWay

void invokeOneWay(T msg)

Invokes a service operation using the one-way interaction mode. The operation invocation is logically non-blocking, subject to the capabilities of the underlying protocol, no results are returned. When the protocol in use is SOAP/HTTP, this method must block until an HTTP response code has been received or an error occurs. The client is responsible for ensuring that the msg object when marshalled is formed according to the requirements of the protocol binding in use.

Parameters:

msg - An object that will form the message or payload of the message used to invoke the operation.

Throws:

WebServiceException - If there is any error in the configuration of the Dispatch instance or if an error occurs during the invocation.
javax.resource.spi Interface
DissociatableManagedConnection

public interface DissociatableManagedConnection

ManagedConnection
version 1.0

This is a mix-in interface that may be optionally implemented by a ManagedConnection implementation. An implementation of this interface must support the lazy connection association optimization.

Version:
1.0
Author:
Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void dissociateConnections()</td>
<td>This method is called by an application server (that is capable of lazy connection association optimization) in order to dissociate a ManagedConnection instance from all of its connection handles.</td>
</tr>
</tbody>
</table>

Method Detail

public void dissociateConnections() throws ResourceException
dissociateConnections

```java
void dissociateConnections()
    throws ResourceException
```

This method is called by an application server (that is capable of lazy connection association optimization) in order to dissociate a ManagedConnection instance from all of its connection handles.

**Throws:**
- `ResourceException` - generic exception if operation fails.
- `ResourceAdapterInternalException` - resource adapter internal error condition
- `IllegalStateException` - Illegal state for calling connection cleanup. Example - if a localtransaction is in progress that doesn't allow connection cleanup.
javax.xml.bind.annotation Interface
DomHandler<ElementT,ResultT extends Result>

All Known Implementing Classes:
   W3CDomHandler

public interface DomHandler<ElementT,ResultT extends Result>

Implemented by: W3CDomHandler

DOM

   XmlAnyElement   XML   W3C DOM

XML  DOM  JAXB

   since   JAXB2.0

Converts an element (and its descendants) from/to DOM (or similar) representation.

Implementations of this interface will be used in conjunction with XmlAnyElement annotation to map an element of XML into a representation of infoset such as W3C DOM.

Implementations hide how a portion of XML is converted into/from such DOM-like representation, allowing JAXB providers to work with arbitrary such library.

This interface is intended to be implemented by library writers and consumed by JAXB providers. None of those methods are intended to be called from applications.

Since:
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createUnmarshaller(ValidationEventHandler errorHandler)</code></td>
<td>When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a <code>Result</code> object.</td>
</tr>
<tr>
<td><code>getElement(ResultT rt)</code></td>
<td>Once the portion is sent to the <code>Result</code>.</td>
</tr>
<tr>
<td><code>marshal(ElementT n, ValidationEventHandler errorHandler)</code></td>
<td>This method is called when a JAXB provider needs to marshal an element to XML.</td>
</tr>
</tbody>
</table>

## Method Detail

```java
public javax.xml.transform.Result
createUnmarshaller(ValidationEventHandler errorHandler)
```

When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a `Result` object.

```java
return null
```

```java
createUnmarshaller
```

When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a `Result` object.
an infoset representation, it first calls this method to create a Result object.

A JAXB provider will then send a portion of the XML into the given result. Such a portion always form a subtree of the whole XML document rooted at an element.

**Parameters:**

- `errorHandler` - if any error happens between the invocation of this method and the invocation of `getElement(Result)`, they must be reported to this handler. The caller must provide a non-null error handler. The Result object created from this method may hold a reference to this error handler.

**Returns:**

null if the operation fails. The error must have been reported to the error handler.

---

getElement

```java
ElementT getElement(ResultT rt)
```

Once the portion is sent to the Result. This method is called by a JAXB provider to obtain the unmarshalled element representation.

Multiple invocations of this method may return different objects. This method can be invoked only when the whole sub-tree are fed to the Result object.

**Parameters:**

- `rt` - The Result object created by `createUnmarshaller(ValidationEventHandler)`.

**Returns:**

null if the operation fails. The error must have been reported to the error handler.
marshal

Source marshal(ElementT n, ValidationEventHandler errorHandler)

This method is called when a JAXB provider needs to marshal an element to XML.

If non-null, the returned Source must contain a whole document rooted at one element, which will then be weaved into a bigger document that the JAXB provider is marshalling.

Parameters:

errorHandler - Receives any errors happened during the process of converting an element into a Source. The caller must provide a non-null error handler.

Returns:

null if there was an error. The error should have been reported to the handler.
javax.faces.convert Class DoubleConverter

java.lang.Object
   \ javax.faces.convert.DoubleConverter

All Implemented Interfaces:
   Converter

public class DoubleConverter

extends Object
implements Converter

Implements: Converter

java.lang.Double double

Converter implementation for java.lang.Double (and double primitive) values.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>static String</td>
</tr>
</tbody>
</table>

Constructor Summary
### DoubleConverter()

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified <code>UIComponent</code>, into a model data object that is appropriate for being stored during the <code>Apply Request Values</code> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified <code>UIComponent</code>, into a String that is suitable for being included in the response generated during the <code>Render Response</code> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`  

### Field Detail

**CONVERTER_ID**

```
public static final String CONVERTER_ID
```

The standard converter id for this converter.

**See Also:**

- [Constant Field Values](#)
DOUBLE_ID

public static final String DOUBLE_ID

The message identifier of the FacesMessage to be created if the conversion to Double fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Double value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail
public DoubleConverter()

DoubleConverter

public DoubleConverter()

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
</table>

public Object getAsObject(FacesContext context, UIComponent component, String value)

Throws ConverterException: NullPointerException

Throws NullPointerException: NullPointerException

getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

Description copied from interface: Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

Specified by:
getAsObject in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - String value to be converted (may be null)

Returns:
null if the value to convert is null, otherwise the result of the
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws  ConverterException: if conversion cannot be successfully performed
Throws  NullPointerException - if context or component is null

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
performed

NullPointerException - if context or component is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.holders Class DoubleHolder

java.lang.Object
   ^ javax.xml.rpc.holders.DoubleHolder

All Implemented Interfaces:
   Holder

public final class DoubleHolder

extends Object
implements Holder

Implements: Holder

Field Summary

| double value |

Constructor Summary

DoubleHolder()

DoubleHolder(double mydouble)

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Field Detail

value
public double value

Constructor Detail

public DoubleHolder()

DoubleHolder

public DoubleHolder()

double

public DoubleHolder(double mydouble)

DoubleHolder

public DoubleHolder(double mydouble)
PS:
javax.faces.validator Class DoubleRangeValidator

java.lang.Object
  ^javax.faces.validator.DoubleRangeValidator

All Implemented Interfaces:
   javax.faces.validator.DoubleRangeValidator
   javax.faces.converter.Converter

javax.faces.converter.Locale

javax.faces.converter.Number

public class DoubleRangeValidator
extends Object
implements Validator, StateHolder

Implements: Validator, StateHolder

DoubleRangeValidator extends Validator

• null
• double String TYPE_MESSAGE_ID
  javax.faces.validator.ValidatorException
• Validator maximum minimum
  #NOT_IN_RANGE_MESSAGE_ID javax.faces.validator.ValidatorException
• Validator maximum MAXIMUM_MESSAGE_ID
  javax.faces.validator.ValidatorException
• Validator minimum MINIMUM_MESSAGE_ID
  javax.faces.validator.ValidatorException

ValidatorException ID javax.faces.Number Converter Locale

DoubleRangeValidator is a Validator that checks the value of the corresponding component against specified minimum and maximum values. The following algorithm is implemented:

• If the passed value is null, exit immediately.
• If the current component value is not a floating point type, or a String that is convertible to double, throw a `ValidatorException` containing a `TYPE_MESSAGE_ID` message.

• If both a `maximum` and `minimum` property has been configured on this `Validator`, check the component value against both limits. If the component value is not within this specified range, throw a `ValidatorException` containing a `NOT_IN_RANGE_MESSAGE_ID` message.

• If a `maximum` property has been configured on this `Validator`, check the component value against this limit. If the component value is greater than the specified maximum, throw a `ValidatorException` containing a `MAXIMUM_MESSAGE_ID` message.

• If a `minimum` property has been configured on this `Validator`, check the component value against this limit. If the component value is less than the specified minimum, throw a `ValidatorException` containing a `MINIMUM_MESSAGE_ID` message.

For all of the above cases that cause a `ValidatorException` to be thrown, if there are parameters to the message that match up with validator parameters, the values of these parameters must be converted using the `Converter` registered in the application under the converter id `javax.faces.Number`. This allows the values to be localized according to the current `Locale`.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String</th>
<th>MAXIMUM_MESSAGE_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the maximum value check fails.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>MINIMUM_MESSAGE_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the minimum value check fails.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>NOT_IN_RANGE_MESSAGE_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>TYPE_MESSAGE_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructor Summary</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td></td>
</tr>
<tr>
<td><strong>DoubleRangeValidator</strong>()</td>
<td></td>
</tr>
<tr>
<td>Construct a <code>Validator</code> with no preconfigured limits.</td>
<td></td>
</tr>
<tr>
<td><strong>DoubleRangeValidator</strong>(double maximum)</td>
<td></td>
</tr>
<tr>
<td>Construct a <code>Validator</code> with the specified preconfigured limit.</td>
<td></td>
</tr>
<tr>
<td><strong>DoubleRangeValidator</strong>(double maximum, double minimum)</td>
<td></td>
</tr>
<tr>
<td>Construct a <code>Validator</code> with the specified preconfigured limits.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boolean equals</strong>(Object otherObj)</td>
</tr>
<tr>
<td><strong>double getMaximum</strong>()</td>
</tr>
<tr>
<td>Return the maximum value to be enforced by this <code>Validator</code> or <code>Double.MAX_VALUE</code> if it has not been set.</td>
</tr>
<tr>
<td><strong>double getMinimum</strong>()</td>
</tr>
<tr>
<td>Return the minimum value to be enforced by this <code>Validator</code>, or <code>Double.MIN_VALUE</code> if it has not been set.</td>
</tr>
<tr>
<td><strong>int hashCode</strong>()</td>
</tr>
<tr>
<td><strong>boolean isTransient</strong>()</td>
</tr>
<tr>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td><strong>void restoreState</strong>(FacesContext context, Object state)</td>
</tr>
<tr>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><strong>Object saveState</strong>(FacesContext context)</td>
</tr>
<tr>
<td>Gets the state of the instance as a <code>Serializable</code> Object.</td>
</tr>
</tbody>
</table>
void setMaximum(double maximum)
   Set the maximum value to be enforced by this Validator.

void setMinimum(double minimum)
   Set the minimum value to be enforced by this Validator.

void setTransient(boolean transientValue)
   Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

void validate(FacesContext context, UIComponent component, Object value)
   Perform the correctness checks implemented by this Validator against the specified UIComponent.

Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, toString, wait, wait

Field Detail

VALIDATOR_ID

public static final String VALIDATOR_ID
   The standard converter id for this converter.

See Also:
   Constant Field Values

MAXIMUM_MESSAGE_ID

public static final String MAXIMUM_MESSAGE_ID
The message identifier of the `FacesMessage` to be created if the maximum value check fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the configured maximum value.
- `{1}` replaced by a `String` whose value is the label of the input component that produced this message.

See Also:

Constant Field Values

---

**MINIMUM_MESSAGE_ID**

definition

The message identifier of the `FacesMessage` to be created if the minimum value check fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the configured minimum value.
- `{1}` replaced by a `String` whose value is the label of the input component that produced this message.

See Also:

Constant Field Values

---

**NOT_IN_RANGE_MESSAGE_ID**

definition

The message identifier of the `FacesMessage` to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set. The message format string for this message may optionally include the following
placeholders:

- \{0\} replaced by the configured minimum value.
- \{1\} replaced by the configured maximum value.
- \{2\} replaced by a \texttt{String} whose value is the label of the input component that produced this message.

See Also:

\texttt{Constant Field Values}

\begin{verbatim}
public static final String TYPE_MESSAGE_ID

    The message identifier of the \texttt{FacesMessage} to be created if the current value of this component is not of the correct type. The message format string for this message may optionally include a \{0\} placeholder that will be replaced by a \texttt{String} whose value is the label of the input component that produced this message.

See Also:

\texttt{Constant Field Values}

\end{verbatim}

\textbf{Constructor Detail}

\begin{verbatim}
public DoubleRangeValidator()

\texttt{Validator}

\end{verbatim}

\textbf{DoubleRangeValidator}

\begin{verbatim}
public DoubleRangeValidator()

\end{verbatim}
Construct a validator with no preconfigured limits.

```
public DoubleRangeValidator(double maximum)
```

Validator

```
maximum
```

DoubleRangeValidator

```
public DoubleRangeValidator(double maximum)
```

Construct a validator with the specified preconfigured limit.

Parameters:

- `maximum`: Maximum value to allow

```
public DoubleRangeValidator(double maximum, double minimum)
```

Validator

```
maximum
minimum
```

DoubleRangeValidator

```
public DoubleRangeValidator(double maximum, double minimum)
```

Construct a validator with the specified preconfigured limits.

Parameters:

- `maximum`: Maximum value to allow
**Minimum** - Minimum value to allow

### Method Detail

#### public double getMaxMaximum()

- **Validator** Double.MAX_VALUE

#### getMaxMaximum

**public double getMaxMaximum()**

Return the maximum value to be enforced by this **Validator** or Double.MAX_VALUE if it has not been set.

#### public void setMaximum(double maximum)

- **Validator**

#### setMaximum

**public void setMaximum(double maximum)**

Set the maximum value to be enforced by this **Validator**.

**Parameters:**
- maximum - The new maximum value

#### public double getMinimum()
**Validator**    Double.MIN_VALUE

### getMinimum

```java
public double getMinimum()
```

Return the minimum value to be enforced by this `Validator`, or `Double.MIN_VALUE` if it has not been set.

### setMinimum

```java
public void setMinimum(double minimum)
```

Set the minimum value to be enforced by this `Validator`.

**Parameters:**
- `minimum` - The new minimum value

### validate

```java
public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException
```

Throws `NullPointerException`: `context`, `component`, `null`

Throws `ValidatorException`: `Null Pointer Exception`
public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException

Description copied from interface: Validator

Perform the correctness checks implemented by this Validator against the specified UIComponent. If any violations are found, a ValidatorException will be thrown containing the FacesMessage describing the failure.

Specified by:
validate in interface Validator

Parameters:
context - FacesContext for the request we are processing
component - UIComponent we are checking for correctness
value - the value to validate

Throws:
NullPointerException - if context or component is null
ValidatorException - if validation fails

public boolean equals(Object otherObj)

equals

public boolean equals(Object otherObj)

Overrides:
equals in class Object

public int hashCode()

hashCode
public int hashCode()

Overrides:
hashCode in class Object

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
saveState in interface StateHolder

public void restoreState(FacesContext context, Object state)
**restoreState**

```java
public void restoreState(FacesContext context, Object state)
```

Description copied from interface: `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
- `restoreState` in interface `StateHolder`

---

**public boolean isTransient()**

**isTransient**

```java
public boolean isTransient()
```

Description copied from interface: `StateHolder`

If true, the Object implementing this interface must not participate in state saving or restoring.

**Specified by:**
- `isTransient` in interface `StateHolder`

---

**public void setTransient(boolean transientValue)**
**setTransient**

public void **setTransient**(boolean transientValue)

**Description copied from interface:** StateHolder

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Specified by:**

setTransient in interface StateHolder

**Parameters:**

transientValue - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.holders  Class DoubleWrapperHolder

java.lang.Object
   \ javax.xml.rpc.holders.DoubleWrapperHolder

All Implemented Interfaces:
   Holder

public final class DoubleWrapperHolder
extends Object
implements Holder

Implements: Holder

Field Summary

| Double value |

Constructor Summary

| DoubleWrapperHolder() |
| DoubleWrapperHolder(Double mydouble) |

Method Summary

| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |
Field Detail

value

public Double value

Constructor Detail

public DoubleWrapperHolder()

public DoubleWrapperHolder()

public DoubleWrapperHolder(Double mydouble)

DoubleWrapperHolder

public DoubleWrapperHolder(Double mydouble)
PS:
javax.xml.stream.events Interface DTD

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface DTD
extends XMLEvent

Implements: XMLEvent

DTD

version 1.0

This is the top level interface for events dealing with DTDs

Version: 1.0

Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants
ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT

Method Summary

getString

getDocumentTypeDeclaration()

Returns the entire Document Type Declaration as a string,
getEntities()  
Return a List containing the general entities, both external and internal, declared in the DTD.

getNotations()  
Return a List containing the notations declared in the DTD.

getProcessedDTD()  
Returns an implementation defined representation of the DTD.

Methods inherited from interface javax.xml.stream.events.XMLEvent
asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public String getDocumentTypeDeclaration()
DTD null null XML 1.0
doctypedecl

getDocumentTypeDeclaration

String getDocumentTypeDeclaration()

Returns the entire Document Type Declaration as a string, including the internal DTD subset. This may be null if there is not an internal subset. If it is not null it must return the entire Document Type Declaration which matches the doctypedecl production in the XML 1.0 specification
public Object getProcessedDTD()

Object getProcessedDTD()

Returns an implementation defined representation of the DTD. This method may return null if no representation is available.

public java.util.List<E> getNotations()

List DTD NotationDeclaration

return NotationDeclaration

See also javax.xml.stream.events.NotationDeclaration

getNotations

List getNotations()

Return a List containing the notations declared in the DTD. This list must contain NotationDeclaration events.

Returns:
an unordered list of NotationDeclaration events

See Also:
NotationDeclaration

public java.util.List<E> getEntities()

DTD EntityDeclaration

return EntityDeclaration

See also javax.xml.stream.events.EntityDeclaration
getEntities

List getEntities()

Return a List containing the general entities, both external and internal, declared in the DTD. This list must contain EntityDeclaration events.

Returns:
an unordered list of EntityDeclaration events

See Also:
EntityDeclaration

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.ejb Class DuplicateKeyException

java.lang.Object
   ▼ java.lang.Throwble
      ▼ java.lang.Exception
         ▼ javax.ejb.CreateException
            ▼ javax.ejb.DuplicateKeyException

All Implemented Interfaces:
   Serializable

public class DuplicateKeyException
   extends CreateException

Extends: Throwable > Exception > CreateException

EJB DuplicateKeyException Bean home

The DuplicateKeyException exception is thrown if an entity EJB object cannot be created because an object with the same key already exists. This exception is thrown by the create methods defined in an entity Bean's home interface.

See Also:
   Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DuplicateKeyException()</td>
<td>Constructs a DuplicateKeyException with no detail message.</td>
</tr>
<tr>
<td>DuplicateKeyException(String message)</td>
<td>Constructs a DuplicateKeyException with the specified detail message.</td>
</tr>
</tbody>
</table>
### Method Summary

Methods inherited from class java.lang.Throwable

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

**public DuplicateKeyException()**

DuplicatesKeyException

**DuplicateKeyException**

**public DuplicateKeyException()**

Constructs a DuplicateKeyException with no detail message.

**public DuplicateKeyException(String message)**

DuplicatesKeyException

**DuplicateKeyException**

**public DuplicateKeyException(String message)**

Constructs a DuplicateKeyException with the specified detail message.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface DynamicAttributes

setDynamicAttribute()

since 2.0

For a tag to declare that it accepts dynamic attributes, it must implement this interface. The entry for the tag in the Tag Library Descriptor must also be configured to indicate dynamic attributes are accepted. For any attribute that is not declared in the Tag Library Descriptor for this tag, instead of getting an error at translation time, the setDynamicAttribute() method is called, with the name and value of the attribute. It is the responsibility of the tag to remember the names and values of the dynamic attributes.

Since:
JSP 2.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void setDynamicAttribute(String uri, String localName, Object value)</td>
</tr>
<tr>
<td>Called when a tag declared to accept dynamic attributes is passed an attribute that is not declared in the Tag Library Descriptor.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void setDynamicAttribute(String uri, String</td>
</tr>
</tbody>
</table>
localName, Object value) throws JspException

Throws  JspException: doStartTag() doTag()

setDynamicAttribute

void setDynamicAttribute(String uri,
              String localName,
              Object value)
throws JspException

Called when a tag declared to accept dynamic attributes is passed an attribute that is not declared in the Tag Library Descriptor.

Parameters:
uri - the namespace of the attribute, or null if in the default namespace.
localName - the name of the attribute being set.
value - the value of the attribute

Throws:
JspException - if the tag handler wishes to signal that it does not accept the given attribute. The container must not call doStartTag() or doTag() for this tag.
PS:
javax.faces.component Interface EditableValueHolder

All Superinterfaces:
ValueHolder

All Known Implementing Classes:
HtmlInputHidden, HtmlInputSecret, HtmlInputText,
HtmlInputTextarea, HtmlSelectBooleanCheckbox,
HtmlSelectManyCheckbox, HtmlSelectManyListbox,
HtmlSelectManyMenu, HtmlSelectOneListbox, HtmlSelectOneMenu,
HtmlSelectOneRadio, UIInput, UISelectBoolean, UISelectMany,
UISelectOne

public interface EditableValueHolder
extends ValueHolder

Implements: ValueHolder
Implemented by: UIInput

EditableValueHolder is an extension of ValueHolder that describes additional features supported by editable components, including ValueChangeEvent and Validators.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addValidator(Validator validator)</td>
<td>Add a Validator instance to the set associated with the component.</td>
</tr>
<tr>
<td>void addValueChangeListener(ValueChangeEvent listener)</td>
<td></td>
</tr>
</tbody>
</table>
void Add a new ValueChangeListener to the set of listeners interested in being notified when ValueChangeEvent{s} occur.

Object getSubmittedValue() Return the submittedValue value of this component.

MethodBinding getValidator() Deprecated. getValidators() should be used instead.

Validator[] getValidators() Return the set of registered Validators for this component instance.

MethodBinding getValueChangeListener() Deprecated. Use getValueChangeListeners() instead.

ValueChangeListener[] getValueChangeListeners() Return the set of registered ValueChangeListener component instance.

boolean isImmediate() Return the "immediate" state for this component.

boolean isLocalValueSet() Return the "local value set" state for this component.

boolean isRequired() Return the "required field" state for this component.

boolean isValid() Return a flag indicating whether the local value of this component is valid (no conversion error has occurred).

void removeValidator(Validator validator) Remove a Validator instance from the set associated with this component, if it was previously associated.

void removeValueChangedListener(ValueChangeListener listener) Remove an existing ValueChangeListener (if any) from the set of listeners interested in being notified when ValueChangeEvent{s} occur.

void setImmediate(boolean immediate) Set the "immediate" state for this component.

void setLocalValueSet(boolean localValueSet) Sets the "local value set" state for this component.

void setRequired(boolean required)
Set the "required field" state for this component.

```java
void setSubmittedValue(Object submittedValue)
```

Set the submittedValue value of this component.

```java
void setValid(boolean valid)
```

Set a flag indicating whether the local value of the component is valid (no conversion error has occurred).

```java
void setValidator(MethodBinding validatorBinding)
```

Deprecated. Use `addValidator(javax.faces.validator.Validator)` instead, obtaining the argument `Validator` by creating an instance of `MethodExpressionValidator`.

```java
void setValueChangeListener(MethodBinding valueChangeMethod)
```

Deprecated. Use `addValueChangeListener(javax.faces.event.ValueChangeListener)` instead, obtaining the argument `ValueChangeListener` by creating an instance of `MethodExpressionValueChangeListener`.

Methods inherited from interface `javax.faces.component.ValueHolder`

- `getConverter`, `getLocalValue`, `getValue`, `setConverter`, `setValue`

### Method Detail

#### public Object getSubmittedValue()

```java
getSubmittedValue Renderer encodeBegin() /
encodeEnd()
```

getSubmittedValue

```java
Object getSubmittedValue()
```

Return the submittedValue value of this component. This method
should only be used by the `encodeBegin()` and/or `encodeEnd()` methods of this component, or its corresponding `Renderer`.

---

**public void setSubmittedValue(Object submittedValue)**

submittedValue

*Renderer* `decode()` `validate()`

---

**setSubmittedValue**

```java
void setSubmittedValue(Object submittedValue)
```

Set the `submittedValue` value of this component. This method should only be used by the `decode()` and `validate()` method of this component, or its corresponding `Renderer`.

**Parameters:**
- `submittedValue` - The new submitted value

---

**public boolean isLocalValueSet()**

"local value set" `setValue()` `true`

---

**isLocalValueSet**

```java
boolean isLocalValueSet()
```

Return the "local value set" state for this component. Calls to `setValue()` automatically reset this property to `true`.

---

**public void setLocalValueSet(boolean localValueSet)**
"local value set"

**setLocalValueSet**

```java
void setLocalValueSet(boolean localValueSet)
```

Sets the "local value set" state for this component.

---

**isValid**

```java
boolean isValid()
```

Return a flag indicating whether the local value of this component is valid (no conversion error has occurred).

---

**setValid**

```java
void setValid(boolean valid)
```

Set a flag indicating whether the local value of this component is valid (no conversion error has occurred).
Parameters:
valid - The new valid flag

public boolean isRequired()

"required field"

isRequired

boolean isRequired()

Return the "required field" state for this component.

public void setRequired(boolean required)

"required field"

required "required field"

setRequired

void setRequired(boolean required)

Set the "required field" state for this component.

Parameters:
required - The new "required field" state

public boolean isImmediate()

"immediate"
**isImmediate**

boolean isImmediate()

    Return the "immediate" state for this component.

---

**public void setImmediate(boolean immediate)**

"immediate" true ValueChangeEvent
false

    immediate "immediate"

**setImmediate**

void setImmediate(boolean immediate)

    Set the "immediate" state for this component. When set to true, the component's value will be converted and validated immediately in the Apply Request Values phase, and ValueChangeEvent's will be delivered in that phase as well. The default value for this property must be false.

**Parameters:**

    immediate - The new "immediate" state

---

**public MethodBinding getValidator()**

    #setValidator null #setValidator #setValidator MethodBinding

    immediate
getValidator

MethodBinding getValidator()

Deprecated. getValidators() should be used instead.

If setValidator(javax.faces.el.MethodBinding) was not previously called for this instance, this method must return null. If it was called, this method must return the exact MethodBinding instance that was passed to setValidator(javax.faces.el.MethodBinding).

This method will be called during the Process Validations or Apply Request Values phases (depending on the value of the immediate property).

public void setValidator(MethodBinding validatorBinding)

javax.faces.validator.Validator validatorBinding

#getValidators setValidator

immediate

void javax.faces.context.FacesContext UIComponent Object

validatorBinding MethodBinding

#addValidator

deprecated javax.faces.validator.MethodExpressionValidator Validator

setValidator

void setValidator(MethodBinding validatorBinding)
Deprecated. Use `addValidator(javax.faces.validator.Validator)` instead, obtaining the argument `Validator` by creating an instance of `MethodExpressionValidator`.

Wrap the argument `validatorBinding` in an implementation of `Validator` and store it in the internal data structure that backs the `getValidators()` method, taking care to over-write any instance that was stored by a previous call to `setValidator`.

The argument method will be called during the `Process Validations` or `Apply Request Values` phases (depending on the value of the `immediate` property).

Any method referenced by such an expression must be public, with a return type of `void`, and accept parameters of type `FacesContext`, `UIComponent`, and `Object`.

**Parameters:**

- `validatorBinding` - The new MethodBinding instance

```java
public MethodBinding getValueChangeListener()
```

```java
#setValueChangeListener null #setValueChangeListener
#setValueChangeListener MethodBinding

deprecated

#getValueChangeListeners
```

**getValueChangeListener**

```java
MethodBinding getValueChangeListener()
```

**Deprecated.** Use `getValueChangeListeners()` instead.

If `setValueChangeListener(javax.faces.el.MethodBinding)` was not previously called for this instance, this method must return `null`. If it was called, this method must return the exact `MethodBinding` instance that was passed to
public void setValueChangeListener(MethodBinding valueChangeMethod)

    ValueChangeListener valueChangeMethod
    #getValueChangeListeners setValueChangeListener
    immediate

    void javax.faces.event.ValueChangeEvent valueChangeMethod

    deprecated javax.faces.event.MethodExpressionValueChangeListener
    #addValueChangeListener
    ValueChangeListener

setValueChangeListener

    void setValueChangeListener(MethodBinding valueChangeMethod)

    Deprecated. Use addValueChangeListener(javax.faces.event.ValueChangeListener) instead, obtaining the argument ValueChangeListener by creating an instance of MethodExpressionValueChangeListener.

    Wrap the argument valueChangeMethod in an implementation of ValueChangeListener and store it in the internal data structure that backs the getValueChangeListeners() method, taking care to overwrite any instance that was stored by a previous call to setValueChangeListener.

    This argument method will be called during the Process Validations or Apply Request Values phases (depending on the value of the immediate property).
Any method referenced by such an expression must be public, with a return type of void, and accept a parameter of type ValueChangeEvent.

**Parameters:**
valueChangeMethod - The new method binding instance

---

```java
public void addValidator(Validator validator)
```

**Validator**

- **validator**

  **Throws**: NullPointerException: validator null

---

```java
addValidator
```

Add a Validator instance to the set associated with this component.

**Parameters:**
validator - The Validator to add

**Throws:**
NullPointerException - if validator is null

---

```java
public Validator[] getValidators()
```

**Validator**

- **getValidators**

---

```java
getValidators
```

**Validator[] getValidators**
Return the set of registered `Validator` s for this component instance. If there are no registered validators, a zero-length array is returned.

```java
public void removeValidator(Validator validator)
```

Remove a `Validator` instance from the set associated with this component, if it was previously associated. Otherwise, do nothing.

**Parameters:**
- `validator` - The `Validator` to remove

```java
public void addValueChangeListener(ValueChangeListener listener)
```

Add a new `ValueChangeListener` to the set of listeners interested in being notified when `ValueChangeEvent` s occur.

**Throws:** `NullPointerException` if `null`
Parameters:
   listener - The `ValueChangeListener` to be added

Throws:
   `NullPointerException` - if listener is null

```java
public ValueChangeListener[] getValueChangeListeners()
```

Return the set of registered `ValueChangeListener`s for this component instance. If there are no registered listeners, a zero-length array is returned.

```java
public void removeValueChangeListener(ValueChangeListener listener)
```

Remove an existing `ValueChangeListener` (if any) from the set of listeners interested in being notified when `ValueChangeEvent`s occur.
Parameters:
   listener - The ValueChangeListener to be removed

Throws:
   NullPointerException - if listener is null
javax.resource.spi  Class EISSystemException

java.lang.Object
  ▼ java.lang.Throwable
       ▼ java.lang.Exception
            ▼ javax.resource.ResourceException
                 ▼ javax.resource.spi.EISSystemException

All Implemented Interfaces:
  Serializable

public class EISSystemException
  extends ResourceException

  Extends: Throwable > Exception > ResourceException

EISSystemException  EIS  EIS  EIS
  version  1.0

An EISSystemException is used to indicate any EIS specific system-level error conditions. The common error conditions are: failure or inactivity of an EIS instance, communication failure and EIS specific error in the creation of a new physical connection.

Version:
  1.0

Author:
  Rahul Sharma, Ram Jeyaraman

See Also:
  Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>EISSystemException()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
</tbody>
</table>
### Constructor Detail

**public EISSystemException()**

null

---

### Method Summary

Methods inherited from class `javax.resource.ResourceException`

- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`

Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public EISSystemException()

Constructs a new instance with null as its detail message.

-----------------------------------------------

public EISSystemException(String message)

message

EISSystemException

public EISSystemException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.

-----------------------------------------------

public EISSystemException(Throwable cause)

cause  throwable

EISSystemException

public EISSystemException(Throwable cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

-----------------------------------------------

public EISSystemException(String message, Throwable cause)

cause  throwable
EISSystemException

public EISSystemException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public EISSystemException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.ejb** Annotation Type EJB

```java
@Target({TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface EJB
```

**Implements:** Annotation

```java
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

Enterprise Java Bean
```

Indicates a dependency on the local or remote view of an Enterprise Java Bean.

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th><code>beanInterface</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Holds one of the following interface types of the target EJB: [ Local business interface, Remote business interface, Local Home interface, Remote Home interface ]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>String</strong></th>
<th><code>beanName</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The ejb-name of the Enterprise Java Bean to which this reference is mapped.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>String</strong></th>
<th><code>description</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>String</strong></th>
<th><code>mappedName</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The product specific name of the EJB component to which this ejb reference should be mapped.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>String</strong></th>
<th><code>name</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The logical name of the ejb reference within the declaring component's (java:comp/env) environment.</td>
</tr>
</tbody>
</table>
abstract public String name()
(java:comp/env) ejb

name

public abstract String name

The logical name of the ejb reference within the declaring component's (java:comp/env) environment.

Default: 

abstract public String description()

description

public abstract String description

Default: 

abstract public String beanName()
 Enterprise Java Bean  ejb-name EJB

beanName

public abstract String beanName

The ejb-name of the Enterprise Java Bean to which this reference is mapped. Only applicable if the target EJB is defined within the same application or stand-alone module as the declaring component.
abstract public Class<T> beanInterface()
EJB [ Home Home ]

beanInterface

public abstract Class beanInterface

Holds one of the following interface types of the target EJB: [ Local business interface, Remote business interface, Local Home interface, Remote Home interface ]

Default:
java.lang.Object.class

abstract public String mappedName()
ejb EJB JNDI

mappedName

public abstract String mappedName

The product specific name of the EJB component to which this ejb reference should be mapped. This mapped name is often a global JNDI name, but may be a name of any form. Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

Default:
""
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.ejb  Class EJBAccessException

java.lang.Object  
  └ java.lang.Throwable  
    └ java.lang.Exception  
      └ java.lang.RuntimeException  
        └ javax.ejb.EJBException  
          └ javax.ejb.EJBAccessException

All Implemented Interfaces:
  Serializable

public class EJBAccessException
  extends EJBException

  Extends: Throwable > Exception > RuntimeException > EJBException

This exception indicates that client access to a business method was denied.

See Also:
 Serialized Form

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJBAccessException()</td>
</tr>
<tr>
<td>Constructs an EJBAccessException with no detail message.</td>
</tr>
<tr>
<td>EJBAccessException(String message)</td>
</tr>
<tr>
<td>Constructs an EJBAccessException with the specified detailed message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>javax.ejb.EJBException</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getCausedByException</code>, <code>getMessage</code>, <code>printStackTrace</code>, <code>printStackTrace</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>java.lang.Throwable</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fillInStackTrace</code>, <code>getCause</code>, <code>getLocalizedMessage</code>, <code>getStackTrace</code>, <code>initCause</code>, <code>setStackTrace</code>, <code>toString</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>java.lang.Object</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
</tr>
</tbody>
</table>

### Constructor Detail

**public EJBAccessException()**

EJBAccessException

**public EJBAccessException(String message)**

EJBAccessException

```java
public EJBAccessException(String message)
```

Constructs an EJBAccessException with the specified detailed
javax.ejb Interface EJBContext

All Known Subinterfaces:

   EntityContext, MessageDrivenContext, SessionContext

```
public interface EJBContext

Implemented by: EntityContext, MessageDrivenContext, SessionContext
```

EJBContext Bean

SessionContextEntityContext MessageDrivenContext Bean

The EJBContext interface provides an instance with access to the container-provided runtime context of an enterprise Bean instance.

This interface is extended by the SessionContext, EntityContext, and MessageDrivenContext interfaces to provide additional methods specific to the enterprise interface Bean type.

### Method Summary

<table>
<thead>
<tr>
<th>Identity</th>
<th>getCallerIdentity()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Deprecated. Use Principal getCallerPrincipal() instead.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Principal</th>
<th>getCallerPrincipal()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obtain the java.security.Principal that identifies the caller.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EJBHome</th>
<th>getEJBHome()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obtain the enterprise bean's remote home interface.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EJBLocalHome</th>
<th>getEJBLocalHome()</th>
</tr>
</thead>
</table>
Obtain the enterprise bean's local home interface.

Properties

getEnvironment()  
Deprecated. Use the JNDI naming context java:comp/env to access enterprise bean's environment.

boolean getRollbackOnly()  
Test if the transaction has been marked for rollback only.

TimerService
getTimerService()  
Get access to the EJB Timer Service.

UserTransaction
getUserTransaction()  
Obtain the transaction demarcation interface.

boolean isCallerInRole(Identity role)  
Deprecated. Use boolean isCallerInRole(String roleName) instead.

boolean isCallerInRole(String roleName)  
Test if the caller has a given security role.

Object lookup(String name)  
Lookup a resource within the component's private naming context.

void setRollbackOnly()  
Mark the current transaction for rollback.

Method Detail

public EJBHome getEJBHome()
Bean home
    return Bean home
    Throws IllegalStateException: Bean home

getEJBHome

EJBHome getEJBHome()
Obtain the enterprise bean's remote home interface.

**Returns:**
The enterprise bean's remote home interface.

**Throws:**
*IllegalStateException* - if the enterprise bean does not have a remote home interface.

```java
public EJBLocalHome getEJBLocalHome()
```

**Bean home**

```java
return Bean home
```

**Throws**
*IllegalStateException:* Bean home

---

getEJBLocalHome

```java
EJBLocalHome getEJBLocalHome()
```

Obtain the enterprise bean's local home interface.

**Returns:**
The enterprise bean's local home interface.

**Throws:**
*IllegalStateException* - if the enterprise bean does not have a local home interface.

---

```java
public java.util.Properties getEnvironment()
```

**Bean**

```java
return Bean
```

**deprecated**
*JNDI java:comp/env Bean*
getEnvironment

Properties getEnvironment()

**Deprecated. Use the JNDI naming context java:comp/env to access enterprise bean's environment.**

Obtain the enterprise bean's environment properties.

**Note:** If the enterprise bean has no environment properties this method returns an empty java.util.Properties object. This method never returns null.

**Returns:**

The environment properties for the enterprise bean.

---

public java.security.Identity getCallerIdentity()

java.security.Identity EJB 1.1 Container
null Bean getCallerPrincipal

return Identity

deprecated Principal getCallerPrincipal()

---

getCallerIdentity

Identity getCallerIdentity()

**Deprecated. Use Principal getCallerPrincipal() instead.**

Obtain the java.security.Identity of the caller. This method is deprecated in EJB 1.1. The Container is allowed to return alway null from this method. The enterprise bean should use the getCallerPrincipal method instead.

**Returns:**

The Identity object that identifies the caller.
public java.security.Principal getCallerPrincipal()
java.security.Principal
    return Principal null
    Throws IllegalStateException: Container

getCallerPrincipal

Principal getCallerPrincipal()

Obtain the java.security.Principal that identifies the caller.

Returns:
The Principal object that identifies the caller. This method never returns null.

Throws:
IllegalStateException - The Container throws the exception if the instance is not allowed to call this method.

public boolean isCallerInRole(java.security.Identity role)

ejb 1.1 bean isCallerInRole(String roleName)

role java.security.Identity
return true
deprecated boolean isCallerInRole(String roleName)

isCallerInRole

boolean isCallerInRole(Identity role)

Deprecated. Use boolean isCallerInRole(String roleName) instead.
Test if the caller has a given role.

This method is deprecated in EJB 1.1. The enterprise bean should use the isCallerInRole(String roleName) method instead.

**Parameters:**
role - The java.security.Identity of the role to be tested.

**Returns:**
True if the caller has the specified role.

---

```java
public boolean isCallerInRole(String roleName)
```

**(roleName**

return true

**Throws**
IllegalStateException: Container

---

**isCallerInRole**

boolean isCallerInRole(String roleName)

Test if the caller has a given security role.

**Parameters:**
roleName - The name of the security role. The role must be one of the security roles that is defined in the deployment descriptor.

**Returns:**
True if the caller has the specified role.

**Throws:**
IllegalStateException - The Container throws the exception if the instance is not allowed to call this method.

---

```java
public UserTransaction getUserTransaction() throws
IllegalStateException
```

Bean Bean UserTransaction Bean

Bean Bean Bean
getUserTransaction

```java
UserTransaction getUserTransaction() throws IllegalStateException
```

Obtain the transaction demarcation interface. Only enterprise beans with bean-managed transactions are allowed to use the UserTransaction interface. As entity beans must always use container-managed transactions, only session beans or message-driven beans with bean-managed transactions are allowed to invoke this method.

**Returns:**

The UserTransaction interface that the enterprise bean instance can use for transaction demarcation.

**Throws:**

`IllegalStateException` - The Container throws the exception if the instance is not allowed to use the UserTransaction interface (i.e. the instance is of a bean with container-managed transactions).

```java
public void setRollbackOnly() throws IllegalStateException
```

Mark the current transaction for rollback. The transaction will

```java
void setRollbackOnly() throws IllegalStateException
```
become permanently marked for rollback. A transaction marked for rollback can never commit. Only enterprise beans with container-managed transactions are allowed to use this method.

**Throws:**

`IllegalStateException` - The Container throws the exception if the instance is not allowed to use this method (i.e. the instance is of a bean with bean-managed transactions).

```java
public boolean getRollbackOnly() throws
IllegalStateException
Bean
return true false
Throws
IllegalStateException: Bean Bean Container
```

**getRollbackOnly**

```java
boolean getRollbackOnly() throws IllegalStateException
```

Test if the transaction has been marked for rollback only. An enterprise bean instance can use this operation, for example, to test after an exception has been caught, whether it is fruitless to continue computation on behalf of the current transaction. Only enterprise beans with container-managed transactions are allowed to use this method.

**Returns:**

True if the current transaction is marked for rollback, false otherwise.

**Throws:**

`IllegalStateException` - The Container throws the exception if the instance is not allowed to use this method (i.e. the instance is of a bean with bean-managed transactions).
public TimerService getTimerService() throws IllegalStateException
EJB Timer Service

Throws
IllegalStateException: Bean Bean Container

getTimerService

TimerService getTimerService()
throws IllegalStateException

Get access to the EJB Timer Service.

Throws:
IllegalStateException - The Container throws the exception if the instance is not allowed to use this method (e.g. if the bean is a stateful session bean)

-----------------------------------------------

public Object lookup(String name)

name java:comp/env

Throws IllegalArgumentException: Container

lookup

Object lookup(String name)

Lookup a resource within the component's private naming context.

Parameters:
name - Name of the entry (relative to java:comp/env).

Throws:
IllegalArgumentException - The Container throws the exception if the given name does not match an entry within the component's environment.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  Class EJBException

java.lang.Object  
  ▼ java.lang.Throwable  
    ▼ java.lang.Exception  
      ▼ java.lang.RuntimeException  
        ▼ javax.ejb.EJBException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  AccessLocalException, ConcurrentAccessException, EJBAccessException, EJBTransactionRequiredException, EJBTransactionRolledbackException, NoSuchEJBException, NoSuchEntityException, NoSuchObjectLocalException, TransactionRequiredLocalException, TransactionRolledbackLocalException

public class EJBException

extends RuntimeException

Extends: Throwable > Exception > RuntimeException
Extended by: AccessLocalException, ConcurrentAccessException, EJBAccessException, EJBTransactionRequiredException, EJBTransactionRolledbackException, NoSuchEJBException, NoSuchEntityException, NoSuchObjectLocalException, TransactionRequiredLocalException, TransactionRolledbackLocalException

Bean  EJBException

The EJBException exception is thrown by an enterprise Bean instance to its container to report that the invoked business method or callback method could not be completed because of an unexpected error (e.g. the instance failed to open a database connection).
See Also:
 Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EJBException()</strong></td>
<td>Constructs an EJBException with no detail message.</td>
</tr>
<tr>
<td><strong>EJBException(Exception ex)</strong></td>
<td>Constructs an EJBException that embeds the originally thrown exception.</td>
</tr>
<tr>
<td><strong>EJBException(String message)</strong></td>
<td>Constructs an EJBException with the specified detailed message.</td>
</tr>
<tr>
<td><strong>EJBException(String message, Exception ex)</strong></td>
<td>Constructs an EJBException that embeds the originally thrown exception with the specified detail message.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getCausedByException()</strong></td>
<td>Obtain the exception that caused the EJBException being thrown.</td>
</tr>
<tr>
<td><strong>getMessage()</strong></td>
<td>Returns the detail message, including the message from the nested exception if there is one.</td>
</tr>
<tr>
<td><strong>printStackTrace()</strong></td>
<td>Prints the composite message to System.err.</td>
</tr>
<tr>
<td><strong>printStackTrace(PrintStream ps)</strong></td>
<td>Prints the composite message and the embedded stack trace to the specified stream ps.</td>
</tr>
<tr>
<td><strong>printStackTrace(PrintWriter pw)</strong></td>
<td>Prints the composite message and the embedded stack trace to the specified print writer pw.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Throwable
**fillInStackTrace**, **getCause**, **getLocalizedMessage**, **getStackTrace**, **toString**
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public EJBException()
EJBException

EJBException
public EJBException()

Constructs an EJBException with no detail message.

public EJBException(String message)
EJBException

EJBException
public EJBException(String message)

Constructs an EJBException with the specified detailed message.

public EJBException(Exception ex)
EJBException

EJBException
public EJBException(Exception ex)

Constructs an EJBException that embeds the originally thrown exception.

public EJBException(String message, Exception ex)

EJBException

Constructs an EJBException that embeds the originally thrown exception with the specified detail message.

Method Detail

public Exception getCausedByException()

EJBException

gCausedByException

public Exception getCausedByException()

Obtain the exception that caused the EJBException being thrown.

public String getMessage()

getMessage
public String getMessage()

    Returns the detail message, including the message from the nested exception if there is one.

    Overrides:
            getMessage in class Throwable

public void printStackTrace(java.io.PrintStream ps)
    ps

printStackTrace

public void printStackTrace(PrintStream ps)

    Prints the composite message and the embedded stack trace to the specified stream ps.

    Overrides:
            printStackTrace in class Throwable

Parameters:
            ps - the print stream

public void printStackTrace()
            System.err

printStackTrace

public void printStackTrace()

    Prints the composite message to System.err.

    Overrides:
            printStackTrace in class Throwable
public void printStackTrace(java.io.PrintWriter pw)
        pw

printStackTrace

public void printStackTrace(PrintWriter pw)

    Prints the composite message and the embedded stack trace to the specified print writer pw.

Overrides:
    printStackTrace in class Throwable

Parameters:
    pw - the print writer

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Interface EJBHome

All Superinterfaces:
Remote

All Known Subinterfaces:
ManagementHome

public interface EJBHome
extends Remote

Implements: java.rmi.Remote
Implemented by: ManagementHome

EJBHome Bean home Bean home EJB
Bean home Bean home

home Bean Bean

The EJBHome interface must be extended by all enterprise Beans' remote home interfaces. An enterprise Bean's remote home interface defines the methods that allow a remote client to create, find, and remove EJB objects, as well as home business methods that are not specific to a bean instance (Session Beans do not have finders and home methods).

The remote home interface is defined by the enterprise Bean provider and implemented by the enterprise Bean container.

Method Summary

<table>
<thead>
<tr>
<th>EJBMetaData</th>
<th>getEJBMetaData()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obtain the EJBMetaData interface for the enterprise Bean.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>getHomeHandle()</code></td>
<td>Obtain a handle for the remote home object.</td>
</tr>
<tr>
<td><code>void remove(Handle handle)</code></td>
<td>Remove an EJB object identified by its handle.</td>
</tr>
<tr>
<td><code>void remove(Object primaryKey)</code></td>
<td>Remove an EJB object identified by its primary key.</td>
</tr>
</tbody>
</table>

## Method Detail

```java
public void remove(Handle handle) throws java.rmi.RemoteException, RemoveException
EJB
```

**Throws:**
- `RemoveException`: Bean
- `java.rmi.RemoteException`: 

```java
void remove(Handle handle) throws RemoteException, RemoveException
```

Remove an EJB object identified by its handle.

**Throws:**
- `RemoveException`: Thrown if the enterprise Bean or the container does not allow the client to remove the object.
- `RemoteException`: Thrown when the method failed due to a system-level failure.

```java
public void remove(Object primaryKey) throws java.rmi.RemoteException, RemoveException
EJB
```


Bean Bean  RemoveException
  Throws  RemoveException:  Bean
  Throws  java.rmi.RemoteException:

remove

void remove(Object primaryKey)
  throws RemoteException,
  RemoveException

Remove an EJB object identified by its primary key.

This method can be used only for an entity bean. An attempt to call
this method on a session bean will result in a RemoveException.

Throws:
  RemoveException - Thrown if the enterprise Bean or the container
does not allow the client to remove the object.
  RemoteException - Thrown when the method failed due to a
system-level failure.

public EJBMetaData getEJBMetaData() throws
java.rmi.RemoteException
Bean  EJBMetaData EJBMetaData  Bean

EJBMetaData
  return  Bean  EJBMetaData
  Throws  java.rmi.RemoteException:

gEtEJBMetaData

EJBMetaData  getEJBMetaData()
  throws  RemoteException
Obtain the EJBMetaData interface for the enterprise Bean. The EJBMetaData interface allows the client to obtain information about the enterprise Bean.

The information obtainable via the EJBMetaData interface is intended to be used by tools.

**Returns:**

The enterprise Bean's EJBMetaData interface.

**Throws:**

RemoteException - Thrown when the method failed due to a system-level failure.

---

```java
public HomeHandle getHomeHandle() throws java.rmi.RemoteException

home home Java
```

**getHomeHandle**

HomeHandle getHomeHandle()

Throws RemoteException:

Obtain a handle for the remote home object. The handle can be used at later time to re-obtain a reference to the remote home object, possibly in a different Java Virtual Machine.

**Returns:**

A handle for the remote home object.

**Throws:**

RemoteException - Thrown when the method failed due to a system-level failure.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.ejb** Interface **EJBLocalHome**

```java
public interface EJBLocalHome

EJBLocalHome Bean home Bean home EJB
Bean home Bean home

home Bean Bean
```

The EJBLocalHome interface must be extended by all enterprise Beans' local home interfaces. An enterprise Bean's local home interface defines the methods that allow local clients to create, find, and remove EJB objects, as well as home business methods that are not specific to a bean instance (session Beans do not have finders and home business methods).

The local home interface is defined by the enterprise Bean provider and implemented by the enterprise Bean container.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>remove(Object primaryKey)</code></td>
<td>Remove an EJB object identified by its primary key.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void remove(Object primaryKey) throws RemoveException, EJBException
EJB
```
Bean Bean RemoveException

Throws  RemoveException:  Bean

Throws  EJBEexception:

remove

void remove(Object primaryKey)
throws RemoveException,
EJBEexception

Remove an EJB object identified by its primary key.

This method can only be used by local clients of an entity bean. An attempt to call this method on a session bean will result in a RemoveException.

Throws:

RemoveException - Thrown if the enterprise Bean or the container does not allow the client to remove the object.
EJBEexception - Thrown when the method failed due to a system-level failure.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms,
<table>
<thead>
<tr>
<th>SUMMARY</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES |
public interface EJBLocalObject

EJBLocalObject Bean Bean EJB Bean
Bean Bean Bean

The EJBLocalObject interface must be extended by all enterprise Beans' local interfaces. An enterprise Bean's local interface provides the local client view of an EJB object. An enterprise Bean's local interface defines the business methods callable by local clients.

The enterprise Bean's local interface is defined by the enterprise Bean provider and implemented by the enterprise Bean container.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJBLocalHome</td>
<td>Obtain the enterprise Bean's local home interface.</td>
</tr>
<tr>
<td>Object</td>
<td>Obtain the primary key of the EJB local object.</td>
</tr>
<tr>
<td>isIdentical(EJBLocalObject obj)</td>
<td>Test if a given EJB local object is identical to the invoked EJB local object.</td>
</tr>
<tr>
<td>remove()</td>
<td>Remove the EJB local object.</td>
</tr>
</tbody>
</table>

### Method Detail
public EJBLocalHome getEJBLocalHome() throws EJBException

Bean home home Bean createfinder
remove home

  return Bean home
  Throws EJBException:

getEJBLocalHome

EJBLocalHome getEJBLocalHome()
  throws EJBException

Obtain the enterprise Bean's local home interface. The local home interface defines the enterprise Bean's create, finder, remove, and home business methods that are available to local clients.

Returns:
  A reference to the enterprise Bean's local home interface.

Throws:
  EJBException - Thrown when the method failed due to a system-level failure.

public Object getPrimaryKey() throws EJBException

Bean Bean EJBException

  return EJB
  Throws EJBException: Bean

getPrimaryKey

Object getPrimaryKey()
  throws EJBException
Obtain the primary key of the EJB local object.

This method can be called on an entity bean. An attempt to invoke this method on a session Bean will result in an EJBException.

**Returns:**
The EJB local object's primary key.
**Throws:**
- `EJBException` - Thrown when the method failed due to a system-level failure or when invoked on a session bean.

```java
public void remove() throws RemoveException, EJBException
```

Remove the EJB local object.

**Throws:**
- `RemoveException` - The enterprise Bean or the container does not allow destruction of the object.
- `EJBException` - Thrown when the method failed due to a system-level failure.

```java
public boolean isIdentical(EJBLocalObject obj) throws EJBException
```

```java
void remove()
 throws RemoveException, EJBException
```

```java
void remove()
 throws RemoveException, EJBException
```
boolean isIdentical(EJBLocalObject obj) throws EJBException

Test if a given EJB local object is identical to the invoked EJB local object.

**Parameters:**

obj - An object to test for identity with the invoked object.

**Returns:**

True if the given EJB local object is identical to the invoked object, false otherwise.

**Throws:**

EJBException - Thrown when the method failed due to a system-level failure.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
</tr>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The EJBMetaData interface allows a client to obtain the enterprise Bean's meta-data information.

The meta-data is intended for development tools used for building applications that use deployed enterprise Beans, and for clients using a scripting language to access the enterprise Bean.

Note that the EJBMetaData is not a remote interface. The class that implements this interface (this class is typically generated by container tools) must be serializable, and must be a valid RMI/IDL value type.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EJBHome</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
</tbody>
</table>
Class

Obtain the Class object for the enterprise Bean's remote interface.

boolean isSession()

Test if the enterprise Bean's type is "session".

boolean isStatelessSession()

Test if the enterprise Bean's type is "stateless session".

Method Detail

public EJBHome getEJBHome()
Bean home
return Bean home

getEJBHome

EJBHome getEJBHome()

Obtain the remote home interface of the enterprise Bean.

Returns:
the remote home interface of the enterprise Bean.

public Class<T> getHomeInterfaceClass()
Bean home Class
return Bean home Class

getHomeInterfaceClass

Class getHomeInterfaceClass()

Obtain the Class object for the enterprise Bean's remote home interface.
Returns:
the Class object for the enterprise Bean's remote home interface.

public Class<T> getRemoteInterfaceClass()
Bean Class
return Bean Class

getRemoteInterfaceClass

Class getRemoteInterfaceClass()

Obtain the Class object for the enterprise Bean's remote interface.

Returns:
the Class object for the enterprise Bean's remote interface.

public Class<T> getPrimaryKeyClass()
Bean Class
return Bean Class

getPrimaryKeyClass

Class getPrimaryKeyClass()

Obtain the Class object for the enterprise Bean's primary key class.

Returns:
the Class object for the enterprise Bean's primary key class.

public boolean isSession()
Bean “”
isSession

boolean isSession()

Test if the enterprise Bean's type is "session".

Returns:
    True if the type of the enterprise Bean is session bean.

isStatelessSession

boolean isStatelessSession()

Test if the enterprise Bean's type is "stateless session".

Returns:
    True if the type of the enterprise Bean is stateless session.
javax.security.jacc  **Class EJBMethodPermission**

**java.lang.Object**
- **java.security.Permission**
  - **javax.security.jacc.EJBMethodPermission**

**All Implemented Interfaces:**
- **Serializable, Guard**

```java
public final class EJBMethodPermission
extends Permission
implements Serializable

Extends: java.security.Permission
Implements: java.io.Serializable

EJB

EJBMethodPermission EJB ejb-name

EJBMethodPermission actions EJB

newPermissionCollection

See also java.security.Permission

Class for EJB method permissions.

The name of an EJBMethodPermission contains the value of the ejb-name element in the application's deployment descriptor that identifies the target EJB.

The actions of an EJBMethodPermission identifies the methods of the EJB to which the permission applies.

Implementations of this class MAY implement newPermissionCollection
or inherit its implementation from the super class.

**Author:**
Ron Monzillo, Gary Ellison

**See Also:**
Permission, Serialized Form

---

### Constructor Summary

**EJBMethodPermission(String name, String actions)**

Creates a new EJBMethodPermission with the specified name and actions.

**EJBMethodPermission(String EJBName, String methodInterface, Method method)**

Creates a new EJBMethodPermission with name corresponding to the EJBName and actions composed from methodInterface, and the Method object.

**EJBMethodPermission(String EJBName, String methodName, String methodInterface, String[] methodParams)**

Creates a new EJBMethodPermission with name corresponding to the EJBName and actions composed from methodName, methodInterface, and methodParams.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>equals(Object o)</strong></td>
</tr>
<tr>
<td>Checks two EJBMethodPermission objects for equality.</td>
</tr>
<tr>
<td><strong>getActions()</strong></td>
</tr>
<tr>
<td>Returns a String containing a canonical representation of the actions of this EJBMethodPermission.</td>
</tr>
<tr>
<td><strong>hashCode()</strong></td>
</tr>
<tr>
<td>Returns the hash code value for this EJBMethodPermission.</td>
</tr>
<tr>
<td><strong>implies(Permission permission)</strong></td>
</tr>
<tr>
<td>Determines if the argument Permission is &quot;implied by&quot; this EJBMethodPermission.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.security.

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkGuard</td>
</tr>
<tr>
<td>getName</td>
</tr>
<tr>
<td>newPermissionCollection</td>
</tr>
<tr>
<td>toString</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone</td>
</tr>
<tr>
<td>finalize</td>
</tr>
<tr>
<td>getClass</td>
</tr>
<tr>
<td>notify</td>
</tr>
<tr>
<td>notifyAll</td>
</tr>
<tr>
<td>wait</td>
</tr>
<tr>
<td>wait</td>
</tr>
<tr>
<td>wait</td>
</tr>
</tbody>
</table>

Constructor Detail

```java
public EJBMethodPermission(String name, String actions)
EJBMethodPermission

EJB  ejb-name

actions  methodSpecactions

methodNameSpec ::= methodName | emptyString

methodInterfaceName ::= String

methodInterfaceSpec ::= methodInterfaceName | emptyString

typeName ::= typeName | typeName []

methodParams ::= typeName | methodParams comma typeName

methodParamsSpec ::= emptyString | methodParams

methodSpec ::= null | methodNameSpec |
methodSpec ::= methodNameSpec comma methodInterfaceName |
methodSpec ::= methodNameSpec comma methodInterfaceSpec comma methodParamsSpec |
```

MethodInterfaceName  String  EJB  method-intf
EJB  EJB  method-intf
EJBMethodPermission

```java
class EJBMethodPermission {
    public EJBMethodPermission(String name, String actions) {
        // Implementation...
    }
}
```

Creates a new EJBMethodPermission with the specified name and actions.

The name contains the value of the ejb-name element corresponding to an EJB in the application's deployment descriptor.

The actions contains a methodSpec. The syntax of the actions
parameter is defined as follows:

```
methodNameSpec ::= methodName | emptyString
methodInterfaceName ::= String
methodInterfaceSpec ::= methodInterfaceName | emptyString
typeName ::= typeName | typeName []
methodParams ::= typeName | methodParams comma typeName
methodParamsSpec ::= emptyString | methodParams
methodSpec ::= null |
    methodNameSpec |
    methodNameSpec comma methodInterfaceName |
    methodNameSpec comma methodInterfaceSpec comma methodParamsSpec
```

A MethodInterfaceName is a non-empty String and should contain a method-intf value as defined for use in EJB deployment descriptors. An implementation must be flexible such that it supports additional interface names especially if they are standardized by the EJB Specification. The EJB Specification currently defines the following method-intf values:

```
{ "Home", "LocalHome", "Remote", "Local", "ServiceEndpoint"
```

A null or empty string methodSpec indicates that the permission applies to all methods of the EJB. A methodSpec with a methodNameSpec of the empty string matches all methods of the EJB that match the methodInterface and methodParams elements of the methodSpec.

A methodSpec with a methodInterfaceSpec of the empty string matches all methods of the EJB that match the methodNameSpec and methodParamsSpec elements of the methodSpec.

A methodSpec without a methodParamsSpec matches all methods of the EJB that match the methodNameSpec and methodInterface elements of the methodSpec.
The order of the typeNames in methodParams array must match the order of occurrence of the corresponding parameters in the method signature of the target method(s). Each typeName in the methodParams must contain the canonical form of the corresponding parameter’s typeName as defined by the getActions method. A methodSpec with an empty methodParamsSpec matches all 0 argument methods of the EJB that match the methodNameSpec and methodInterfaceSpec elements of the methodSpec.

**Parameters:**
- `name` - of the EJB to which the permission pertains.
- `actions` - identifies the methods of the EJB to which the permission pertains.

```java
public EJBMethodPermission(String EJBName, String methodName, String methodInterface, String[] methodParams)
```

- `EJBName` - `EJBNamemethodName methodInterface methodParams actions EJBMethodPermission`
- `methodName` - `EJB name``
- `methodInterface` - `EJB null ***`
- `methodParams` - `typeName methodParams null`

**EJBMethodPermission**

```java
public EJBMethodPermission(String EJBName, String methodName,
                          String methodInterface,
                          String[] methodParams)
```
Creates a new EJBMethodPermission with name corresponding to the EJBName and actions composed from methodName, methodInterface, and methodParams.

Parameters:

EJBName - The string representation of the name of the EJB as it appears in the corresponding ejb-name element in the deployment descriptor.

methodName - A string that may be used to indicate the method of the EJB to which the permission pertains. A value of null or "" indicates that the permission pertains to all methods that match the other parameters of the permission specification without consideration of method name.

methodInterface - A string that may be used to specify the EJB interface to which the permission pertains. A value of null or "" indicates that the permission pertains to all methods that match the other parameters of the permission specification without consideration of the interface they occur on.

methodParams - An array of strings that may be used to specify (by typeNames) the parameter signature of the target methods. The order of the typeNames in methodParams array must match the order of occurrence of the corresponding parameters in the method signature of the target method(s). Each typeName in the methodParams array must contain the canonical form of the corresponding parameter's typeName as defined by the getActions method. An empty methodParams array is used to represent a method signature with no arguments. A value of null indicates that the permission pertains to all methods that match the other parameters of the permission specification without consideration of method signature.

public EJBMethodPermission(String EJBName, String methodInterface, Method method)
EJBMethodPermission

public EJBMethodPermission(String EJBName, String methodInterface, Method method)

Creates a new EJBMethodPermission with name corresponding to the EJBName and actions composed from methodInterface, and the Method object.

A container uses this constructor prior to checking if a caller has permission to call the method of an EJB.

Parameters:

EJBName - The string representation of the name of the EJB as it appears in the corresponding ejb-name element in the deployment descriptor.

methodInterface - A string that may be used to specify the EJB interface to which the permission pertains. A value of null or """, indicates that the permission pertains to all methods that match the other parameters of the permission specification without consideration of the interface they occur on.

method - an instance of the Java.lang.reflect.Method class corresponding to the method that the container is trying to
determine whether the caller has permission to access. This value must not be null.

### Method Detail

**public boolean equals(Object o)**

EJBMethodPermission EJBMethodPermission

**actions**

P1.implies(P2) && P2.implies(P1) Permission P1 P2

```
o EJBMethodPermission EJBMethodPermission
```

```
return EJBMethodPermission EJBMethodPermission
```

true

**equals**

**public boolean equals(Object o)**

Checks two EJBMethodPermission objects for equality. EJBMethodPermission objects are equivalent if they have case sensitive equivalent name and actions values.

Two Permission objects, P1 and P2, are equivalent if and only if P1.implies(P2) && P2.implies(P1).

**Specified by:**

equals in class Permission

**Parameters:**

- **o** - the EJBMethodPermission object being tested for equality with this EJBMethodPermission

**Returns:**
true if the argument EJBMethodPermission object is equivalent to this EJBMethodPermission.

public String getActions()
  EJBMethodPermission actions String
EJBMethodPermission actions

methodNameSpec ::= methodName | emptyString

methodInterfaceName ::= String

methodInterfaceSpec ::= methodInterfaceName | emptyString

typeName ::= typeName | typeName []

methodParams ::= typeName | methodParams comma typeName

methodParamsSpec ::= emptyString | methodParams

methodSpec ::= null |
  methodName |
  methodNameSpec comma methodInterfaceName |
  methodNameSpec comma methodInterfaceSpec comma methodParamsSpec

typeName Java typeName Java "[]"

MethodInterfaceName String EJB method-intf
EJB EJB method-intf

{ "Home", "LocalHome", "Remote", "Local", "ServiceEndpoint" :

  return EJBMethodPermission actions String
**getActions**

```java
public String getActions()
```

Returns a `String` containing a canonical representation of the actions of this `EJBMethodPermission`. The Canonical form of the actions of an `EJBMethodPermission` is described by the following syntax description.

```
methodNameSpec ::= methodName | emptyString
methodInterfaceName ::= String
methodInterfaceSpec ::= methodInterfaceName | emptyString
typeName ::= typeName | typeName []
methodParams ::= typeName | methodParams comma typeName
methodParamsSpec ::= emptyString | methodParams
methodSpec ::= null |
    methodName |
    methodNameSpec comma methodInterfaceName |
    methodNameSpec comma methodInterfaceSpec comma methodParamsSpec
```

The canonical form of each `typeName` must begin with the fully qualified Java name of the corresponding parameter's type. The canonical form of a `typeName` for an array parameter is the fully qualified Java name of the array's component type followed by as many instances of the string "[]" as there are dimensions to the array. No additional characters (e.g. blanks) may occur in the canonical form.

A `MethodInterfaceName` is a non-empty `String` and should contain a `method-intf` value as defined for use in EJB deployment descriptors. An implementation must be flexible such that it supports additional interface names especially if they are standardized by the EJB Specification. The EJB Specification currently defines the following `method-intf` values:

```
{ "Home", "LocalHome", "Remote", "Local", "ServiceEndpoint" }
```
Specified by:
getActions in class Permission

Returns:
a String containing the canonicalized actions of this EJBMethodPermission.

public int hashCode()
EJBMethodPermission

- Java EJBMethodPermission hashCode
  EJBMethodPermission hashCode
- equals EJBMethodPermission
  Permission hashCode

  return

hashCode

public int hashCode()

Returns the hash code value for this EJBMethodPermission. The properties of the returned hash code must be as follows:

- During the lifetime of a Java application, the hashCode method must return the same integer value every time it is called on a EJBMethodPermission object. The value returned by hashCode for a particular EJBMethodPermission need not remain consistent from one execution of an application to another.
- If two EJBMethodPermission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result (within an application).
public boolean implies(java.security.Permission permission)
EJBMethodPermission "" Permission

- EJBMethodPermission
- EJBMethodPermission
- Permission actions
  EJBMethodPermission actions

Permission

- methodNameSpec null Permission
- methodNameSpec null Permission
- methodNameSpec null permission

actions

    permission ""EJBMethodPermission Permission
    return true false

implies

public boolean implies(Permission permission)

Determines if the argument Permission is "implied by" this EJBMethodPermission. For this to be the case,

- The argument must be an instance of EJBMethodPermission
with name equivalent to that of this EJBMethoMethodPermission, and
the methods to which the argument permission applies (as defined in its actions) must be a subset of the methods to which this EJBMethoMethodPermission applies (as defined in its actions).

The argument permission applies to a subset of the methods to which this permission applies if all of the following conditions are met.

- the method name component of the methodNameSpec of this permission is null, the empty string, or equivalent to the method name of the argument permission, and
- the method interface component of the methodNameSpec of this permission is null, the empty string, or equivalent to the method interface of the argument permission, and
- the method parameter list component of the methodNameSpec of this permission is null, the empty string, or equivalent to the method parameter list of the argument permission.

The name and actions comparisons described above are case sensitive.

**Specified by:**
implies in class Permission

**Parameters:**
permission - "this" EJBMethoMethodPermission is checked to see if it implies the argument permission.

**Returns:**
true if the specified permission is implied by this object, false if not.
PS:
javax.ejb Interface EJBObject

All Superinterfaces:
  Remote

All Known Subinterfaces:
  Management

public interface EJBObject
  extends Remote

Implements: java.rmi.Remote
Implemented by: Management

EJBObject Bean Bean EJB Bean
javax.ejb.EJBObject Bean
Bean Bean Bean

The EJBObject interface is extended by all enterprise Beans' remote interfaces. An enterprise Bean's remote interface provides the remote client view of an EJB object. An enterprise Bean's remote interface defines the business methods callable by a remote client.

The remote interface must extend the javax.ejb.EJBObject interface, and define the enterprise Bean specific business methods.

The enterprise Bean's remote interface is defined by the enterprise Bean provider and implemented by the enterprise Bean container.

Method Summary

| EJBHome | getEJBHome() |
Obtain the enterprise Bean's remote home interface.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getHandle</strong></td>
<td>Obtain a handle for the EJB object.</td>
</tr>
<tr>
<td><strong>getPrimaryKey</strong></td>
<td>Obtain the primary key of the EJB object.</td>
</tr>
<tr>
<td><strong>isIdentical(EJBObject obj)</strong></td>
<td>Test if a given EJB object is identical to the invoked EJB object.</td>
</tr>
<tr>
<td><strong>remove()</strong></td>
<td>Remove the EJB object.</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public EJBHome getEJBHome() throws java.rmi.RemoteException

Bean home home Bean createfinder remove home

return Bean home

Throws java.rmi.RemoteException:
```

**getEJBHome**

```java
EJBHome getEJBHome() 

throws RemoteException
```

Obtain the enterprise Bean's remote home interface. The remote home interface defines the enterprise Bean's create, finder, remove, and home business methods.

**Returns:**
A reference to the enterprise Bean's home interface.

**Throws:**
RemoteException - Thrown when the method failed due to a system-level failure.
public Object getPrimaryKey() throws java.rmi.RemoteException
EJB

Bean Bean RemoteException
return EJB
Throws java.rmi.RemoteException: Bean

getByPrimaryKey

Object getPrimaryKey()
throws RemoteException

Obtain the primary key of the EJB object.

This method can be called on an entity bean. An attempt to invoke this method on a session bean will result in RemoteException.

Returns:
The EJB object's primary key.

Throws:
RemoteException - Thrown when the method failed due to a system-level failure or when invoked on a session bean.

public void remove() throws java.rmi.RemoteException, RemoveException
EJB

Throws java.rmi.RemoteException:
Throws RemoveException: Bean

remove
void remove()
    throws RemoteException,
        RemoveException

Remove the EJB object.

**Throws:**

- **RemoteException** - Thrown when the method failed due to a system-level failure.
- **RemoveException** - The enterprise Bean or the container does not allow destruction of the object.

---

```java
public Handle getHandle() throws java.rmi.RemoteException

EJB EJB Java
return EJB
```

Obtain a handle for the EJB object. The handle can be used at later time to re-obtain a reference to the EJB object, possibly in a different Java Virtual Machine.

**Returns:**

A handle for the EJB object.

**Throws:**

- **RemoteException** - Thrown when the method failed due to a system-level failure.

---

```java
public boolean isIdentical(EJBObject obj) throws java.rmi.RemoteException

EJB EJB
```
isIdentical

boolean isIdentical(EJBObject obj)
throws RemoteException

Test if a given EJB object is identical to the invoked EJB object.

Parameters:
   obj - An object to test for identity with the invoked object.

Returns:
   True if the given EJB object is identical to the invoked object, false otherwise.

Throws:
   RemoteException - Thrown when the method failed due to a system-level failure.
javax.security.jacc  **Class EJBRoleRefPermission**

`java.lang.Object`  
`java.security.Permission`  
`javax.security.jacc.EJBRoleRefPermission`

**All Implemented Interfaces:**  
`Serializable`, `Guard`

```java
public final class EJBRoleRefPermission

extends Permission
implements Serializable

Extends: java.security.Permission
Implements: java.io.Serializable

EJB isCallerInRole (String reference) EJBRoleRefPermission

EJBRoleRefPermission  EJB ejb-name

EJBRoleRefPermission  actions  EJBRoleRefPermission

newPermissionCollection

See also  java.security.Permission
```

Class for EJB *isCallerInRole (String reference)* permissions. An EJBRoleRefPermission is a named permission and has actions.

The name of an EJBRoleRefPermission contains the value of the ejb-name element in the application’s deployment descriptor that identifies the EJB in whose context the permission is being evaluated.

The actions of an EJBRoleRefPermission identifies the role reference to which the permission applies. An EJBRoleRefPermission is checked to
determine if the subject is a member of the role identified by the reference.

Implementations of this class MAY implement newPermissionCollection or inherit its implementation from the super class.

**Author:**
Ron Monzillo, Gary Ellison

**See Also:**
Permission, Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EJBRoleRefPermission(String name, String actions)</code></td>
<td>Creates a new EJBRoleRefPermission with the specified name and actions.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(Object o)</code></td>
<td>Checks two EJBRoleRefPermission objects for equality.</td>
</tr>
<tr>
<td><code>getActions()</code></td>
<td>Returns a canonical String representation of the actions of this EJBRoleRefPermission.</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td>Returns the hash code value for this EJBRoleRefPermission.</td>
</tr>
<tr>
<td><code>implies(Permission permission)</code></td>
<td>Determines if the argument Permission is &quot;implied by&quot; this EJBRoleRefPermission.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.security.Permission:
- `checkGuard`, `getName`, `newPermissionCollection`, `toString`

Methods inherited from class java.lang.Object:
- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
Constructor Detail

public EJBRoleRefPermission(String name, String actions)

Creates a new EJBRoleRefPermission with the specified name and actions.

Parameters:

name - the ejb-name that identifies the EJB in whose context the role references are to be evaluated.

actions - identifies the role reference to which the permission pertains. The role reference is scoped to the EJB identified in the name parameter. The value of the role reference must not be null or the empty string.

Method Detail

public boolean equals(Object o)

P1.implies(P2) && P2.implies(P1) Permission P1 P2
equals

public boolean equals(Object o)

Checks two EJBRoleRefPermission objects for equality. EJBRoleRefPermission objects are equivalent if they have case equivalent name and actions values.

Two Permission objects, P1 and P2, are equivalent if and only if P1.implies(P2) && P2.implies(P1).

Specified by:
   equals in class Permission

Parameters:
   o - the EJBRoleRefPermission object being tested for equality with this EJBRoleRefPermission.

Returns:
   true if the argument EJBRoleRefPermission object is equivalent to this EJBRoleRefPermission.

public String getActions()
EJBRoleRefPermission actions String

return EJBRoleRefPermission actions String

getActions

public String getActions()
Returns a canonical String representation of the actions of this EJBRoleRefPermission.

**Specified by:**
getActions in class Permission

**Returns:**
a String containing the canonicalized actions of this EJBRoleRefPermission.

---

```java
public int hashCode()
EJBRoleRefPermission
```

- Java EJBRoleRefPermission hashCode
  EJBRoleRefPermission hashCode
- equals EJBRoleRefPermission
  Permission hashCode

  return

---

**hashCode**

```java
public int hashCode()
```

Returns the hash code value for this EJBRoleRefPermission. The properties of the returned hash code must be as follows:

- During the lifetime of a Java application, the hashCode method must return the same integer value, every time it is called on a EJBRoleRefPermission object. The value returned by hashCode for a particular EJBRoleRefPermission need not remain consistent from one execution of an application to another.
- If two EJBRoleRefPermission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result (within an application).
public boolean implies(java.security.Permission permission)
EJBRoleRefPermission "" Permission

• EJBRoleRefPermission
• EJBRoleRefPermission
• EJBRoleRefPermission

actions

permission ""EJBRoleRefPermission Permission

return true false

implies

public boolean implies(Permission permission)

Determines if the argument Permission is "implied by" this EJBRoleRefPermission. For this to be the case,

• The argument must be an instance of EJBRoleRefPermission
• with name equivalent to that of this EJBRoleRefPermission, and
• with the role reference equivalent to that of this EJBRoleRefPermission applies.

The name and actions comparisons described above are case sensitive.

Specified by:
`implies` in class `Permission`

**Parameters:**
permission - "this" EJBRoleRefPermission is checked to see if it implies the argument permission.

**Returns:**
true if the specified permission is implied by this object, false if not.
javax.ejb  Annotation Type EJBs

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface EJBs

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

TYPE-level @EJB

Declares multiple TYPE-level @EJB annotations.

Required Element Summary

| EJB[] | value |

Element Detail

abstract public EJB[] value()

value

public abstract EJB[] value

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
public interface EJBStats extends Stats

Implements: Stats
Implemented by: EntityBeanStats, MessageDrivenBeanStats, SessionBeanStats, StatefulSessionBeanStats, StatelessSessionBeanStats

EJB

Specifies statistics provided by all EJB component types.

Method Summary

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of times create was called.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getRemoveCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of times remove was called.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats

gGetStatistic, getStatisticNames, getStatistics
public CountStatistic getCreateCount()
create

getCreateCount

CountStatistic getCreateCount()

Number of times create was called.

public CountStatistic getRemoveCount()
remove

getRemoveCount

CountStatistic getRemoveCount()

Number of times remove was called.
javax.ejb Class EJBTransactionRequiredException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.ejb.EJBException
                  └ javax.ejb.EJBTransactionRequiredException

All Implemented Interfaces:
  Serializable

public class EJBTransactionRequiredException

extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

null

This exception indicates that a request carried a null transaction context, but the target object requires an active transaction.

See Also:
  Serialized Form

Constructor Summary

EJBTransactionRequiredException()

Constructs an EJBTransactionRequiredException with no detail message.

EJBTransactionRequiredException(String message)

Constructs an EJBTransactionRequiredException with the specified detailed message.
### Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class javax.ejb.EJBException</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCausedByException, getMessage, printStackTrace, printStackTrace</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, setStackTrace, toString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

**public EJBTransactionRequiredException()**

EJBTransactionRequiredException

**EJBTransactionRequiredException**

**public EJBTransactionRequiredException()**

Constructs an EJBTransactionRequiredException with no detail message.

**public EJBTransactionRequiredException(String message)**

EJBTransactionRequiredException
public EJBTransactionRequiredException(String message)

Constructs an EJBTransactionRequiredException with the specified detailed message.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.ejb  Class EJBTransactionRolledbackException

java.lang.Object
   └ java.lang.Throwable
       └ java.lang.Exception
           └ java.lang.RuntimeException
               └ javax.ejb.EJBException
                   └ javax.ejb.EJBTransactionRolledbackException

All Implemented Interfaces:
   Serializable

public class EJBTransactionRolledbackException
extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back. Thus the requested operation either could not be performed or was not performed because further computation on behalf of the transaction would be fruitless

See Also:
   Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJBTransactionRolledbackException()</td>
<td>Constructs an EJBTransactionRolledbackException with no detail message.</td>
</tr>
<tr>
<td>EJBTransactionRolledbackException(String message)</td>
<td>Constructs an EJBTransactionRolledbackException with the</td>
</tr>
</tbody>
</table>
EJBTransactionRolledbackException(String message, Exception ex)

Constructs an EJBTransactionRolledbackException with the specified detail message and a nested exception.

Method Summary

Methods inherited from class javax.ejb.EJBException
getCausedByException, getMessage, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.ThrowException
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public EJBTransactionRolledbackException()

EJBTransactionRolledbackException

public EJBTransactionRolledbackException()

    Constructs an EJBTransactionRolledbackException with no detail message.

public EJBTransactionRolledbackException(String
public EJBTransactionRolledbackException(String message)

Constructs an EJBTransactionRolledbackException with the specified detailed message.

public EJBTransactionRolledbackException(String message, Exception ex)

Constructs an EJBTransactionRolledbackException with the specified detail message and a nested exception.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.el Class ELContext

javax.lang.Object
   ↓ javax.el.ELContext

public abstract class ELContext
   extends Object

Expression   ELContextELContext

- base   ELResolver   base
- FunctionMapper  EL
- VariableMapper  EL
- ELResolver  Collection

Collection   ELResolver JSP  Faces

ELContext  JSP
ELContextListener  JspContext.getELContext()

ELContext  ELContext

since  JSP 2.1
See also  javax.el.ELContextListener, javax.el.ELContextEvent,
          javax.el.ELResolver, javax.el.FunctionMapper,
          javax.el.VariableMapper, javax.servlet.jsp.JspContext

Context information for expression evaluation.

To evaluate an Expression, an ELContext must be provided. The
ELContext holds:
• a reference to the base **ELResolver** that will be consulted to resolve model objects and their properties
• a reference to **FunctionMapper** that will be used to resolve EL Functions.
• a reference to **VariableMapper** that will be used to resolve EL Variables.
• a collection of all the relevant context objects for use by **ELResolver**
• state information during the evaluation of an expression, such as whether a property has been resolved yet

The collection of context objects is necessary because each **ELResolver** may need access to a different context object. For example, JSP and Faces resolvers need access to a **JspContext** and a **FacesContext**, respectively.

Creation of **ELContext** objects is controlled through the underlying technology. For example, in JSP the **JspContext.getELContext()** factory method is used. Some technologies provide the ability to add an **ELContextListener** so that applications and frameworks can ensure their own context objects are attached to any newly created **ELContext**.

Because it stores state during expression evaluation, an **ELContext** object is not thread-safe. Care should be taken to never share an **ELContext** instance between two or more threads.

**Since:**
JSP 2.1

**See Also:**
**ELContextListener**, **ELContextEvent**, **ELResolver**, **FunctionMapper**, **VariableMapper**, **JspContext**

---

### Constructor Summary

**ELContext()**

### Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Object</code></td>
<td></td>
</tr>
<tr>
<td><code>getContext(Class key)</code></td>
<td>Returns the context object associated with the given key.</td>
</tr>
<tr>
<td><code>abstract ELResolver</code></td>
<td></td>
</tr>
<tr>
<td><code>getELResolver()</code></td>
<td>Retrieves the ELResolver associated with this context.</td>
</tr>
<tr>
<td><code>abstract FunctionMapper</code></td>
<td></td>
</tr>
<tr>
<td><code>getFunctionMapper()</code></td>
<td>Retrieves the FunctionMapper associated with this context.</td>
</tr>
<tr>
<td><code>Locale</code></td>
<td></td>
</tr>
<tr>
<td><code>getLocale()</code></td>
<td>Get the Locale stored by a previous invocation to setLocale(java.util.Locale).</td>
</tr>
<tr>
<td><code>abstract VariableMapper</code></td>
<td></td>
</tr>
<tr>
<td><code>getVariableMapper()</code></td>
<td>Retrieves the VariableMapper associated with this ELContext.</td>
</tr>
<tr>
<td><code>boolean isPropertyResolved()</code></td>
<td>Returns whether an ELResolver has successfully resolved a given (base, property) pair.</td>
</tr>
<tr>
<td><code>void putContext(Class key, Object contextObject)</code></td>
<td>Associates a context object with this ELContext.</td>
</tr>
<tr>
<td><code>void setLocale(Locale locale)</code></td>
<td>Set the Locale for this instance.</td>
</tr>
<tr>
<td><code>void setPropertyResolved(boolean resolved)</code></td>
<td>Called to indicate that a ELResolver has successfully resolved a given (base, property) pair.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public ELContext()
ELContext

public ELContext()

### Method Detail

**public void setPropertyResolved(boolean resolved)**

ELResolver (base, property)

<table>
<thead>
<tr>
<th>Method Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>resolved</code></td>
<td>true</td>
</tr>
</tbody>
</table>

See also: [javax.el.CompositeELResolver](https://docs.oracle.com/javase/8/docs/api/javax/el/CompositeELResolver.html)

**setPropertyResolved**

public void `setPropertyResolved`(boolean resolved)

Called to indicate that a ELResolver has successfully resolved a given (base, property) pair.

The CompositeELResolver checks this property to determine whether it should consider or skip other component resolvers.

**Parameters:**
- `resolved`: true if the property has been resolved, or false if not.

**See Also:** [CompositeELResolver](https://docs.oracle.com/javase/8/docs/api/javax/el/CompositeELResolver.html)

**public boolean isPropertyResolved()**

ELResolver (base, property)

<table>
<thead>
<tr>
<th>Method Reference</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>return</code></td>
<td>true</td>
</tr>
</tbody>
</table>
isPropertyResolved

public boolean isPropertyResolved()

Returns whether an ELResolver has successfully resolved a given (base, property) pair.

The CompositeELResolver checks this property to determine whether it should consider or skip other component resolvers.

Returns:
true if the property has been resolved, or false if not.
See Also:
CompositeELResolver

putContext

public void putContext(Class<T> key, Object contextObject)

ELContext

ELContext Collection ELResolver Collection

contextObject key

key contextObject ELResolver Collection

Throws

NullPointerException: key null contextObject null

putContext

public void putContext(Class key,
Object contextObject)
Associates a context object with this ELContext.

The ELContext maintains a collection of context objects relevant to the evaluation of an expression. These context objects are used by ELResolvers. This method is used to add a context object to that collection.

By convention, the contextObject will be of the type specified by the key. However, this is not required and the key is used strictly as a unique identifier.

**Parameters:**
- **key** - The key used by an ELResolver to identify this context object.
- **contextObject** - The context object to add to the collection.

**Throws:**
- NullPointerException - if key is null or contextObject is null.

**public Object getContext(Class<T> key)**

**ELContext** Collection    ELResolver Collection

key

    key    ELContext
return null

Throws null

NullPointerException: key null

**getContext**

public Object getContext(Class key)

Returns the context object associated with the given key.
The ELContext maintains a collection of context objects relevant to the evaluation of an expression. These context objects are used by ELResolvers. This method is used to retrieve the context with the given key from the collection.

By convention, the object returned will be of the type specified by the key. However, this is not required and the key is used strictly as a unique identifier.

**Parameters:**
key - The unique identifier that was used to associate the context object with this ELContext.

**Returns:**
The context object associated with the given key, or null if no such context was found.

**Throws:**
NullPointerException - if key is null.

```java
abstract public ELResolver getELResolver()

ELResolver

getELResolver

public abstract ELResolver getELResolver()

Retrieves the ELResolver associated with this context.

The ELContext maintains a reference to the ELResolver that will be consulted to resolve variables and properties during an expression evaluation. This method retrieves the reference to the resolver.
Once an `ELContext` is constructed, the reference to the `ELResolver` associated with the context cannot be changed.

**Returns:**

The resolver to be consulted for variable and property resolution during expression evaluation.

---

**abstract public** `FunctionMapper` `getFunctionMapper()`

```java
ELContext FunctionMapper
return EL
```

**getFunctionMapper**

```java
public abstract FunctionMapper getFunctionMapper()
```

Retrieves the `FunctionMapper` associated with this `ELContext`.

**Returns:**

The function mapper to be consulted for the resolution of EL functions.

---

**public** `java.util.Locale` `getLocale()`

```java
#setLocale Locale null Locale Locale
Locale
return Locale
```

**getLocale**

```java
public Locale getLocale()
```

Get the `Locale` stored by a previous invocation to `setLocale(java.util.Locale)`. If this method returns non `null`, this `Locale` must be used for all localization needs in the implementation. The `Locale` must not be cached to allow for applications that change
Locale dynamically.

**Returns:**
The `Locale` in which this instance is operating. Used primarily for message localization.

```java
public void setLocale(java.util.Locale locale)
locale.setLocale(Locale.getDefault(), JSP EL)
Locale.getDefault();
```

**setLocale**

```java
public void setLocale(Locale locale)
    Locale.setLocale(Locale.getDefault());
```

Set the `Locale` for this instance. This method may be called by the party creating the instance, such as JavaServer Faces or JSP, to enable the EL implementation to provide localized messages to the user. If no `Locale` is set, the implementation must use the locale returned by `Locale.getDefault()`.

**abstract public** `VariableMapper` `getVariableMapper()`

```java
ELContext VariableMapper return
EL
```

**getVariableMapper**

```java
public abstract VariableMapper getVariableMapper()
```

Retrieves the `VariableMapper` associated with this `ELContext`.

**Returns:**
The variable mapper to be consulted for the resolution of EL variables.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.el  Class ELContextEvent

**java.lang.Object**  
└ java.util.EventObject
    └ javax.el.ELContextEvent

All Implemented Interfaces:  
Serializable

public class ELContextEvent  
extends EventObject

Extends: java.util.EventObject

**ELContext**  
ELContext

since JSP 2.1

See also  
javax.el.ELContext, javax.el.ELContextListener

An event which indicates that an ELContext has been created. The source object is the ELContext that was created.

**Since:**  
JSP 2.1

**See Also:**  
ELContext, ELContextListener, Serialized Form

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class java.util.EventObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

---

**Constructor Summary**
ELContextEvent(ELContext source)

Constructs an ELContextEvent object to indicate that an ELContext has been created.

Method Summary

<table>
<thead>
<tr>
<th>ELContext</th>
<th>getELContext()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the ELContext that was created.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.util.EventObject
ggetSource, toString

Methods inherited from class java.lang.Object
close, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

public ELContextEvent(ELContext source)

ELContextEvent

ELContext

source

ELContext

ELContextEvent

public ELContextEvent(ELContext source)

Constructs an ELContextEvent object to indicate that an ELContext has been created.

Parameters:

source - the ELContext that was created.
public **ELContext** getELContext()

    ELContext #getSource

    return ELContext

---

**getELContext**

public **ELContext** getELContext()

    Returns the **ELContext** that was created. This is a type-safe equivalent of the **EventObject.getSource()** method.

    **Returns:**
    
    the **ELContext** that was created.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to **license terms**.

PS:
javax.el Interface ELContextListener

All Superinterfaces:
   EventListener

public interface ELContextListener
extends EventListener

Implements: java.util.EventListener

ELContext
since JSP 2.1
See also javax.el.ELContext, javax.el.ELContextEvent

The listener interface for receiving notification when an ELContext is created.

Since: JSP 2.1
See Also: ELContext, ELContextEvent

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void contextCreated(ELContextEvent ece)</td>
<td>Invoked when a new ELContext has been created.</td>
</tr>
</tbody>
</table>

Method Detail

public void contextCreated(ELContextEvent ece)
ElContext
contextCreated

void contextCreated(ELContextEvent ece)

Invoked when a new ELContext has been created.

Parameters:

ece - the notification event.
This is an element marker interface. Under certain circumstances, it is necessary for the binding compiler to generate derived java content classes that implement this interface. In those cases, client applications must supply element instances rather than types of elements. For more detail, see section 5.7 "Element Declaration" and 5.7.1 "Bind to Java Element Interface" of the specification.

Since:
JAXB1.0

Version:
$Revision: 1.1$

Author:
- Ryan Shoemaker, Sun Microsystems, Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems, Inc.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.el Class ELEntryException

java.lang.object
   ▼ java.lang.Throwable
      ▼ java.lang.Exception
         ▼ java.lang.RuntimeException
            ▼ javax.el.ELEntryException

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   MethodNotFoundException, PropertyNotFoundException,
   PropertyNotWritableException

public class ELEntryException

extends RuntimeException

Extends: Throwable > Exception > RuntimeException
Extended by: MethodNotFoundException, PropertyNotFoundException,
PropertyNotWritableException

since JSP 2.1

Represents any of the exception conditions that can arise during expression evaluation.

Since:
   JSP 2.1

See Also:
   Serialized Form

---

Constructor Summary
<table>
<thead>
<tr>
<th>ELException()</th>
<th>Creates an ELException with no detail message.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELException(String pMessage)</td>
<td>Creates an ELException with the provided detail message.</td>
</tr>
<tr>
<td>ELException(String pMessage, Throwable pRootCause)</td>
<td>Creates an ELException with the given detail message and root cause.</td>
</tr>
<tr>
<td>ELException(Throwable pRootCause)</td>
<td>Creates an ELException with the given cause.</td>
</tr>
</tbody>
</table>

### Method Summary

Methods inherited from class java.lang.Throwable
- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.Object
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

public ELException()

ELException

public ELException()

 Creates an ELException with no detail message.
public ELEException(String pMessage)

    ELEException

    pMessage

ELEException

public ELEException(String pMessage)

    Creates an ELEException with the provided detail message.

    Parameters:

    pMessage - the detail message

public ELEException(Throwable pRootCause)

    ELEException

    pRootCause

ELEException

public ELEException(Throwable pRootCause)

    Creates an ELEException with the given cause.

    Parameters:

    pRootCause - the originating cause of this exception

public ELEException(String pMessage, Throwable pRootCause)

    ELEException

    pMessage
    pRootCause
public ELException(String pMessage, Throwable pRootCause)

Creates an ELException with the given detail message and root cause.

Parameters:
- pMessage - the detail message
- pRootCause - the originating cause of this exception
javax.servlet.jsp.el Class ELException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.servlet.jsp.el.ELEException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  ELParseException

---

Deprecated. As of JSP 2.1, replaced by ELEException

Extends: Throwable > Exception
Extended by: ELParseException

Since: 2.0

public class ELException
extends Exception

Represents any of the exception conditions that arise during the operation evaluation of the evaluator.

Since:
  JSP 2.0

See Also:
  Serialized Form

---

Constructor Summary

ELEException()
**Deprecated.** Creates an ELException with no detail message.

ELException(String pMessage)

**Deprecated.** Creates an ELException with the provided detail message.

ELException(String pMessage, Throwable pRootCause)

**Deprecated.** Creates an ELException with the given detail message and root cause.

ELException(Throwable pRootCause)

**Deprecated.** Creates an ELException with the given root cause.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable getRootCause()</td>
<td><strong>Deprecated.</strong> Returns the root cause.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

public ELException()

ELException

**Deprecated.**

public ELException()
Creates an ELException with no detail message.

```java
public ELException(String pMessage)
```

**Parameters:**
- pMessage - the detail message

---

**Deprecated.**
Creates an ELException with the provided detail message.

**Parameters:**
- pMessage - the detail message

```java
public ELException(Throwable pRootCause)
```

**Parameters:**
- pRootCause - the originating cause of this exception

---

**Deprecated.**
Creates an ELException with the given root cause.

```java
public ELException(String pMessage, Throwable pRootCause)
```
public ELException(String pMessage, Throwable pRootCause)

Deprecated.
Creates an ELException with the given detail message and root cause.

Parameters:
  pMessage - the detail message
  pRootCause - the originating cause of this exception

public Throwable getRootCause()

    return

getRootCause

public Throwable getRootCause()

    Deprecated.
    Returns the root cause.

    Returns:
    the root cause of this exception
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet.jsp.el  Class ELParseException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.servlet.jsp.el.ELException
              └ javax.servlet.jsp.el.ELParseException

All Implemented Interfaces:
  Serializable

---

Deprecated. As of JSP 2.1, replaced by ELException

Extends: Throwable > Exception > ELException

EL

since 2.0

public class ELParseException
extends ELException

Represents a parsing error encountered while parsing an EL expression.

Since:
  JSP 2.0
See Also:
  Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
</table>
|             | ELParseException()  
  |   Deprecated. Creates an ELParseException with no detail message. |
|             | ELParseException(String pMessage)  
  |   Deprecated. Creates an ELParseException with the provided message. |
Method Summary

Methods inherited from class javax.servlet.jsp.el.ELException
getRootCause

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public ELParseException()
ELParseException

ELParseException

public ELParseException()

Deprecated.
Creates an ELParseException with no detail message.

public ELParseException(String pMessage)
ELParseException

pMessage
public ELParseException(String pMessage)

    Deprecated.
    Creates an ELParseException with the provided detail message.

    Parameters:
    pMessage - the detail message
javax.el Class ELResolver

java.lang.Object
   \_ javax.el.ELResolver

Direct Known Subclasses:
   ArrayELResolver, BeanELResolver, CompositeELResolver,
   ImplicitObjectELResolver, ListELResolver, MapELResolver,
   ResourceBundleELResolver, ScopedAttributeELResolver

public abstract class ELResolver

extends Object

Extended by: ArrayELResolver, BeanELResolver,
   CompositeELResolver, ListELResolver, MapELResolver,
   ResourceBundleELResolver

EL

   ELContext  ELResolver  .  []

EL  ${employee.lastName}  ELResolver  employee
lastName

   base  property  ${employee.lastName}  employee
   base  nullproperty  String  property  String
   ELResolver

   base  base  property  base
$employee.lastName  base  employee  property  "lastName"
${y[x]}  base  y  property  x

   ELResolver  ELContext  ELResolver
   CompositeELResolver

   \#getValue\#getType\#setValue  \#isReadOnly  ELResolver  (base,
   property)  base  (base, property)
Enables customization of variable and property resolution behavior for EL expression evaluation.

While evaluating an expression, the ELResolver associated with the ELContext is consulted to do the initial resolution of the first variable of an expression. It is also consulted when a . or [ ] operator is encountered, except for the last such operator in a method expression, in which case the resolution rules are hard coded.

For example, in the EL expression ${employee.lastName}, the ELResolver determines what object employee refers to, and what it means to get the lastName property on that object.

Most methods in this class accept a base and property parameter. In the case of variable resolution (e.g. determining what employee refers to in ${employee.lastName}), the base parameter will be null and the property parameter will always be of type String. In this case, if the property is not a String, the behavior of the ELResolver is undefined.

In the case of property resolution, the base parameter identifies the base object and the property object identifies the property on that base. For example, in the expression ${employee.lastName}, base is the result of the variable resolution for employee and property is the string "lastName". In the expression ${y[x]}, base is the result of the variable resolution for y and property is the result of the variable resolution for x.

Though only a single ELResolver is associated with an ELContext, there are usually multiple resolvers considered for any given variable or property resolution. ELResolverS are combined together using CompositeELResolverS, to define rich semantics for evaluating an
expression.

For the `getValue(javax.el.ELContext, java.lang.Object, java.lang.Object)`, `getType(javax.el.ELContext, java.lang.Object, java.lang.Object)`, `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` and `isReadOnly(javax.el.ELContext, java.lang.Object, java.lang.Object)` methods, an `ELResolver` is not responsible for resolving all possible (base, property) pairs. In fact, most resolvers will only handle a base of a single type. To indicate that a resolver has successfully resolved a particular (base, property) pair, it must set the `propertyResolved` property of the `ELContext` to true. If it could not handle the given pair, it must leave this property alone. The caller must ignore the return value of the method if `propertyResolved` is false.

The `getFeatureDescriptors(javax.el.ELContext, java.lang.Object)` and `getCommonPropertyType(javax.el.ELContext, java.lang.Object)` methods are primarily designed for design-time tool support, but must handle invocation at runtime as well. The `Beans.isDesignTime()` method can be used to determine if the resolver is being consulted at design-time or runtime.

Since:  
JSP 2.1

See Also:  
`CompositeELResolver`, `ELContext.getELResolver()`

### Field Summary

<table>
<thead>
<tr>
<th></th>
<th><code>RESOLVABLE_AT_DESIGN_TIME</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static String</code></td>
<td>The attribute name of the named attribute in the <code>FeatureDescriptor</code> that specifies whether the variable or property can be resolved at runtime.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th><code>TYPE</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static String</code></td>
<td>The attribute name of the named attribute in the <code>FeatureDescriptor</code> that specifies the runtime type of the variable or property.</td>
</tr>
</tbody>
</table>
Constructor Summary

ELResolver()

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getCommonPropertyType</strong> (ELContext context, Object base)</td>
<td>Returns the most general type that this resolver accepts for the property argument, given a base object.</td>
</tr>
<tr>
<td><strong>getFeatureDescriptors</strong> (ELContext context, Object base)</td>
<td>Returns information about the set of variables or properties that can be resolved for the given base object.</td>
</tr>
<tr>
<td><strong>getType</strong> (ELContext context, Object base, Object property)</td>
<td>For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) method.</td>
</tr>
<tr>
<td><strong>getValue</strong> (ELContext context, Object base, Object property)</td>
<td>Attempts to resolve the given property object on the given base object.</td>
</tr>
<tr>
<td><strong>isReadOnly</strong> (ELContext context, Object base, Object property)</td>
<td>For a given base and property, attempts to determine whether a call to setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always fail.</td>
</tr>
<tr>
<td><strong>setValue</strong> (ELContext context, Object base, Object property, Object value)</td>
<td></td>
</tr>
</tbody>
</table>
abstract void Attempts to set the value of the given property object on the given base object.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

TYPE
public static final String TYPE

The attribute name of the named attribute in the FeatureDescriptor that specifies the runtime type of the variable or property.

See Also:
Constant Field Values

RESOLVABLE_AT_DESIGN_TIME
public static final String RESOLVABLE_AT_DESIGN_TIME

The attribute name of the named attribute in the FeatureDescriptor that specifies whether the variable or property can be resolved at runtime.

See Also:
Constant Field Values

Constructor Detail
public ELResolver()

ELResolver

public ELResolver()

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getValue</code></td>
<td>abstract public Object getValue(ELContext context, Object base, Object property)</td>
</tr>
<tr>
<td></td>
<td>base	property</td>
</tr>
<tr>
<td></td>
<td>(base, property)            ELContext	propertyResolved false	false</td>
</tr>
<tr>
<td></td>
<td>true	true                 ELContext	propertyResolved true   true</td>
</tr>
<tr>
<td></td>
<td>context                    base	null</td>
</tr>
<tr>
<td></td>
<td>base                      base	null</td>
</tr>
<tr>
<td></td>
<td>property                   ELContext	propertyResolved true   true</td>
</tr>
<tr>
<td></td>
<td>return                    ELContext	propertyResolved true   true</td>
</tr>
<tr>
<td></td>
<td>Throws                    NullPointerException: context	null</td>
</tr>
<tr>
<td></td>
<td>PropertyNotFoundException: ELResolver (base, property)</td>
</tr>
<tr>
<td></td>
<td>ELException: cause</td>
</tr>
</tbody>
</table>

getValue

public abstract Object getValue(ELContext context, Object base, Object property)

Attempts to resolve the given property object on the given base object.

If this resolver handles the given (base, property) pair, the propertyResolved property of the ELContext object must be set to
true by the resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

**Parameters:**
context - The context of this evaluation.
base - The base object whose property value is to be returned, or null to resolve a top-level variable.
property - The property or variable to be resolved.

**Returns:**
If the propertyResolved property of ELContext was set to true, then the result of the variable or property resolution; otherwise undefined.

**Throws:**
NullPointerException - if context is null
PropertyNotFoundException - if the given (base, property) pair is handled by this ELResolver but the specified variable or property does not exist or is not readable.
ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

abstract public Class<T> getType(ELContext context, Object base, Object property)

(base, property) ELContext propertyResolved
true true

g getValue().getClass() ArrayELResolver getType

context base property
null

return ELContext propertyResolved true

Throws NullPointerException: context null
Throws `PropertyNotFoundException`: ELResolver (base, property)

Throws `ELException`: cause

**getType**

```java
public abstract Class<? super T> getType(ELContext context,
                                     Object base,
                                     Object property)
```

For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the `setValue(javax.el.ELContext,

If this resolver handles the given (base, property) pair, the `propertyResolved` property of the ELContext object must be set to `true` by the resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

This is not always the same as `getValue().getClass()`. For example, in the case of an `ArrayELResolver`, the `getType` method will return the element type of the array, which might be a superclass of the type of the actual element that is currently in the specified array element.

**Parameters:**
- context - The context of this evaluation.
- base - The base object whose property value is to be analyzed, or `null` to analyze a top-level variable.
- property - The property or variable to return the acceptable type for.

**Returns:**
- If the `propertyResolved` property of ELContext was set to `true`, then the most general acceptable type; otherwise undefined.

**Throws:**
- `NullPointerException` - if `context` is `null`.
- `PropertyNotFoundException` - if the given (base, property) pair is handled by this ELResolver but the specified variable or property references are not found.
abstract public void setValue(ELContext context, Object base, Object property, Object value)

(base, property) ELContext propertyResolved
true true

context base
base null
property
value

Throws NullPointerException: context null
Throws PropertyNotFoundException: ELResolver (base, property)
Throws PropertyNotWritableException: ELResolver (base, property)
Throws ELException: cause

setValue

public abstract void setValue(ELContext context,
Object base,
Object property,
Object value)

Attempts to set the value of the given property object on the given base object.

If this resolver handles the given (base, property) pair, the propertyResolved property of the ELContext object must be set to true by the resolver, before returning. If this property is not true after
this method is called, the caller can safely assume no value has been set.

**Parameters:**
- context - The context of this evaluation.
- base - The base object whose property value is to be set, or null to set a top-level variable.
- property - The property or variable to be set.
- value - The value to set the property or variable to.

**Throws:**
- NullPointerException - if context is null
- PropertyNotFoundException - if the given (base, property) pair is handled by this ELResolver but the specified variable or property does not exist.
- PropertyNotWritableException - if the given (base, property) pair is handled by this ELResolver but the specified variable or property is not writable.
- ELException - if an exception was thrown while attempting to set the property or variable. The thrown exception must be included as the cause property of this exception, if available.

```java
abstract public boolean isReadOnly(ELContext context, Object base, Object property)
    base  property  #setValue
(base, property)  ELContext  propertyResolved
true   true

context
base  base  null
property
return  ELContext  propertyResolved  true  true  false
Throws  NullPointerException:  context  null
Throws  PropertyNotFoundException:  ELResolver (base, property)
Throws  ELException:  cause
```
**isReadOnly**

```java
public abstract boolean isReadOnly(ELContext context, Object base, Object property)
```

For a given `base` and `property`, attempts to determine whether a call to `setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object)` will always fail.

If this resolver handles the given `(base, property)` pair, the `propertyResolved` property of the `ELContext` object must be set to `true` by the resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

**Parameters:**
- **context** - The context of this evaluation.
- **base** - The base object whose property value is to be analyzed, or `null` to analyze a top-level variable.
- **property** - The property or variable to return the read-only status for.

**Returns:**
- If the `propertyResolved` property of `ELContext` was set to `true`, then `true` if the property is read-only or `false` if not; otherwise undefined.

**Throws:**
- `NullPointerException` - if `context` is `null`
- `PropertyNotFoundException` - if the given `(base, property)` pair is handled by this `ELResolver` but the specified variable or property does not exist.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

```java
abstract public java.util.Iterator<E>
getFeatureDescriptors(ELContext context, Object base)
```
getFeatureDescriptors

```java
public abstract Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext base) {
    // Returns information about the set of variables or properties that can be resolved for the given base object. One use for this method is to assist tools in auto-completion.
```

If the base parameter is null, the resolver must enumerate the list of top-level variables it can resolve.

The Iterator returned must contain zero or more instances of FeatureDescriptor, in no guaranteed order. In the case of primitive types such as int, the value null must be returned. This is to prevent the useless iteration through all possible primitive values. A return value of null indicates that this resolver does not handle the given base object or that the results are too complex to represent with this method and the getCommonPropertyType(javax.el.ELContext, java.lang.Object) method should be used instead.

Each FeatureDescriptor will contain information about a single variable or property. In addition to the standard properties, the FeatureDescriptor must have two named attributes (as set by the setValue method):

- **TYPE** - The value of this named attribute must be an instance of java.lang.Class and specify the runtime type of the variable or property.
- **RESOLVABLE_AT_DESIGN_TIME** - The value of this named attribute must be an instance of java.lang.Boolean and indicates whether it is safe to attempt to resolve this property at design-time. For instance, it may be unsafe to attempt a resolution at design time if the ELResolver needs access to a resource that is only available at runtime and no acceptable simulated value can be provided.

The caller should be aware that the Iterator returned might iterate through a very large or even infinitely large set of properties. Care should be taken by the caller to not get stuck in an infinite loop.

This is a "best-effort" list. Not all ELResolvers will return completely accurate results, but all must be callable at both design-time and runtime (i.e. whether or not Beans.isDesignTime() returns true), without causing errors.

The propertyResolved property of the ELContext is not relevant to this method. The results of all ELResolvers are concatenated in the case
of composite resolvers.

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The base object whose set of valid properties is to be enumerated, or `null` to enumerate the set of top-level variables that this resolver can evaluate.

**Returns:**
- An `Iterator` containing zero or more (possibly infinitely more) `FeatureDescriptor` objects, or `null` if this resolver does not handle the given `base` object or that the results are too complex to represent with this method.

**See Also:**
- `FeatureDescriptor`

---

**abstract public Class<T>**

**getCommonPropertyType(ELContext context, Object base)**

```java
Base property
```

`ArrayELResolver`  `int`  `property`

`Integer.class`

```java
context
base
null
```

**getCommonPropertyType**

```java
public abstract Class<? super T> getCommonPropertyType(ELContext context, Object base)
```

Returns the most general type that this resolver accepts for the `property` argument, given a `base` object. One use for this method is to assist tools in auto-completion.
This assists tools in auto-completion and also provides a way to express that the resolver accepts a primitive value, such as an integer index into an array. For example, the `ArrayELResolver` will accept any `int` as a property, so the return value would be `Integer.class`.

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The base object to return the most general property type for, or `null` to enumerate the set of top-level variables that this resolver can evaluate.

**Returns:**
- `null` if this `ELResolver` does not know how to handle the given base object; otherwise `Object.class` if any type of property is accepted; otherwise the most general property type accepted for the given base.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES  NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
Interface EmailAddress

public interface EmailAddress

See also javax.xml.registry.infomodel.User

Represents an email address.

**Author:**
Farrukh S. Najmi

**See Also:**
User

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String</strong></td>
<td><strong>getAddress()</strong>&lt;br&gt;Returns the email address for this object.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><strong>getType()</strong>&lt;br&gt;Gets the type for this object.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setAddress(String address)</strong>&lt;br&gt;Sets the email address for this object.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setType(String type)</strong>&lt;br&gt;Sets the type for this object.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public String getAddress() throws JAXRException
null
```
getAddress

`String getAddress()`

Throws: `JAXRException`

Returns the email address for this object. Default is a NULL String.

Capability Level: 0

Returns:
- the actual email address (e.g. john.doe@acme.com)

Throws:
- `JAXRException` - If the JAXR provider encounters an internal error

public void setAddress(String address) throws `JAXRException`

setAddress

`void setAddress(String address)`

Throws: `JAXRException`

Sets the email address for this object.
Capability Level: 0

Parameters:
address - the actual email address (e.g. john.doe@acme.com)

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public String getType() throws JAXRException
NULL
0

    return "Home" "Office"

Throws  JAXRException: JAXR

get Type

String getType() throws JAXRException

Gets the type for this object. Default is a NULL String.

Capability Level: 0

Returns:
the usage type for this object which is an arbitrary value (e.g.
"Home" or "Office")

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void setType(String type) throws JAXRException
setType

void setType(String type)
throws JAXRException

Sets the type for this object.

Capability Level: 0

Parameters:
  type - the usage type for this object which is an arbitrary value (e.g. "Home" or "Office")

Throws:
  JAXRException - If the JAXR provider encounters an internal error
javax.persistence Annotation Type Embeddable

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
public @interface Embeddable

**Implements**: Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

**Transient**  
since Java Persistence 1.0

Defines a class whose instances are stored as an intrinsic part of an owning entity and share the identity of the entity. Each of the persistent properties or fields of the embedded object is mapped to the database table for the entity. Only Basic, Column, Lob, Temporal, and Enumerated mapping annotations may portably be used to map the persistent fields or properties of classes annotated as Embeddable.

Note that the Transient annotation may be used to designate the non-persistent state of an embeddable class.

**Since**:  
Java Persistence 1.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
javax.persistence  Annotation Type Embedded

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Embedded

**Implements:** Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@Embedded
@AttributeOverrides(
    @AttributeOverride(name="startDate",
                        column=@Column("EMP_START")),
    @AttributeOverride(name="endDate",
                        column=@Column("EMP_END")))

public EmploymentPeriod getEmploymentPeriod() { ... }

**since**  Java Persistence 1.0

Defines a persistent field or property of an entity whose value is an instance of an embeddable class. The embeddable class must be annotated as **Embeddable**.

Example:

@Embedded
@AttributeOverrides(
    @AttributeOverride(name="startDate",
                        column=@Column("EMP_START")),
    @AttributeOverride(name="endDate",
                        column=@Column("EMP_END")))

public EmploymentPeriod getEmploymentPeriod() { ... }

Since:
javax.persistence Annotation Type EmbeddedId

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface EmbeddedId

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@EmbeddedId
protected EmployeePK empPK;

since Java Persistence 1.0

Is applied to a persistent field or property of an entity class or mapped superclass to denote a composite primary key that is an embeddable class. The embeddable class must be annotated as Embeddable.

Example:

@EmbeddedId
protected EmployeePK empPK;

Since:
Java Persistence 1.0

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream.events

Interface EndDocument

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface EndDocument
extends XMLEvent

Implements: XMLEvent

version 1.0

A marker interface for the end of the document

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.xml.stream.XMLStreamConstants</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from interface</th>
</tr>
</thead>
</table>
### javax.xml.stream.events

| asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode |

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES  NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream.events Interface EndElement

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface EndElement
extends XMLEvent
Implements: XMLEvent

EndElement
    version 1.0
See also javax.xml.stream.events.XMLEvent

An interface for the end element event. An EndElement is reported for each End Tag in the document.

Version:
    1.0
Author:
    Copyright (c) 2003 by BEA Systems. All Rights Reserved.
See Also:
   XMLEvent

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.xml.stream.XMLStreamConstants</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT,</td>
</tr>
<tr>
<td>END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE,</td>
</tr>
<tr>
<td>NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE,</td>
</tr>
<tr>
<td>START_DECLARATION, START_DOCUMENT, START_ELEMENT</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QName getName()</td>
<td>Get the name of this event</td>
</tr>
<tr>
<td>Iterator getNamespaces()</td>
<td>Returns an Iterator of namespaces that have gone out of scope.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.stream.events.XMLEvent

asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public javax.xml.namespace.QName getName()

    return

gName

QName getName()  

Get the name of this event

Returns:  
the qualified name of this event

public java.util.Iterator<E> getNamespaces()

Iterator  

return  
Namespace  Iterator
getNamespaces

`Iterator getNamespaces()`

Returns an Iterator of namespaces that have gone out of scope. Returns an empty iterator if no namespaces have gone out of scope.

Returns:
- an Iterator over Namespace interfaces, or an empty iterator

```
Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD
```

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws Class Endpoint

java.lang.Object
   javax.xml.ws.Endpoint

public abstract class Endpoint
   extends Object

Web

   Binding
   publish stop

Executor ThreadPoolExecutor

Binding

WSDL XMLSchema JAX-WS
since JAX-WS 2.0
See javax.xml.ws.Binding, javax.xml.ws.BindingType,
javax.xml.ws.soap.SOAPBinding,
java.util.concurrent.Executor

A Web service endpoint.

Endpoints are created using the static methods defined in this class. An endpoint is always tied to one Binding and one implementor, both set at endpoint creation time.

An endpoint is either in a published or an unpublished state. The publish methods can be used to start publishing an endpoint, at which point it starts accepting incoming requests. Conversely, the stop method can be used to stop accepting incoming requests and take the endpoint down.
Once stopped, an endpoint cannot be published again.

An `Executor` may be set on the endpoint in order to gain better control over the threads used to dispatch incoming requests. For instance, thread pooling with certain parameters can be enabled by creating a `ThreadPoolExecutor` and registering it with the endpoint.

Handler chains can be set using the contained `Binding`.

An endpoint may have a list of metadata documents, such as WSDL and XMLSchema documents, bound to it. At publishing time, the JAX-WS implementation will try to reuse as much of that metadata as possible instead of generating new one based on the annotations present on the implementor.

**Since:**  
JAX-WS 2.0

**See Also:**  
`Binding`, `BindingType`, `SOAPBinding`, `Executor`

<table>
<thead>
<tr>
<th>Field Summary</th>
<th></th>
</tr>
</thead>
</table>
| static `String` | **WSDL_PORT**  
Standard property: name of WSDL port.  |
| static `String` | **WSDL_SERVICE**  
Standard property: name of WSDL service.  |

<table>
<thead>
<tr>
<th>Constructor Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Endpoint()</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
<th></th>
</tr>
</thead>
</table>
| static `Endpoint` | **create**(Object implementor)  
Creates an endpoint with the specified implementor object.  |
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>create(String bindingId, Object implementor)</code></td>
<td>Creates an endpoint with the specified binding type and implementor object.</td>
</tr>
<tr>
<td><code>getBinding()</code></td>
<td>Returns the binding for this endpoint.</td>
</tr>
<tr>
<td><code>getExecutor()</code></td>
<td>Returns the executor for this <code>Endpoint</code> instance.</td>
</tr>
<tr>
<td><code>getImplementor()</code></td>
<td>Returns the implementation object for this endpoint.</td>
</tr>
<tr>
<td><code>getMetadata()</code></td>
<td>Returns a list of metadata documents for the service.</td>
</tr>
<tr>
<td><code>getProperties()</code></td>
<td>Returns the property bag for this <code>Endpoint</code> instance.</td>
</tr>
<tr>
<td><code>isPublished()</code></td>
<td>Returns true if the endpoint is in the published state.</td>
</tr>
<tr>
<td><code>publish(Object serverContext)</code></td>
<td>Publishes this endpoint at the provided server context.</td>
</tr>
<tr>
<td><code>publish(String address)</code></td>
<td>Publishes this endpoint at the given address.</td>
</tr>
<tr>
<td><code>publish(String address, Object implementor)</code></td>
<td>Creates and publishes an endpoint for the specified implementor object at the given address.</td>
</tr>
<tr>
<td><code>setExecutor(Executor executor)</code></td>
<td>Sets the executor for this <code>Endpoint</code> instance.</td>
</tr>
<tr>
<td><code>setMetadata(List&lt;Source&gt; metadata)</code></td>
<td>Sets the metadata for this endpoint.</td>
</tr>
<tr>
<td><code>setProperties(Map&lt;String, Object&gt; properties)</code></td>
<td>Sets the property bag for this <code>Endpoint</code> instance.</td>
</tr>
<tr>
<td><code>stop()</code></td>
<td>Stops publishing this endpoint.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

WSDL_SERVICE

public static final String WSDL_SERVICE

Standard property: name of WSDL service.

Type: javax.xml.namespace.QName

See Also:
Constant Field Values

WSDL_PORT

public static final String WSDL_PORT

Standard property: name of WSDL port.

Type: javax.xml.namespace.QName

See Also:
Constant Field Values

Constructor Detail


public Endpoint()

Endpoint

public Endpoint()

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
</table>

public static Endpoint create(Object implementor)

BindingType  SOAP 1.1 / HTTP

javax.xml.ws.Endpoint#publish(String)
javax.xml.ws.Endpoint#publish(Object)

Implementor

return

create

public static Endpoint create(Object implementor)

Creates an endpoint with the specified implementor object. If there is a binding specified via a BindingType annotation then it MUST be used else a default of SOAP 1.1 / HTTP binding MUST be used.

The newly created endpoint may be published by calling one of the javax.xml.ws.Endpoint#publish(String) and javax.xml.ws.Endpoint#publish(Object) methods.

Parameters:
implementor - The endpoint implementor.

Returns:
The newly created endpoint.
public static Endpoint create(String bindingId, Object implementor)

javax.xml.ws.Endpoint#publish(String)
javax.xml.ws.Endpoint#publish(Object)

bindingId URI bindingID null BindingType
implementor SOAP 1.1 / HTTP

create

public static Endpoint create(String bindingId, Object implementor)

Creates an endpoint with the specified binding type and implementor object.

The newly created endpoint may be published by calling one of the
javax.xml.ws.Endpoint#publish(String) and
javax.xml.ws.Endpoint#publish(Object) methods.

Parameters:
  bindingId - A URI specifying the binding to use. If the bindingID
  is null and no binding is specified via a BindingType annotation
  then a default SOAP 1.1 / HTTP binding MUST be used.
  implementor - The endpoint implementor.

Returns:
The newly created endpoint.

abstract public Binding getBinding()

return
getBinding

public abstract Binding getBinding()

Returns the binding for this endpoint.

Returns:
The binding for this endpoint

abstract public Object getImplementor()

return

getImplementor

public abstract Object getImplementor()

Returns the implementation object for this endpoint.

Returns:
The implementor for this endpoint

abstract public void publish(String address)

JAX-WS
javax.xml.ws.Endpoint#publish(Object)

address URI

Throws IllegalArgumentException: URI

Throws IllegalStateException: IllegalStateException:

publish
public abstract void publish(String address)

Publishes this endpoint at the given address. The necessary server infrastructure will be created and configured by the JAX-WS implementation using some default configuration. In order to get more control over the server configuration, please use the javax.xml.ws.Endpoint#publish(Object) method instead.

**Parameters:**
- address - A URI specifying the address to use. The address must be compatible with the binding specified at the time the endpoint was created.

**Throws:**
- IllegalArgumentException - If the provided address URI is not usable in conjunction with the endpoint's binding.
- IllegalStateException - If the endpoint has been published already or it has been stopped.

public static Endpoint publish(String address, Object implementor)

**JAX-WS**
javax.xml.ws.Endpoint#create(String,Object)
javax.xml.ws.Endpoint#publish(Object)

address / URI http URI SOAP 1.1/HTTP URI

implementor

return

publish

public static Endpoint publish(String address, Object implementor)
Creates and publishes an endpoint for the specified implementor object at the given address.

The necessary server infrastructure will be created and configured by the JAX-WS implementation using some default configuration. In order to get more control over the server configuration, please use the javax.xml.ws.Endpoint#create(String,Object) and javax.xml.ws.Endpoint#publish(Object) method instead.

**Parameters:**
- **address** - A URI specifying the address and transport/protocol to use. A http: URI must result in the SOAP 1.1/HTTP binding being used. Implementations may support other URI schemes.
- **implementor** - The endpoint implementor.

**Returns:**
The newly created endpoint.

---

```java
abstract public void publish(Object serverContext)
```

**Throws**
- `IllegalArgumentException`:  
- `IllegalStateException`:  

---

**publish**

```java
public abstract void publish(Object serverContext)
```

Publishes this endpoint at the provided server context. A server context encapsulates the server infrastructure and addressing information for a particular transport. For a call to this method to succeed, the server context passed as an argument to it must be compatible with the endpoint's binding.

**Parameters:**
- **serverContext** - An object representing a server context to be used for publishing the endpoint.
Throws:

**IllegalArgumentException** - If the provided server context is not supported by the implementation or turns out to be unusable in conjunction with the endpoint's binding.

**IllegalStateException** - If the endpoint has been published already or it has been stopped.

abstract public void stop()

stop

public abstract void stop()

Stops publishing this endpoint. If the endpoint is not in a published state, this method has not effect.

abstract public boolean isPublished()

ture

return true

isPublished

public abstract boolean isPublished()

Returns true if the endpoint is in the published state.

Returns:

ture if the endpoint is in the published state.

abstract public java.util.List<E> getMetadata()

return List<javax.xml.transform.Source>
getMetadata

public abstract List<Source> getMetadata()

Returns a list of metadata documents for the service.

Returns:
List<javax.xml.transform.Source> A list of metadata documents for the service

setMetadata

public abstract void setMetadata(List<Source> metadata)

Sets the metadata for this endpoint.

Parameters:
metadata - A list of XML document sources containing metadata information for the endpoint (e.g. WSDL or XML Schema documents)

Throws:
IllegalStateException - If the endpoint has already been published.

abstract public java.util.concurrent.Executor getExecutor()

Endpoint
return java.util.concurrent.Executor
See also java.util.concurrent.Executor

getExecutor
public abstract Executor getExecutor()

Returns the executor for this Endpoint instance. The executor is used to dispatch an incoming request to the implementor object.

Returns:
The java.util.concurrent.Executor to be used to dispatch a request.

See Also:
Executor

abstract public void setExecutor(java.util.concurrent.Executor executor)

Sets the executor for this Endpoint instance. The executor is used to dispatch an incoming request to the implementor object. If this Endpoint is published using the publish(Object) method and the specified server context defines its own threading behavior, the executor may be ignored.

Parameters:

executor - The java.util.concurrent.Executor to be used to dispatch a request.

Throws:

SecurityException - If the instance does not support setting an executor for security reasons (e.g. the necessary permissions are missing).
abstract public java.util.Map<K, V> getProperties()

    Endpoint
    return Map<String, Object>

getProperties

public abstract Map<String, Object> getProperties()

    Returns the property bag for this Endpoint instance.

    Returns:
    Map<String, Object> The property bag associated with this instance.

setProperties

public abstract void setProperties(Map<String, Object> properties)

    Sets the property bag for this Endpoint instance.

    Parameters:
    properties - The property bag associated with this instance.
PS:
javax.ejb

Interface EnterpriseBean

All Superinterfaces:
Serializable

All Known Subinterfaces:
EntityBean, MessageDrivenBean, SessionBean

public interface EnterpriseBean
extends Serializable

Implements: java.io.Serializable
Implemented by: EntityBean, MessageDrivenBean, SessionBean

Bean EnterpriseBean SessionBean EntityBean MessageDrivenBean

The EnterpriseBean interface must be implemented by every enterprise Bean class. It is a common superinterface for the SessionBean, EntityBean and MessageDrivenBean interfaces.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence  **Annotation Type Entity**

```java
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Entity
```

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

---

**since**  
Java Persistence 1.0

Specifies that the class is an entity. This annotation is applied to the entity class.

**Since:**
Java Persistence 1.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Optional Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>name</strong></td>
<td>The name of an entity.</td>
</tr>
</tbody>
</table>

**abstract public String name()**

Java Persistence

**name**

```java
public abstract String name
```

The name of an entity. Defaults to the unqualified name of the entity.
class. This name is used to refer to the entity in queries. The name must not be a reserved literal in the Java Persistence query language.

Default:

"""
javax.ejb Interface EntityBean

All Superinterfaces:
   EnterpriseBean, Serializable

public interface EntityBean
extends EnterpriseBean

Implements: EnterpriseBean

EntityBean Bean EntityBean Bean

The EntityBean interface is implemented by every entity enterprise Bean class. The container uses the EntityBean methods to notify the enterprise Bean instances of the instance's life cycle events.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void ejbActivate()</td>
<td>A container invokes this method when the instance is taken out of the pool of available instances to become associated with a specific EJB object.</td>
</tr>
<tr>
<td>void ejbLoad()</td>
<td>A container invokes this method to instruct the instance to synchronize its state by loading its state from the underlying database.</td>
</tr>
<tr>
<td>void ejbPassivate()</td>
<td>A container invokes this method on an instance before the instance becomes disassociated with a specific EJB object.</td>
</tr>
<tr>
<td>void ejbRemove()</td>
<td>A container invokes this method before it removes the EJB object that is currently associated with the instance.</td>
</tr>
<tr>
<td>ejbStore()</td>
<td></td>
</tr>
</tbody>
</table>
A container invokes this method to instruct the instance to synchronize its state by storing it to the underlying database.

void setEntityContext(EntityContext ctx)
    Set the associated entity context.

void unsetEntityContext()
    Unset the associated entity context.

Method Detail

public void setEntityContext(EntityContext ctx) throws EJBException, java.rmi.RemoteException

    ctx   EntityContext
    Throws   EJBException:
        java.rmi.RemoteException: EJB 1.0 Bean EJB
    Throws 1.1 Bean javax.ejb.EJBException EJB2.0
        Bean javax.ejb.EJBException

setEntityContext

void setEntityContext(EntityContext ctx)
    throws EJBException,
            RemoteException

    Set the associated entity context. The container invokes this method on an instance after the instance has been created.

    This method is called in an unspecified transaction context.

Parameters:
    ctx - An EntityContext interface for the instance. The instance should store the reference to the context in an instance variable.
Throws:

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the `javax.ejb.EJBException` instead of this exception. Enterprise beans written for the EJB 2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.

```java
public void unsetEntityContext() throws EJBException, java.rmi.RemoteException
```

Java `finalize()`

Throws: **EJBException**

- `java.rmi.RemoteException`: EJB 1.0 Bean EJB

Throws: 1.1 Bean `javax.ejb.EJBException` EJB 2.0 Bean `javax.ejb.EJBException`

```java
unsetEntityContext
```

Unset the associated entity context. The container calls this method before removing the instance.

This is the last method that the container invokes on the instance. The Java garbage collector will eventually invoke the `finalize()` method on the instance.
This method is called in an unspecified transaction context.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the `javax.ejb.EJBException` instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.

```java
public void ejbRemove() throws RemoveException, EJBException, java.rmi.RemoteException {
    // Implementation...
}
```

A container invokes this method before it removes the EJB object that is currently associated with the instance. This method is invoked when a client invokes a remove operation on the enterprise Bean's home interface or the EJB object's remote interface. This method transitions the instance from the ready state to the pool of available
instances.

This method is called in the transaction context of the remove operation.

**Throws:**

- **RemoveException** - The enterprise Bean does not allow destruction of the object.
- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the javax.ejb.EJBException instead of this exception.

```java
public void ejbActivate() throws EJBException, java.rmi.RemoteException
EJB
```

**Throws**

- **EJBException**:
  - java.rmi.RemoteException: EJB 1.0 Bean EJB

**Throws 1.1**

- Bean javax.ejb.EJBException EJB2.0
  - Bean javax.ejb.EJBException

**ejbActivate**

```java
void ejbActivate() throws EJBException, RemoteException
```

A container invokes this method when the instance is taken out of the pool of available instances to become associated with a specific
EJB object. This method transitions the instance to the ready state.

This method executes in an unspecified transaction context.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the javax.ejb.EJBException instead of this exception.

```java
public void ejbPassivate() throws EJBException, java.rmi.RemoteException
```

A container invokes this method on an instance before the instance becomes disassociated with a specific EJB object. After this method completes, the container will place the instance into the pool of available instances.
This method executes in an unspecified transaction context.

**Throws:**
- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the javax.ejb.EJBException instead of this exception.

```java
public void ejbLoad() throws EJBException, java.rmi.RemoteException
```

**A container invokes this method to instruct the instance to synchronize its state by loading it state from the underlying database.**

This method always executes in the transaction context determined by the value of the transaction attribute in the deployment descriptor.
Throws:

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the `javax.ejb.EJBException` instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.

```java
public void ejbStore() throws EJBException, java.rmi.RemoteException
```

A container invokes this method to instruct the instance to synchronize its state by storing it to the underlying database.

This method always executes in the transaction context determined by the value of the transaction attribute in the deployment descriptor.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
RemoteException - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the `javax.ejb.EJBException` instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS: 
public interface EntityBeanStats
extends EJBStats
Implements: EJBStats

Bean

Specifies statistics provided by entity beans.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>RangeStatistic</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>RangeStatistic</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.EJBStats
getCreateCount, getRemoveCount

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail
public RangeStatistic getReadyCount()
Bean

getReadyCount

RangeStatistic getReadyCount()

Number of beans in the ready state.

public RangeStatistic getPooledCount()
Bean

getPooledCount

RangeStatistic getPooledCount()

Number of beans in the pooled state.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.ejb  Interface EntityContext

All Superinterfaces:
   EJBContext

public interface EntityContext
   extends EJBContext

Implements: EJBContext

EntityContext  Bean  EntityContext  Bean

EntityContext  EntityContext  getPrimaryKey()
EJB

The EntityContext interface provides an instance with access to the container-provided runtime context of an entity enterprise Bean instance. The container passes the EntityContext interface to an entity enterprise Bean instance after the instance has been created.

The EntityContext interface remains associated with the instance for the lifetime of the instance. Note that the information that the instance obtains using the EntityContext interface (such as the result of the getPrimaryKey() method) may change, as the container assigns the instance to different EJB objects during the instance's life cycle.

---

Method Summary

<table>
<thead>
<tr>
<th>EJBLocalObject</th>
<th>getEJBLocalObject()</th>
<th>Obtain a reference to the EJB local object that is currently associated with the instance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJBObject</td>
<td>getEJBObject()</td>
<td>Obtain a reference to the EJB object that is currently</td>
</tr>
</tbody>
</table>
**getPrimaryKey()**

Obtain the primary key of the EJB object that is currently associated with this instance.

---

**Methods inherited from interface javax.ejb.EJBContext**

- `getCallerIdentity`, `getCallerPrincipal`, `getEJBHome`, `getEJBLocalHome`, `getEnvironment`, `getRollbackOnly`, `getTimerService`, `getUserTransaction`, `isCallerInRole`, `isCallerInRole`, `lookup`, `setRollbackOnly`

---

**Method Detail**

```java
public EJBLocalObject getEJBLocalObject() throws IllegalStateException

EJB Bean

ejbActivate

ejbPassivate

ejbPostCreate

ejbRemove

ejbLoad

ejbStore

return

Throws

IllegalStateException:

getEJBLocalObject

EJBLocalObject getEJBLocalObject() throws IllegalStateException

Obtain a reference to the EJB local object that is currently associated with the instance.

An instance of an entity enterprise Bean can call this method only
when the instance is associated with an EJB local object identity, i.e. in the ejbActivate, ejbPassivate, ejbPostCreate, ejbRemove, ejbLoad, ejbStore, and business methods.

An instance can use this method, for example, when it wants to pass a reference to itself in a method argument or result.

**Returns:**

The EJB local object currently associated with the instance.

**Throws:**

*IllegalStateException* - if the instance invokes this method while the instance is in a state that does not allow the instance to invoke this method, or if the instance does not have a local interface.

```
public EJBObject getEJBObject() throws IllegalStateException

EJB Bean EJB ejbActivate ejbPassivate ejbPostCreate ejbRemove ejbLoad ejbStore

return EJB

Throws IllegalStateException:
```

**getEJBObject**

```
EJBObject getEJBObject()

throws IllegalStateException

Obtain a reference to the EJB object that is currently associated with the instance.

An instance of an entity enterprise Bean can call this method only
```
when the instance is associated with an EJB object identity, i.e. in the ejbActivate, ejbPassivate, ejbPostCreate, ejbRemove, ejbLoad, ejbStore, and business methods.

An instance can use this method, for example, when it wants to pass a reference to itself in a method argument or result.

**Returns:**

The EJB object currently associated with the instance.

**Throws:**

*IllegalStateException* - Thrown if the instance invokes this method while the instance is in a state that does not allow the instance to invoke this method, or if the instance does not have a remote interface.

```java
public Object getPrimaryKey() throws IllegalStateException

EJB

Bean EJB

ejbActivate

ejbPassivate

ejbPostCreate

ejbRemove

ejbLoad

ejbStore

getEJBObject().getPrimaryKey()

return

Throws

IllegalStateException:

getPrimarykey

Object getPrimarykey()

throws IllegalStateException

Obtain the primary key of the EJB object that is currently associated with this instance.

An instance of an entity enterprise Bean can call this method only when the instance is associated with an EJB object identity, i.e. in
the ejbActivate, ejbPassivate, ejbPostCreate, ejbRemove, ejbLoad, ejbStore, and business methods.

**Note**: The result of this method is that same as the result of getEJBObject().getPrimaryKey.

**Returns**: The primary key currently associated with the instance.

**Throws**: 

- `IllegalStateException` - Thrown if the instance invokes this method while the instance is in a state that does not allow the instance to invoke this method.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream.events  **Interface EntityDeclaration**

**All Superinterfaces:**

[XMLEvent, XMLStreamConstants]

```java
public interface EntityDeclaration
extends XMLEvent

Implements: XMLEvent

version 1.0
```

An interface for handling Entity Declarations
This interface is used to record and report unparsed entity declarations.

**Version:**

1.0

**Author:**

Copyright (c) 2003 by BEA Systems. All Rights Reserved.

---

### Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants

| ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT |

### Method Summary

| [getBaseURI()]() |

---
**String**  Get the base URI for this reference or null if this information is not available

**String**  getPublicId()  The entity's public identifier, or null if none was given

**String**  getNotationName()  The name of the associated notation.

**String**  getSystemId()  The entity's system identifier.

**String**  getName()  The entity's name

**String**  getReplacementText()  The replacement text of the entity.

Methods inherited from interface javax.xml.stream.events.XMLEvent

asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

---

**Method Detail**

generic method: String getPublicId()

null

    return null

getPublicId

**String**  getPublicId()

The entity's public identifier, or null if none was given

Returns:
public String getSystemId()

        return ID null

getSystemId

String getSystemId()

The entity's system identifier.

Returns:
the system ID for this declaration or null

public String getName()

        return null

getName

String getName()

The entity's name

Returns:
the name, may not be null

public String getNotationName()

        return
getNotationName

String getNotationName()

The name of the associated notation.

Returns:
the notation name

public String getReplacementText()

null

return null

getReplacementText

String getReplacementText()

The replacement text of the entity. This method will only return non-null if this is an internal entity.

Returns:
null or the replacement text

public String getBaseURI()

URI null

return URI null

getBaseURI

String getBaseURI()

Get the base URI for this reference or null if this information is not available
Returns:
the base URI or null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence  Class EntityExistsException

javax.lang.Object
  java.lang.Throwable
    java.lang.Exception
      java.lang.RuntimeException
        javax.persistence.PersistenceException
          javax.persistence.EntityExistsException

All Implemented Interfaces:
  Serializable

public class EntityExistsException
  extends PersistenceException

Extends: Throwable > Exception > RuntimeException > PersistenceException

EntityManager.persist(Object)

  since    Java Persistence 1.0
  See also persist(Object)

Thrown by the persistence provider when EntityManager.persist(Object) is called and the entity already exists. The current transaction, if one is active, will be marked for rollback.

Since:
  Java Persistence 1.0

See Also:
  EntityManager.persist(Object), Serialized Form

Constructor Summary

EntityExistsException()
  Constructs a new EntityExistsException exception with null as
EntityExistsException(String message)

Constructs a new EntityExistsException exception with the specified detail message.

EntityExistsException(String message, Throwable cause)

Constructs a new EntityExistsException exception with the specified detail message and cause.

EntityExistsException(Throwable cause)

Constructs a new EntityExistsException exception with the specified cause.

### Method Summary

Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

public EntityExistsException()

null EntityExistsException

EntityExistsException

public EntityExistsException()  

Constructs a new EntityExistsException exception with null as its detail message.
public EntityExistsException(String message)
    throws IllegalArgumentException

    Constructs a new EntityExistsException exception with the specified detail message.

    Parameters:
    message - the detail message.

public EntityExistsException(String message, Throwable cause)
    throws IllegalArgumentException

    Constructs a new EntityExistsException exception with the specified detail message and cause.

    Parameters:
    message - the detail message.
    cause - the cause.
public EntityExistsException(Throwables cause)

Constructs a new EntityExistsException exception with the specified cause.

Parameters:
cause - the cause.
javax.persistence  Annotation Type EntityListeners

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface EntityListeners

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

since Java Persistence 1.0

Specifies the callback listener classes to be used for an entity or mapped superclass. This annotation may be applied to an entity class or mapped superclass.

**Since:**
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>Class[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The callback listener classes</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public Class<T>[] value()

value
public abstract Class[] value

The callback listener classes
javax.persistence  **Interface EntityManager**

```java
public interface EntityManager
```

**EntityManager**  **EntityManager API**

```java
since Java Persistence 1.0
```

Interface used to interact with the persistence context.

An EntityManager instance is associated with a persistence context. A persistence context is a set of entity instances in which for any persistent entity identity there is a unique entity instance. Within the persistence context, the entity instances and their lifecycle are managed. This interface defines the methods that are used to interact with the persistence context. The EntityManager API is used to create and remove persistent entity instances, to find entities by their primary key, and to query over entities.

The set of entities that can be managed by a given EntityManager instance is defined by a persistence unit. A persistence unit defines the set of all classes that are related or grouped by the application, and which must be colocated in their mapping to a single database.

**Since:**
Java Persistence 1.0

---

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clear()</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td><code>void</code></td>
</tr>
<tr>
<td><code>void close()</code></td>
</tr>
<tr>
<td><code>boolean contains(Object entity)</code></td>
</tr>
<tr>
<td><code>Query createNamedQuery(String name)</code></td>
</tr>
<tr>
<td><code>Query createNativeQuery(String sqlString)</code></td>
</tr>
<tr>
<td><code>Query createNativeQuery(String sqlString, Class resultClass)</code></td>
</tr>
<tr>
<td><code>Query createNativeQuery(String sqlString, String resultSetMapping)</code></td>
</tr>
<tr>
<td><code>Query createQuery(String qlString)</code></td>
</tr>
</tbody>
</table>

`find(Class<T> entityClass, Object primaryKey)`
Find by primary key.  
Synchronize the persistence context to the underlying database.

`Object getDelegate()`
Return the underlying provider object for the EntityManager, if available.

`FlushModeType getFlushMode()`
Get the flush mode that applies to all objects contained in the persistence context.

`getReference(Class<T> entityClass, Object primaryKey)`
Get an instance, whose state may be lazily fetched.

`EntityTransaction`  
Returns the resource-level transaction object.

`boolean isOpen()`
Determine whether the EntityManager is open.  

```java
void joinTransaction()
```

Indicate to the EntityManager that a JTA transaction is active.  

```java
void lock(Object entity, LockModeType lockMode)
```

Set the lock mode for an entity object contained in the persistence context.

```java
<T> T merge(T entity)
```

Merge the state of the given entity into the current persistence context.

```java
void persist(Object entity)
```

Make an entity instance managed and persistent.

```java
void refresh(Object entity)
```

Refresh the state of the instance from the database, overwriting changes made to the entity, if any.

```java
void remove(Object entity)
```

Remove the entity instance.

```java
void setFlushMode(FlushModeType flushMode)
```

Set the flush mode that applies to all objects contained in the persistence context.

---

**Method Detail**

**public void persist(Object entity)**

- `entity`  
- Throws `EntityExistsException`: EntityExistsException  
- Throws `PersistenceException`: EntityExistsException  
- Throws `IllegalArgumentException`: IllegalStateException: EntityManager  
- Throws `TransactionRequiredException`: PersistenceContextType.TRANSACTION

**persist**

```java
void persist(Object entity)
```

Make an entity instance managed and persistent.
Parameters:
   entity -

Throws:
   EntityExistsException  - if the entity already exists. (The EntityExistsException may be thrown when the persist operation is invoked, or the EntityExistsException or another PersistenceException may be thrown at flush or commit time.)
   IllegalStateException  - if this EntityManager has been closed.
   IllegalArgumentException  - if not an entity
   TransactionRequiredException  - if invoked on a container-managed entity manager of type PersistenceContextType.TRANSACTION and there is no transaction.

merge

<T> T merge(T entity)

Merge the state of the given entity into the current persistence context.

Parameters:
   entity -

Returns:
   the instance that the state was merged to

Throws:
   IllegalStateException  - if this EntityManager has been closed.
   IllegalArgumentException  - if instance is not an entity or is a removed entity
   TransactionRequiredException  - if invoked on a container-managed entity manager of type PersistenceContextType.TRANSACTION and there is no transaction.

public void remove(Object entity)
remove

```java
void remove(Object entity)
```

Remove the entity instance.

**Parameters:**
- `entity` -

**Throws:**
- `IllegalStateException` - if this EntityManager has been closed.
- `IllegalArgumentException` - if not an entity or if a detached entity
- `TransactionRequiredException` - if invoked on a container-managed entity manager of type `PersistenceContextType.TRANSACTION` and there is no transaction.

find

```java
<T> T find(Class<T> entityClass, Object primaryKey)
```

Find by primary key.

**Parameters:**
- `entityClass` -
- `primaryKey` -

**Returns:**
the found entity instance or null if the entity does not exist
Throws:

- `IllegalArgumentException` - if the first argument does not denote an entity type or the second argument is not a valid type for that entity's primary key
- `IllegalStateException` - if this EntityManager has been closed.

getReference

```java
<T> T getReference(Class<T> entityClass, Object primaryKey)
```

Get an instance, whose state may be lazily fetched. If the requested instance does not exist in the database, throws `EntityNotFoundException` when the instance state is first accessed. (The persistence provider runtime is permitted to throw `EntityNotFoundException` when `getReference(java.lang.Class, java.lang.Object)` is called.) The application should not expect that the instance state will be available upon detachment, unless it was accessed by the application while the entity manager was open.

Parameters:
- `entityClass` - the entity type
- `primaryKey` - the primary key

Returns:
- the found entity instance

Throws:

- `IllegalArgumentException` - if the first argument does not denote an entity type or the second argument is not a valid type for that entity's primary key
- `IllegalStateException` - if this EntityManager has been closed.
- `EntityNotFoundException` - if the entity state cannot be accessed

public void flush()}

Throws

- `IllegalStateException`: EntityManager
flush

void flush()

Synchronize the persistence context to the underlying database.

Throws:

- `IllegalStateException` - if this EntityManager has been closed.
- `TransactionRequiredException` - if there is no transaction
- `PersistenceException` - if the flush fails

public void setFlushMode(FlushModeType flushMode)

flushMode

Throws `IllegalStateException`: EntityManager

setFlushMode

void setFlushMode(FlushModeType flushMode)

Set the flush mode that applies to all objects contained in the persistence context.

Parameters:

flushMode -

Throws:

- `IllegalStateException` - if this EntityManager has been closed.

public FlushModeType getFlushMode()

return
getFlushMode

FlushModeType getFlushMode()

Get the flush mode that applies to all objects contained in the persistence context.

Returns:
  flush mode

Throws:
  EntityManager

public void lock(Object entity, LockModeType lockMode)

entity
lockMode

Throws:  EntityManager
Throws:  PersistenceException:
Throws:  IllegalArgumentException:
Throws:  TransactionRequiredException:

lock

void lock(Object entity, LockModeType lockMode)

Set the lock mode for an entity object contained in the persistence context.

Parameters:
  entity -
  lockMode -

Throws:
**IllegalStateException** - if this EntityManager has been closed.

**PersistenceException** - if an unsupported lock call is made

**IllegalArgumentException** - if the instance is not an entity or is a detached entity

**TransactionRequiredException** - if there is no transaction

---

**public void refresh(Object entity)**

- **entity**
- **Throws**
  - `IllegalStateException`: EntityManager
  - `IllegalArgumentException`: 
  - `TransactionRequiredException`: PersistenceContextType.TRANSACTION
  - `EntityNotFoundException`: 

---

**refresh**

**void refresh(Object entity)**

Refresh the state of the instance from the database, overwriting changes made to the entity, if any.

**Parameters:**
- **entity**

**Throws:**
- `IllegalStateException` - if this EntityManager has been closed.
- `IllegalArgumentException` - if not an entity or entity is not managed
- `TransactionRequiredException` - if invoked on a container-managed entity manager of type PersistenceContextType.TRANSACTION and there is no transaction.
- `EntityNotFoundException` - if the entity no longer exists in the database.

---
public void clear()

Throws: IllegalStateException - EntityManager

clear

void clear()

Clear the persistence context, causing all managed entities to become detached. Changes made to entities that have not been flushed to the database will not be persisted.

Throws:
IllegalArgumentException - if not an entity

public boolean contains(Object entity)

entity
return true

Throws: IllegalStateException: EntityManager

contains

boolean contains(Object entity)

Check if the instance belongs to the current persistence context.

Parameters:
entity -

Returns:
true if the instance belongs to the current persistence context.

Throws:
IllegalArgumentException - if not an entity
public Query createQuery(String qlString)

Create an instance of Query for executing a Java Persistence query language statement.

Parameters:
qlString - a Java Persistence query language query string

Returns:
the new query instance

Throws:
IllegalStateException - if this EntityManager has been closed.
IllegalArgumentException - if query string is not valid

createNamedQuery

Query createNamedQuery(String name)

Create a named query using the specified SQL query statement.

Parameters:
name - SQL query statement

Returns:
the new query instance

Throws:
IllegalStateException - if this EntityManager has been closed.
IllegalArgumentException - if query string is not valid
Create an instance of Query for executing a named query (in the Java Persistence query language or in native SQL).

**Parameters:**
- name - the name of a query defined in metadata

**Returns:**
- the new query instance

**Throws:**
- `IllegalStateException` - if this EntityManager has been closed.
- `IllegalArgumentException` - if a query has not been defined with the given name

```java
public Query createNativeQuery(String sqlString)
```

Create an instance of Query for executing a native SQL statement, e.g., for update or delete.

**Parameters:**
- sqlString - a native SQL query string

**Returns:**
- the new query instance

**Throws:**
- `IllegalStateException` - if this EntityManager has been closed.

```java
public Query createNativeQuery(String sqlString, Class<T> resultClass)
```
Creates an instance of Query for executing a native SQL query.

Parameters:
- `sqlString` - a native SQL query string
- `resultSetMapping` - the class of the resulting instance(s)

Returns:
the new query instance

Throws:
- `IllegalStateException` - if this EntityManager has been closed.

```java
public Query createNativeQuery(String sqlString, String resultSetMapping)
```

Create an instance of Query for executing a native SQL query.

```java
Query createNativeQuery(String sqlString, String resultSetMapping)
```
**Parameters:**
- sqlString - a native SQL query string
- resultSetMapping - the name of the result set mapping

**Returns:**
the new query instance

**Throws:**
- **IllegalStateException** - if this EntityManager has been closed.

---

**public void joinTransaction()**

Indicate to the EntityManager that a JTA transaction is active. This method should be called on a JTA application managed EntityManager that was created outside the scope of the active transaction to associate it with the current JTA transaction.

**Throws:**
- **IllegalStateException** - if this EntityManager has been closed.
- **TransactionRequiredException** - if there is no transaction.

---

**public Object getDelegate()**

**Throws**
- **IllegalStateException** - if this EntityManager has been closed.
Object getDelegate()

Return the underlying provider object for the EntityManager, if available. The result of this method is implementation specific.

Throws:
   IllegalStateException - if this EntityManager has been closed.

public void close()
EntityManager close EntityManager Query
getTransaction isOpen false
IllegalStateException EntityManager
   Throws  IllegalStateException: EntityManager

close

void close()

Close an application-managed EntityManager. After the close method has been invoked, all methods on the EntityManager instance and any Query objects obtained from it will throw the IllegalStateException except for getTransaction and isOpen (which will return false). If this method is called when the EntityManager is associated with an active transaction, the persistence context remains managed until the transaction completes.

Throws:
   IllegalStateException - if the EntityManager is container-managed or has been already closed..

public boolean isOpen()
EntityManager
   return EntityManager true
boolean isOpen()

Determine whether the EntityManager is open.

Returns: true until the EntityManager has been closed.

public EntityTransaction getTransaction()

EntityTransaction

return EntityTransaction

Throws IllegalArgumentException: JTA EntityManager

getTransaction

EntityTransaction getTransaction()

Returns the resource-level transaction object. The EntityTransaction instance may be used serially to begin and commit multiple transactions.

Returns: EntityTransaction instance

Throws: IllegalArgumentException - if invoked on a JTA EntityManager.
PS:
The `EntityManagerFactory` interface is used by the application to obtain an application-managed entity manager. When the application has finished using the entity manager factory, and/or at application shutdown, the application should close the entity manager factory. Once an `EntityManagerFactory` has been closed, all its entity managers are considered to be in the closed state.

**Since:**
Java Persistence 1.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>close()</strong></td>
</tr>
<tr>
<td><code>createEntityManager()</code></td>
</tr>
<tr>
<td><code>createEntityManager(Map map)</code></td>
</tr>
<tr>
<td><strong>isOpen()</strong></td>
</tr>
</tbody>
</table>
Method Detail

```
public EntityManager createEntityManager()
EntityManager EntityManager isOpen true

createEntityManager

EntityManager createEntityManager()

Create a new EntityManager. This method returns a new EntityManager instance each time it is invoked. The isOpen method will return true on the returned instance.

public EntityManager createEntityManager(java.util.Map<K, V> map)
Map EntityManager EntityManager isOpen true

createEntityManager

EntityManager createEntityManager(Map map)

Create a new EntityManager with the specified Map of properties. This method returns a new EntityManager instance each time it is invoked. The isOpen method will return true on the returned instance.

public void close()
IllegalStateException isOpen false
EntityManagerFactory
```
close

void close()

Close the factory, releasing any resources that it holds. After a factory instance is closed, all methods invoked on it will throw an IllegalStateException, except for isOpen, which will return false. Once an EntityManagerFactory has been closed, all its entity managers are considered to be in the closed state.

public boolean isOpen()

Indicates whether or not this factory is open. Returns true until a call to close has been made.
javax.persistence  **Class EntityNotFoundException**

**java.lang.Object**

  ▼ **java.lang.Throwable**
    ▼ **java.lang.Exception**
      ▼ **java.lang.RuntimeException**
        ▼ **javax.persistence.PersistenceException**
          ▼ **javax.persistence.EntityNotFoundException**

**All Implemented Interfaces:**

  [Serializable](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/Serializable.html)

---

```java
public class EntityNotFoundException extends PersistenceException
```

**Extends:** Throwable > Exception > RuntimeException > PersistenceException

- `getReference(EntityManager.getReference(Class, Object))`
- `refresh(EntityManager.refresh(Object))`

**since** Java Persistence 1.0

**See also** `getReference(Class, Object)`, `refresh(Object)`

Thrown by the persistence provider when an entity reference obtained by `EntityManager.getReference(Class, Object)` is accessed but the entity does not exist. Also thrown when `EntityManager.refresh(Object)` is called and the object no longer exists in the database. The current transaction, if one is active, will be marked for rollback.

**Since:** Java Persistence 1.0

**See Also:**

  `EntityManager.getReference(Class, Object)`, `EntityManager.refresh(Object)`, [Serialized Form](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/Serializable.html)
Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EntityNotFoundException()</code></td>
<td>Constructs a new EntityNotFoundException exception with null as its detail message.</td>
</tr>
<tr>
<td><code>EntityNotFoundException(String message)</code></td>
<td>Constructs a new EntityNotFoundException exception with the specified detail message.</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

```java
public EntityNotFoundException()
null EntityNotFoundException
```

```java
public EntityNotFoundException()
Constructs a new EntityNotFoundException exception with null as its detail message.
```
public EntityNotFoundException(String message)

EntityNotFoundException
message

Constructs a new EntityNotFoundException exception with the specified detail message.

Parameters:
message - the detail message.
javax.xml.stream.events Interface EntityReference

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface EntityReference
extends XMLEvent

Implements: XMLEvent

javax.xml.stream.isReplacingEntityReferences false
javax.xml.stream.isReplacingEntityReferences true
javax.xml.stream.isReplacingEntityReferences true(2)
javax.xml.stream.isReplacingEntityReferences false
EntityReference

   version 1.0

An interface for handling Entity events. This event reports entities that
have not been resolved and reports their replacement text unprocessed
(if available). This event will be reported if
javax.xml.stream.isReplacingEntityReferences is set to false. If
javax.xml.stream.isReplacingEntityReferences is set to true entity
references will be resolved transparently. Entities are handled in two
possible ways: (1) If javax.xml.stream.isReplacingEntityReferences is set
to true all entity references are resolved and reported as markup
transparently. (2) If javax.xml.stream.isReplacingEntityReferences is set
to false Entity references are reported as an EntityReference Event.

Version:
   1.0

Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.
### Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants

ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT

### Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityDeclaration</td>
<td>getDeclaration()</td>
<td>Return the declaration of this entity.</td>
</tr>
<tr>
<td>String</td>
<td>getName()</td>
<td>The name of the entity</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public EntityDeclaration getDeclaration()
```

**getDeclaration**

```java
EntityDeclaration getDeclaration()
```

Return the declaration of this entity.
public String getName()

        return null

getName

String getName()

The name of the entity

Returns:
    the entity's name, may not be null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.persistence\n
Annotation Type EntityResult

@Target(value={})
@Retention(value=RUNTIME)
public @interface EntityResult

Implements: Annotation
@Target(value={})
@Retention(value=RUNTIME)

SQL SELECT SQL

Query q = em.createNativeQuery(
    "SELECT o.id, o.quantity, o.item, i.id, i.name, i.description
    "FROM Order o, Item i " +
    "WHERE (o.quantity > 25) AND (o.item = i.id)",
    "OrderItemResults");
@SqlResultSetMapping(name="OrderItemResults",
    entities={
        @EntityResult(entityClass=com.acme.Order.class),
        @EntityResult(entityClass=com.acme.Item.class)
    })

since Java Persistence 1.0

References an entity in the SELECT clause of a SQL query. If this annotation is used, the SQL statement should select all of the columns that are mapped to the entity object. This should include foreign key columns to related entities. The results obtained when insufficient data is available are undefined.

Example
Query q = em.createNativeQuery(
    "SELECT o.id, o.quantity, o.item, i.id, i.name, i.description
    "FROM Order o, Item i " +
    "WHERE (o.quantity > 25) AND (o.item = i.id)",
    "OrderItemResults");
@SqlResultSetMapping(name="OrderItemResults",
    entities={

@EntityResult(entityClass=com.acme.Order.class),
@EntityResult(entityClass=com.acme.Item.class)
}

Since:
Java Persistence 1.0

---

## Required Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>entityClass</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The class of the result</td>
</tr>
</tbody>
</table>

## Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>discriminatorColumn</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Specifies the column name (or alias) of the column in the SELECT list that is used to determine the type of the entity instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FieldResult[]</th>
<th>fields</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Maps the columns specified in the SELECT list of the query to the properties or fields of the entity class.</td>
</tr>
</tbody>
</table>

## Element Detail

```java
abstract public Class<T> entityClass()
```

**entityClass**

```java
public abstract Class entityClass
```

The class of the result
abstract public FieldResult[] fields()

SELECT

fields

public abstract FieldResult[] fields

Maps the columns specified in the SELECT list of the query to the properties or fields of the entity class.

Default:

{}

abstract public String discriminatorColumn()

SELECT

discriminatorColumn

public abstract String discriminatorColumn

Specifies the column name (or alias) of the column in the SELECT list that is used to determine the type of the entity instance.

Default:

""
PS:
javax.persistence Interface EntityTransaction

public interface EntityTransaction

EntityTransaction getTransaction
EntityManager.getTransaction() EntityTransaction

since Java Persistence 1.0 en

The EntityTransaction interface is used to control resource transactions on resource-local entity managers. The EntityManager.getTransaction() method returns the EntityTransaction interface.

Since:
Java Persistence 1.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>begin()</strong></td>
</tr>
<tr>
<td><strong>commit()</strong></td>
</tr>
<tr>
<td><strong>getRollbackOnly()</strong></td>
</tr>
<tr>
<td><strong>isActive()</strong></td>
</tr>
<tr>
<td><strong>rollback()</strong></td>
</tr>
<tr>
<td><strong>setRollbackOnly()</strong></td>
</tr>
</tbody>
</table>
Method Detail

public void begin()

   Throws           IllegalStateException:   isActive() true

begin

void begin()

Start the resource transaction.

Throws:
IllegalStateException - if isActive() is true.

public void commit()

   Throws           IllegalStateException:   isActive() false
   Throws         RollbackException:

commit

void commit()

Commit the current transaction, writing any unflushed changes to the database.

Throws:
IllegalStateException - if isActive() is false.
RollbackException - if the commit fails.
public void rollback()

Throws IllegalStateException: #isActive() false
Throws PersistenceException:

rollback

void rollback()

Roll back the current transaction

Throws:
IllegalStateException - if isActive() is false.
PersistenceException - if an unexpected error condition is encountered.

public void setRollbackOnly()

Throws IllegalStateException: #isActive() false

setRollbackOnly

void setRollbackOnly()

Mark the current transaction so that the only possible outcome of the transaction is for the transaction to be rolled back.

Throws:
IllegalStateException - if isActive() is false.

public boolean getRollbackOnly()

Throws IllegalStateException: #isActive() false
**getRollbackOnly**

boolean `getRollbackOnly()`

Determine whether the current transaction has been marked for rollback.

**Throws:**

`IllegalStateException` - if `isActive()` is false.

---

**public boolean isActive()**

Throws `PersistenceException`:

---

**isActive**

boolean `isActive()`

Indicate whether a transaction is in progress.

**Throws:**

`PersistenceException` - if an unexpected error condition is encountered.
PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: NESTED</th>
<th>FRAME</th>
<th>NO FRAME</th>
<th>DETAIL</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**javax.faces.convert** Class EnumConverter

**java.lang.Object**

- javax.faces.convert.EnumConverter

**All Implemented Interfaces:**
  - StateHolder, Converter

---

```java
public class EnumConverter

extends Object

implements Converter, StateHolder

```

**Implements:** Converter, StateHolder

---

```java
java.lang.Enum enum

Converter

since 1.2

```

Converter implementation for java.lang.Enum (and enum primitive) values.

**Since:**

1.2

---

### Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>CONVERGE_ID</th>
<th>The standard converter id for this converter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String enum</td>
<td>ENUM_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion to Enum fails.</td>
</tr>
<tr>
<td>static String enum</td>
<td>ENUM_NO_CLASS_ID</td>
<td>The message identifier of the FacesMessage to be</td>
</tr>
</tbody>
</table>
```java
static String created if the conversion to Enum fails and no target class has been provided.

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnumConverter()</td>
</tr>
<tr>
<td>EnumConverter(Class targetClass)</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object getAsObject(FacesContext context, UICOMPONENT component, String value)</td>
<td>Convert the <code>value</code> argument to one of the enum constants of the class provided in our constructor.</td>
<td></td>
</tr>
<tr>
<td>String getAsString(FacesContext context, UICOMPONENT component, Object value)</td>
<td>Convert the enum constant given by the <code>value</code> argument into a String.</td>
<td></td>
</tr>
<tr>
<td>boolean isTransient()</td>
<td></td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void restoreState(FacesContext facesContext, Object object)</td>
<td></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object saveState(FacesContext facesContext)</td>
<td></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void setTransient(boolean b)</td>
<td></td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

class, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
```
CONVERTER_ID

public static final String CONVERTER_ID

The standard converter id for this converter.

See Also:
Constant Field Values

ENUM_ID

public static final String ENUM_ID

The message identifier of the FacesMessage to be created if the conversion to Enum fails. The message format string for this message may optionally include the following placeholders:

- {0} replaced by the unconverted value.
- {1} replaced by one of the enum constants or the empty string if none can be found.
- {2} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

ENUM_NO_CLASS_ID

public static final String ENUM_NO_CLASS_ID
The message identifier of the `FacesMessage` to be created if the conversion to `Enum` fails and no target class has been provided. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a string whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

### Constructor Detail

**public EnumConverter()**

**EnumConverter**

**public EnumConverter()**

**public EnumConverter(Class<T> targetClass)**

**EnumConverter**

**public EnumConverter(Class targetClass)**

### Method Detail

**public Object getAsObject(FacesContext context, UIComponent component, String value)**

| value | 
|--------|--------|--------|--------|--------|
| null 0 | #ENUM_NO_CLASS_ID | null value | ConverterException | Enum.valueOf |
getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

Convert the value argument to one of the enum constants of the class provided in our constructor. If no target class argument has been provided to the constructor of this instance, throw a ConverterException containing the ENUM_NO_CLASS_ID message with proper parameters. If the value argument is null or it has a length of zero, return null. Otherwise, perform the equivalent of Enum.valueOf using target class and value and return the Object. If the conversion fails, throw a ConverterException containing the ENUM_ID message with proper parameters.

Specified by:
getAsObject in interface Converter

Parameters:
context - the FacesContext for this request.
component - the UIComponent to which this value will be applied.
value - the String value to be converted to Object.

Returns:
null if the value to convert is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null
public String getAsString(FacesContext context, UIComponent component, Object value)

value String #ENUM_NO_CLASS_ID
ConverterException value null null
(value.toString()) ConverterException #ENUM_ID

Throws ConverterException: NullPointerExcpetion
Throws NullPointerExcpetion: NullPointerExcpetion context
component null

gAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Convert the enum constant given by the value argument into a String. If no target class argument has been provided to the constructor of this instance, throw a ConverterException containing the ENUM_NO_CLASS_ID message with proper parameters. If the value, argument is null, return null. If the value is an instance of the provided target class, return its string value (value.toString()). Otherwise, throw a ConverterException containing the ENUM_ID message with proper parameters.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion
Throws:

- `ConverterException` - if conversion cannot be successfully performed
- `NullPointerException` - if context or component is null

public void restoreState(FacesContext facesContext, Object object)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

Specified by:

`restoreState` in interface StateHolder

public Object saveState(FacesContext facesContext)

saveState

public Object saveState(FacesContext facesContext)

Description copied from interface: StateHolder
Gets the state of the instance as a **Serializable** Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a **UIComponent** with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**

`saveState` in interface `StateHolder`

---

**public void setTransient(boolean b)**

**setTransient**

```
public void setTransient(boolean b)
```

**Description copied from interface: **`StateHolder`

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Specified by:**

`setTransient` in interface `StateHolder`

**Parameters:**

- `b` - boolean pass `true` if this Object will participate in state saving or restoring, otherwise pass `false`.  

public boolean isTransient()

isTransient

public boolean isTransient()

Description copied from interface: StateHolder

If true, the Object implementing this interface must not participate in state saving or restoring.

Specified by:

isTransient in interface StateHolder
javax.persistence  Annotation Type Enumerated

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public interface Enumerated

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

public enum EmployeeStatus {FULL_TIME, PART_TIME, CONTRACT}

public enum SalaryRate {JUNIOR, SENIOR, MANAGER, EXECUTIVE}

@Entity
public class Employee {
    public EmployeeStatus getStatus() {...}
    @Enumerated(STRING)
    public SalaryRate getPayScale() {...}
}

since Java Persistence 1.0

Specifies that a persistent property or field should be persisted as an enumerated type. It may be used in conjunction with the 
Basic annotation.

Example:

public enum EmployeeStatus {FULL_TIME, PART_TIME, CONTRACT}

public enum SalaryRate {JUNIOR, SENIOR, MANAGER, EXECUTIVE}

@Entity
public class Employee {
    public EmployeeStatus getStatus() {...}
    @Enumerated(STRING)
    public SalaryRate getPayScale() {...}
Since:
Java Persistence 1.0

**Optional Element Summary**

<table>
<thead>
<tr>
<th>EnumType</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) The type used in mapping an enum type.</td>
</tr>
</tbody>
</table>

**abstract public** `EnumType` **value()**

**value**

`public abstract EnumType value`  

(Optional) The type used in mapping an enum type.

**Default:**  
ORDINAL

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

---

PS:
javax.persistence Enum EnumType

java.lang.Object
  ↓ java.lang.Enum<EEnumType>
  ↓ javax.persistence.EnumType

All Implemented Interfaces:
      Serializable, Comparable<EnumType>

public enum EnumType
  extends Enum<EEnumType>

Extends: Enum<E>

since Java Persistence 1.0

Defines mapping for the enumerated types. The constants of this enumerated type specify how persistent property or field should be persisted as an enumerated type.

Since:
      Java Persistence 1.0

---

**Enum Constant Summary**

| ORDINAL | Persist enumerated type property or field as an integer |
| STRING  | Persist enumerated type property or field as a string |

---

**Method Summary**

<p>| valueOf(String name) |</p>
<table>
<thead>
<tr>
<th>static EnumType</th>
<th>Returns the enum constant of this type with the specified name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static EnumType[]</td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

ORDINAL

public static final EnumType ORDINAL

Persist enumerated type property or field as an integer

STRING

public static final EnumType STRING

Persist enumerated type property or field as a string

Method Detail

final public static EnumType[] values()
values

public static final EnumType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

for(EnumType c : EnumType.values())
    System.out.println(c);

Returns:
an array containing the constants of this enum type, in the order they're declared

valueOf

public static EnumType valueOf(String name)

valueOf

public static EnumType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
name - the name of the enum constant to be returned.

Returns:
the enum constant with the specified name

Throws:
IllegalArgumentException - if this enum type has no constant with the specified name
javax.servlet.jsp Class ErrorData

java.lang.Object
   \--- javax.servlet.jsp.ErrorData

public final class ErrorData
extends Object

JSP isErrorPage "true"
   since 2.0
   See also getErrorData

Contains information about an error, for error pages. The information contained in this instance is meaningless if not used in the context of an error page. To indicate a JSP is an error page, the page author must set the isErrorPage attribute of the page directive to "true".

Since:
   JSP 2.0
See Also:
   PageContext.getErrorData()

Constructor Summary

| ErrorData( | Throwable throwable, int statusCode, String uri, String servletName) |
| Creates a new ErrorData object. |

Method Summary

<p>| String getRequestURI() |
| Returns the request URI. |
| String getServletName() |</p>
<table>
<thead>
<tr>
<th>String</th>
<th>Returns the name of the servlet invoked.</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>getStatusCode()</td>
</tr>
<tr>
<td></td>
<td>Returns the status code of the error.</td>
</tr>
<tr>
<td>Throwable</td>
<td>getThrowable()</td>
</tr>
<tr>
<td></td>
<td>Returns the Throwable that caused the error.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
clon, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

```java
public ErrorData(Throwable throwable, int statusCode, String uri, String servletName)
```

Creates a new ErrorData object.

**Parameters:**

- `throwable` - The Throwable that is the cause of the error
- `statusCode` - The status code of the error
- `uri` - The request URI
- `servletName` - The name of the servlet invoked

**ErrorData**

```java
public ErrorData(Throwable throwable, int statusCode, String uri, String servletName)
```
public Throwable getThrowable()
  Throwable

  return Throwable

getThrowable

public Throwable getThrowable()

  Returns the Throwable that caused the error.

  Returns:
    The Throwable that caused the error

getStatusCode

public int getStatusCode()

  return

getStatusCode

public int getStatusCode()

  Returns the status code of the error.

  Returns:
    The status code of the error

getRequestURI

public String getRequestURI()
  URI

  return URI
getRequestURI

public String getRequestURI()  

Returns the request URI.

Returns: 
The request URI

getServletName

public String getServletName()  
servlet 

return servlet

getServletName

public String getServletName()  

Returns the name of the servlet invoked.

Returns: 
The name of the servlet invoked
javax.faces.el  Class EvaluationException

java.lang.Object  
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.faces.FacesException
                  └ javax.faces.el.EvaluationException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  MethodNotFoundException, PropertyNotFoundException, ReferenceSyntaxException

Deprecated. This has been replaced by ELException.

Extends: Throwable > Exception > RuntimeException > FacesException
Extended by: MethodNotFoundException, PropertyNotFoundException, ReferenceSyntaxException

MethodBinding    ValueBinding

deprecated       javax.el.ELException

class EvaluationException

extends FacesException

An exception reporting an error that occurred during the evaluation of an expression in a MethodBinding or ValueBinding.

See Also:
  Serialized Form
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>EvaluationException()</code></td>
<td><strong>Deprecated.</strong> Construct a new exception with no detail message or root cause.</td>
</tr>
<tr>
<td><code>EvaluationException(String message)</code></td>
<td><strong>Deprecated.</strong> Construct a new exception with the specified detail message and no root cause.</td>
</tr>
<tr>
<td><code>EvaluationException(String message, Throwable cause)</code></td>
<td><strong>Deprecated.</strong> Construct a new exception with the specified detail message and root cause.</td>
</tr>
<tr>
<td><code>EvaluationException(Throwable cause)</code></td>
<td><strong>Deprecated.</strong> Construct a new exception with the specified root cause.</td>
</tr>
</tbody>
</table>

### Method Summary

#### Methods inherited from class `javax.faces.FacesException`
- `getCause`

#### Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

#### Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
EvaluationException

public EvaluationException()

    Deprecated.

    Construct a new exception with no detail message or root cause.

public EvaluationException(String message)

    message

EvaluationException

public EvaluationException(String message)

    Deprecated.

    Construct a new exception with the specified detail message and no root cause.

    Parameters:
    message - The detail message for this exception

public EvaluationException(Throwable cause)

    (cause == null ? null : cause.toString())
    cause
EvaluationException

public EvaluationException(Throwable cause)

    Deprecated.

    Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

    Parameters:
    cause - The root cause for this exception

public EvaluationException(String message, Throwable cause)

    message
    cause

EvaluationException

public EvaluationException(String message, Throwable cause)

    Deprecated.

    Construct a new exception with the specified detail message and root cause.

    Parameters:
    message - The detail message for this exception
    cause - The root cause for this exception
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream  Interface EventFilter

public interface EventFilter

XMLEventReader

version 1.0

This interface declares a simple filter interface that one can create to filter XMLEventReaders

Version:
  1.0
Author:
  Copyright (c) 2003 by BEA Systems. All Rights Reserved.

### Method Summary

<table>
<thead>
<tr>
<th>boolean accept(XMLEvent event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tests whether this event is part of this stream.</td>
</tr>
</tbody>
</table>

### Method Detail

public boolean accept(XMLEvent event)
true false
event
return true false

accept
boolean accept(XMLEvent event)

Tests whether this event is part of this stream. This method will return true if this filter accepts this event and false otherwise.

**Parameters:**

   event - the event to test

**Returns:**

   true if this filter accepts this event, false otherwise
javax.xml.stream.util  Class EventReaderDelegate

java.lang.Object
  └─javax.xml.stream.util.EventReaderDelegate

All Implemented Interfaces:
   Iterator, XMLEventReader

public class EventReaderDelegate
  extends Object
  implements XMLEventReader

Implements: XMLEventReader

XMLEventReader  XMLEventReader  XMLEventReader

version  1.0

See   javax.xml.stream.XMLEventReader,
also   javax.xml.stream.util.StreamReaderDelegate

This is the base class for deriving an XMLEventReader filter. This class is
designed to sit between an XMLEventReader and an application's
XMLEventReader. By default each method does nothing but call the
corresponding method on the parent interface.

Version:
  1.0
Author:
  Copyright (c) 2003 by BEA Systems. All Rights Reserved.
See Also:
   XMLEventReader, StreamReaderDelegate

Constructor Summary
### EventReaderDelegate()
Construct an empty filter with no parent.

### EventReaderDelegate(XMLEventReader reader)
Construct an empty filter with the specified parent.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void close()</td>
<td>Frees any resources associated with this Reader.</td>
</tr>
<tr>
<td>String getElementText()</td>
<td>Reads the content of a text-only element.</td>
</tr>
<tr>
<td>XMLEventReader getParent()</td>
<td>Get the parent of this instance.</td>
</tr>
<tr>
<td>Object getProperty(String name)</td>
<td>Get the value of a feature/property from the underlying implementation</td>
</tr>
<tr>
<td>boolean hasNext()</td>
<td>Check if there are more events.</td>
</tr>
<tr>
<td>Object next()</td>
<td></td>
</tr>
<tr>
<td>XMLEvent nextEvent()</td>
<td>Get the next XMLEvent</td>
</tr>
<tr>
<td>XMLEvent nextTag()</td>
<td>Skips any insignificant space events until a START_ELEMENT or END_ELEMENT is reached.</td>
</tr>
<tr>
<td>XMLEvent peek()</td>
<td>Check the next XMLEvent without reading it from the stream.</td>
</tr>
<tr>
<td>void remove()</td>
<td></td>
</tr>
<tr>
<td>void setParent(XMLEventReader reader)</td>
<td>Set the parent of this instance.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
public EventReaderDelegate()

EventReaderDelegate

public EventReaderDelegate()

Construct an empty filter with no parent.

public EventReaderDelegate(XMLEventReader reader)

reader

EventReaderDelegate

public EventReaderDelegate(XMLEventReader reader)

Construct an filter with the specified parent.

Parameters:
reader - the parent

public void setParent(XMLEventReader reader)

reader
**setParent**

```java
public void setParent(XMLEventReader reader)
```

Set the parent of this instance.

**Parameters:**
reader - the new parent

**getParent**

```java
public XMLEventReader getParent()
```

```java
return null
```

**getParent**

```java
public XMLEventReader getParent()
```

Get the parent of this instance.

**Returns:**
the parent or null if none is set

**nextEvent**

```java
public XMLEvent nextEvent() throws XMLStreamException
```

**Description copied from interface: XMLEventReader**
Get the next XMLEvent

**Specified by:**
nextEvent in interface XMLEventReader

Throws:
XMLStreamException - if there is an error with the underlying XML.

See Also:
XMLEvent

---

public Object next()

next

public Object next()

Specified by:
next in interface Iterator

---

public boolean hasNext()

hasNext

public boolean hasNext()

Description copied from interface: XMLEventReader
Check if there are more events. Returns true if there are more events and false otherwise.

Specified by:
hasNext in interface Iterator
Specified by:
hasNext in interface XMLEventReader

Returns:
true if the event reader has more events, false otherwise
public XMLEvent peek() throws XMLStreamException

peek

public XMLEvent peek()
    throws XMLStreamException

    Description copied from interface: XMLEventReader
    Check the next XMLEvent without reading it from the stream.
    Returns null if the stream is at EOF or has no more XMLEvents. A
    call to peek() will be equal to the next return of next().

    Specified by:
        peek in interface XMLEventReader
    Throws:
        XMLStreamException
    See Also:
        XMLEvent

public void close() throws XMLStreamException

close

public void close()
    throws XMLStreamException

    Description copied from interface: XMLEventReader
    Frees any resources associated with this Reader. This method does
    not close the underlying input source.

    Specified by:
        close in interface XMLEventReader
    Throws:
        XMLStreamException - if there are errors freeing associated
        resources
public String getElementText() throws XMLStreamException

g getElementText

**public String getElementText() throws XMLStreamException**

Description copied from interface: [XMLEventReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEventReader.html)
Reads the content of a text-only element. Precondition: the current event is START_ELEMENT. Postcondition: The current event is the corresponding END_ELEMENT.

Specified by:
- `getElementText` in interface [XMLEventReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEventReader.html)

Throws:
- [XMLStreamException](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLStreamException.html) - if the current event is not a START_ELEMENT or if a non text element is encountered

public [XMLEvent](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEvent.html) nextTag() throws XMLStreamException

**nextTag**

**public [XMLEvent](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEvent.html) nextTag() throws XMLStreamException**

Description copied from interface: [XMLEventReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEventReader.html)
Skips any insignificant space events until a START_ELEMENT or END_ELEMENT is reached. If anything other than space characters are encountered, an exception is thrown. This method should be used when processing element-only content because the parser is not able to recognize ignorable whitespace if the DTD is missing or not interpreted.

Specified by:
- `nextTag` in interface [XMLEventReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.stream/XMLEventReader.html)
Throws:

**XMLStreamException** - if anything other than space characters are encountered

---

```java
public Object getProperty(String name) throws IllegalArgumentException
```

**getProperty**

```java
public Object getProperty(String name)
```

Throws **IllegalArgumentException** - if the property is not supported

---

```java
public void remove()
```

**remove**

```java
public void remove()
```

Specified by: **remove** in interface **Iterator**
public interface ExceptionListener

JMS Connection Connection ExceptionListener
onException JMSException

JMS

version 1.0 - 9 March 1998
See also setExceptionListener(ExceptionListener)

If a JMS provider detects a serious problem with a Connection object, it informs the Connection object's ExceptionListener, if one has been registered. It does this by calling the listener's onException method, passing it a JMSException argument describing the problem.

An exception listener allows a client to be notified of a problem asynchronously. Some connections only consume messages, so they would have no other way to learn that their connection has failed.

A JMS provider should attempt to resolve connection problems itself before it notifies the client of them.

Version:
1.0 - 9 March 1998

Author:
Mark Hapner, Rich Burridge

See Also:
Connection.setExceptionListener(ExceptionListener)
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void onException(JMSException exception)</td>
<td>Notifies user of a JMS exception.</td>
</tr>
</tbody>
</table>

Method Detail

```java
public void onException(JMSException exception)
```

JMS

`exception` - the JMS exception

Notifies user of a JMS exception.

Parameters:

- `exception` - the JMS exception

Overview Package Tree Deprecated Index Help

PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.interceptor  Annotation Type
ExcludeClassInterceptors

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface ExcludeClassInterceptors

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Interceptor

Used to exclude class-level interceptors for a business method.
javax.interceptor Annotation Type
ExcludeDefaultInterceptors

@Target(value={TYPE, METHOD})
@Retention(value=RUNTIME)
public @interface ExcludeDefaultInterceptors

Implements: Annotation
@Target(value={TYPE, METHOD})
@Retention(value=RUNTIME)

Bean Interceptor

Used to exclude default interceptors for a bean or a business method.
javax.persistence Annotation Type
ExcludeDefaultListeners

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface ExcludeDefaultListeners

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

since Java Persistence 1.0

Specifies that the invocation of default listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.

Since:
Java Persistence 1.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.persistence Annotation Type
ExcludeSuperclassListeners

```java
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface ExcludeSuperclassListeners
```

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Specifies that the invocation of superclass listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.

**Since:**
Java Persistence 1.0

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/technetwork/java/index.html).

PS:
javax.resource.spi.work  Class ExecutionContext

java.lang.Object  
   \javax.resource.spi.work.ExecutionContext

public class ExecutionContext
extends Object

Work  ExecutionContext

ExecutionContext

•
•  ExecutionContext  ExecutionContext

version  1.0

This class models an execution context (transaction, security, etc) with which the Work instance must be executed. This class is provided as a convenience for easily creating ExecutionContext instances by extending this class and overriding only those methods of interest.

Some reasons why it is better for ExecutionContext to be a class rather than an interface:

• There is no need for a resource adapter to implement this class. It only needs to implement the context information like transaction, etc.
• The resource adapter code does not have to change when the ExecutionContext class evolves. For example, more context types could be added to the ExecutionContext class (in the future) without forcing resource adapter implementations to change.

Version:  
1.0

Author:
Constructor Summary

**ExecutionContext()**

Method Summary

<table>
<thead>
<tr>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>getTransactionTimeout()</td>
</tr>
<tr>
<td></td>
<td>Get the transaction timeout value for a imported transaction.</td>
</tr>
<tr>
<td>Xid</td>
<td>getXid()</td>
</tr>
<tr>
<td>void</td>
<td>setTransactionTimeout(long timeout)</td>
</tr>
<tr>
<td></td>
<td>Set the transaction timeout value for a imported transaction.</td>
</tr>
<tr>
<td>void</td>
<td>setXid(Xid xid)</td>
</tr>
<tr>
<td></td>
<td>set a transaction context.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public ExecutionContext()

ExecutionContext

public ExecutionContext()
public void setXid(Xid xid)

    xid

setXid

public void setXid(Xid xid)

    set a transaction context.

    Parameters:
    xid - transaction context.

----------------------------------------

public Xid getXid()

getXid

public Xid getXid()  

----------------------------------------

public void setTransactionTimeout(long timeout) throws NotSupportedException

    timeout  0
    Throws   NotSupportedException:

setTransactionTimeout

public void setTransactionTimeout(long timeout)  
    throws NotSupportedException

    Set the transaction timeout value for a imported transaction.

    Parameters:
timeout - transaction timeout value in seconds. Only positive non-zero values are accepted. Other values are illegal and are rejected with a NotSupportedException.

Throws:

NotSupportedException - thrown to indicate an illegal timeout value.

public long getTransactionTimeout()

return -1 (getTransactionTimeout)

getTransactionTimeout

public long getTransactionTimeout()

Get the transaction timeout value for a imported transaction.

Returns:

the specified transaction timeout value in seconds. When no timeout value or an illegal timeout value had been specified, a value of -1 (WorkManager.UNKNOWN) is returned; such a transaction is excluded from regular timeout processing.
javax.el Class Expression

java.lang.Object
   \ javax.el.Expression

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   MethodExpression, ValueExpression

public abstract class Expression
   extends Object
   implements Serializable

Implements: java.io.Serializable
Extended by: MethodExpression, ValueExpression

   ValueExpression    MethodExpression

   equals()    hashCode()

   Serializable

Expression   String/   FunctionMapper

   since   JSP 2.1

Base class for the expression subclasses ValueExpression and MethodExpression, implementing characteristics common to both.

All expressions must implement the equals() and hashCode() methods so that two expressions can be compared for equality. They are redefined abstract in this class to force their implementation in subclasses.

All expressions must also be Serializable so that they can be saved and
Expressions are also designed to be immutable so that only one instance needs to be created for any given expression String / FunctionMapper. This allows a container to pre-create expressions and not have to re-parse them each time they are evaluated.

**Since:**
JSP 2.1

**See Also:**
Serialized Form

---

### Constructor Summary

**Expression()**

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(Object obj)</code></td>
<td>Determines whether the specified object is equal to this Expression.</td>
</tr>
<tr>
<td><code>getExpressionString()</code></td>
<td>Returns the original String used to create this Expression, unmodified.</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td>Returns the hash code for this Expression.</td>
</tr>
<tr>
<td><code>isLiteralText()</code></td>
<td>Returns whether this expression was created from only literal text.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

*clone, finalize, getClass, notify, notifyAll, toString, wait, wait*
### Constructor Detail

**public Expression()**

### Method Detail

**abstract public String getExpressionString()**

```java
Expression String

getExpressionString

public abstract String getExpressionString()

Returns the original String used to create this Expression, unmodified.

This is used for debugging purposes but also for the purposes of comparison (e.g. to ensure the expression in a configuration file has not changed).

This method does not provide sufficient information to re-create an expression. Two different expressions can have exactly the same expression string but different function mappings. Serialization should be used to save and restore the state of an Expression.
Returns:
The original expression String.

abstract public boolean equals(Object obj)

Determines whether the specified object is equal to this Expression.

The result is true if and only if the argument is not null, is an Expression object that is of the same type (ValueExpression or MethodExpression), and has an identical parsed representation.

Note that two expressions can be equal if their expression Strings are different. For example, ${fn1:foo()} and ${fn2:foo()} are equal if their corresponding FunctionMappers mapped fn1:foo and fn2:foo to the same method.

Overrides:
equals in class Object

Parameters:
obj - the Object to test for equality.

Returns:
true if obj equals this Expression; false otherwise.

See Also:
Hashtable, Object.equals(java.lang.Object)

---

abstract public int hashCode()

Expression

String #equals equals(Object)
hashCode hashCode

return Expression

See also equals, java.util.Hashtable, hashCode()

---

hashCode

public abstract int hashCode()

Returns the hash code for this Expression.

See the note in the equals(java.lang.Object) method on how two expressions can be equal if their expression Strings are different. Recall that if two objects are equal according to the equals(Object) method, then calling the hashCode method on each of the two objects must produce the same integer result. Implementations must take special note and implement hashCode correctly.

Overrides: 
hashCode in class Object

Returns:
The hash code for this Expression.

See Also:
equals(java.lang.Object), Hashtable, Object.hashCode()
isLiteralText

public abstract boolean isLiteralText()

Returns whether this expression was created from only literal text.

This method must return true if and only if the expression string this expression was created from contained no unescaped EL delimiters (${...} or #{...}).

Returns:
true if this expression was created from only literal text; false otherwise.
Class Expression

`javax.servlet.jsp.el` Expression

`java.lang.Object`<br>`javax.servlet.jsp.el.Expression`

**Deprecated.** As of JSP 2.1, replaced by `ValueExpression`

Expression ExpressionEvaluator

Expression evaluate() ELParseException

```
public abstract class Expression
extends Object
```

The abstract class for a prepared expression.

An instance of an Expression can be obtained via from an ExpressionEvaluator instance.

An Expression may or not have done a syntactic parse of the expression. A client invoking the evaluate() method should be ready for the case where ELParseException exceptions are raised.

**Since:**

JSP 2.0

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Expression()</code></td>
<td>Deprecated.</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract public</td>
<td>Object evaluate(VariableResolver vResolver)</td>
</tr>
<tr>
<td></td>
<td>Deprecated. Evaluates an expression that was previously prepared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

close, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public Expression()

Expression

public Expression()

Deprecated.

Method Detail

abstract public Object evaluate(VariableResolver vResolver) throws ELException

ELParseException   ELException

vResolver VariableResolver Objects

return

Throws ELException:

evaluate
public abstract Object evaluate(VariableResolver vResolver) throws ELException

 Deprecated. Evaluates an expression that was previously prepared. In some implementations preparing an expression involves full syntactic validation, but others may not do so. Evaluating the expression may raise an ELParseException as well as other ELExceptions due to run-time evaluation.

 Parameters:
 vResolver - A VariableResolver instance that can be used at runtime to resolve the name of implicit objects into Objects.

 Returns:
 The result of the expression evaluation.

 Throws:
 ELException - Thrown if the expression evaluation failed.
Class ExpressionEvaluator

ExpressionEvaluator JspContext / PageContext

parseExpression() evaluate() ExpressionEvaluator

transient ExpressionEvaluator

'${' '}' EL EL "abc${1+1}def${1+1}ghi"

"${1+1}${1+1}" ELEException

EL

- ${person.lastName}
- ${8 * 8}
- ${my:reverse('hello')}

since 2.0

public abstract class ExpressionEvaluator

extends Object

The abstract base class for an expression-language evaluator. Classes that implement an expression language expose their functionality via this abstract class.

An instance of the ExpressionEvaluator can be obtained via the JspContext / PageContext
The `parseExpression()` and `evaluate()` methods must be thread-safe. That is, multiple threads may call these methods on the same `ExpressionEvaluator` object simultaneously. Implementations should synchronize access if they depend on transient state. Implementations should not, however, assume that only one object of each `ExpressionEvaluator` type will be instantiated; global caching should therefore be static.

Only a single EL expression, starting with '${' and ending with '}', can be parsed or evaluated at a time. EL expressions cannot be mixed with static text. For example, attempting to parse or evaluate "abc${1+1}def${1+1}ghi" or even "${1+1}${1+1}" will cause an `ELException` to be thrown.

The following are examples of syntactically legal EL expressions:

- `${person.lastName}
- `${8 * 8}
- `${my:reverse('hello')}

**Since:**

JSP 2.0

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExpressionEvaluator()</code></td>
<td><code>Deprecated.</code></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>evaluate</code></td>
<td><code>String expression, Class expectedType, VariableResolver vResolver, FunctionMapper fMapper</code> <code>Deprecated. Evaluates an expression.</code></td>
</tr>
<tr>
<td><code>parseExpression</code></td>
<td><code>String expression, Class expectedType, FunctionMapper fMapper</code> <code>Deprecated. Prepare an expression for later evaluation.</code></td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.\textbf{Object}
\texttt{clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait}

**Constructor Detail**

public \textbf{ExpressionEvaluator}()

\textbf{ExpressionEvaluator}

class \textbf{ExpressionEvaluator} (public)

public ExpressionEvaluator()

\textit{Deprecated.}

**Method Detail**

\textbf{abstract public \texttt{Expression}} parseExpression(String expression, \texttt{Class<T>} expectedType, \textbf{FunctionMapper} fMapper) throws \textbf{ELException}

\textbf{ELParseException}

\texttt{expression}

\texttt{expectedType}

\texttt{fMapper}

FunctionMapper null

ExpressionEvaluator FunctionMapper

ExpressionEvaluator.parseExpression()

Expression.evaluate()

return Expression

\textbf{Throws} \texttt{ELException}

parseExpression

\textbf{public abstract \texttt{Expression}} parseExpression(String expression,
Deprecated.
Prepare an expression for later evaluation. This method should perform syntactic validation of the expression; if in doing so it detects errors, it should raise an ELParseException.

Parameters:
- `expression`: The expression to be evaluated.
- `expectedType`: The expected type of the result of the evaluation.
- `fMapper`: A FunctionMapper to resolve functions found in the expression. It can be null, in which case no functions are supported for this invocation. The ExpressionEvaluator must not hold on to the FunctionMapper reference after returning from `parseExpression()`. The Expression object returned must invoke the same functions regardless of whether the mappings in the provided FunctionMapper instance change between calling `ExpressionEvaluator.parseExpression()` and `Expression.evaluate()`.

Returns:
The Expression object encapsulating the arguments.

Throws:
- `ELException` - Thrown if parsing errors were found.

```java
abstract public Object evaluate(String expression, Class<T> expectedType, VariableResolver vResolver, FunctionMapper fMapper) throws ELException
```
public abstract Object evaluate(String expression, Class expectedType, VariableResolver vResolver, FunctionMapper fMapper)
    throws ELException

Deprecated.
Evaluates an expression. This method may perform some syntactic validation and, if so, it should raise an ELParseException error if it encounters syntactic errors. EL evaluation errors should cause an ELException to be raised.

Parameters:
expression - The expression to be evaluated.
extendedType - The expected type of the result of the evaluation
vResolver - A VariableResolver instance that can be used at runtime to resolve the name of implicit objects into Objects.
fMapper - A FunctionMapper to resolve functions found in the expression. It can be null, in which case no functions are supported for this invocation.

Returns:
The result of the expression evaluation.

Throws:
ELException - Thrown if the expression evaluation failed.
javax.el Class ExpressionFactory

java.lang.Object
  ↓ javax.el.ExpressionFactory

public abstract class ExpressionFactory
extends Object

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExpressionFactory()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract Object</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>abstract MethodExpression</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>abstract ValueExpression</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>abstract ValueExpression</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### Constructor Detail

**public ExpressionFactory()**

**ExpressionFactory**

**public ExpressionFactory()**

### Method Detail

**public static ExpressionFactory newInstance()**

```
ExpressionFactory<ExpressionFactory>
```

- **Services API JAR**
  
  META-INF/services/javax.el.ExpressionFactory

- **JRE "lib/el.properties"**
  
  java.util.Properties.load(InputStream)

  "javax.el.ExpressionFactory"

- **javax.el.ExpressionFactory**
newInstance
public static ExpressionFactory newInstance()

Creates a new instance of a ExpressionFactory. This method uses the following ordered lookup procedure to determine the ExpressionFactory implementation class to load:

- Use the Services API (as detailed in the JAR specification). If a resource with the name of META-INF/services/javax.el.ExpressionFactory exists, then its first line, if present, is used as the UTF-8 encoded name of the implementation class.
- Use the properties file "lib/el.properties" in the JRE directory. If this file exists and it is readable by the java.util.Properties.load(InputStream) method, and it contains an entry whose key is "javax.el.ExpressionFactory", then the value of that entry is used as the name of the implementation class.
- Use the javax.el.ExpressionFactory system property. If a system property with this name is defined, then its value is used as the name of the implementation class.
- Use a platform default implementation.

public static ExpressionFactory newInstance(java.util.Properties properties)

ExpressionFactory newInstance() properties null java.util.Properties

"javax.el"
• `javax.el.cacheSize`

    `properties`    null

`newInstance`

```java
public static ExpressionFactory newInstance(Properties properties)
```

Create a new instance of a `ExpressionFactory`, with optional properties. This method uses the same lookup procedure as the one used in `newInstance()`.

If the argument `properties` is not null, and if the implementation contains a constructor with a single parameter of type `java.util.Properties`, then the constructor is used to create the instance.

Properties are optional and can be ignored by an implementation.

The name of a property should start with "javax.el."

The following are some suggested names for properties.

• `javax.el.cacheSize`

**Parameters:**

- `properties` - Properties passed to the implementation. If null, then no properties.

`createValueExpression`

```java
public abstract ValueExpression createValueExpression(ELContext context, String expression, Class<?> expectedType)
```

Parses an expression into a `ValueExpression` for later evaluation.
Use this method for expressions that refer to values.

This method should perform syntactic validation of the expression. If in doing so it detects errors, it should raise an `ELException`.

**Parameters:**
- `context` - The EL context used to parse the expression. The `FunctionMapper` and `VariableMapper` stored in the ELContext are used to resolve functions and variables found in the expression. They can be `null`, in which case functions or variables are not supported for this expression. The object returned must invoke the same functions and access the same variable mappings regardless of whether the mappings in the provided `FunctionMapper` and `VariableMapper` instances change between calling `ExpressionFactory.createValueExpression()` and any method on `ValueExpression`.

Note that within the EL, the `${}` and `#{}` syntaxes are treated identically. This includes the use of `VariableMapper` and `FunctionMapper` at expression creation time. Each is invoked if not null, independent of whether the `#{}` or `${}` syntax is used for the expression.

- `expression` - The expression to parse
- `expectedType` - The type the result of the expression will be coerced to after evaluation.

**Returns:**
- The parsed expression

**Throws:**
- `NullPointerException` - Thrown if `expectedType` is null.
- `ELException` - Thrown if there are syntactical errors in the provided expression.

---

```java
createValueExpression

public abstract ValueExpression createValueExpression(Object instance, Class<?> expectedType)
```
Creates a ValueExpression that wraps an object instance. This method can be used to pass any object as a ValueExpression. The wrapper ValueExpression is read only, and returns the wrapped object via its getValue() method, optionally coerced.

**Parameters:**
- `instance` - The object instance to be wrapped.
- `expectedType` - The type the result of the expression will be coerced to after evaluation. There will be no coercion if it is `Object.class`.

**Throws:**
- `NullPointerException` - Thrown if `expectedType` is null.

---

createMethodExpression

```java
public abstract MethodExpression createMethodExpression(ELContext context, String expression, Class<? extends MethodExpression> expectedReturnType, Class<?> expectedParamTypes)
```

 Parses an expression into a `MethodExpression` for later evaluation. Use this method for expressions that refer to methods.

If the expression is a String literal, a `MethodExpression` is created, which when invoked, returns the String literal, coerced to `expectedReturnType`. An ELException is thrown if `expectedReturnType` is void or if the coercion of the String literal to the `expectedReturnType` yields an error (see Section "1.16 Type Conversion").

This method should perform syntactic validation of the expression. If in doing so it detects errors, it should raise an ELException.

**Parameters:**
- `context` - The EL context used to parse the expression. The `FunctionMapper` and `VariableMapper` stored in the ELContext are used to resolve functions and variables found in the expression.
They can be `null`, in which case functions or variables are not supported for this expression. The object returned must invoke the same functions and access the same variable mappings regardless of whether the mappings in the provided FunctionMapper and VariableMapper instances change between calling `ExpressionFactory.createMethodExpression()` and any method on `MethodExpression`.

Note that within the EL, the `{}$` and `#{}` syntaxes are treated identically. This includes the use of VariableMapper and FunctionMapper at expression creation time. Each is invoked if not null, independent of whether the `#{}` or `{}` syntax is used for the expression.

- `expression` - The expression to parse
- `expectedReturnType` - The expected return type for the method to be found. After evaluating the expression, the `MethodExpression` must check that the return type of the actual method matches this type. Passing in a value of `null` indicates the caller does not care what the return type is, and the check is disabled.
- `expectedParamTypes` - The expected parameter types for the method to be found. Must be an array with no elements if there are no parameters expected. It is illegal to pass `null`.

**Returns:**
The parsed expression

**Throws:**

- `ELException` - Thrown if there are syntactical errors in the provided expression.
- `NullPointerException` - if `paramTypes` is `null`.

---

coerceToType

```java
public abstract Object coerceToType(Object obj,
                                     Class<?> targetType)
```

Coerces an object to a specific type according to the EL type
conversion rules.

An `EL Exception` is thrown if an error results from applying the conversion rules.

**Parameters:**
- `obj` - The object to coerce.
- `targetType` - The target type for the coercion.

**Throws:**
- `EL Exception` - thrown if an error results from applying the conversion rules.

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.xml.registry.infomodel Interface ExtensibleObject

All Known Subinterfaces:
- Association, AuditableEvent, Classification, ClassificationScheme, Concept, ExternalIdentifier, ExternalLink, ExtrinsicObject, Organization, PostalAddress, RegistryEntry, RegistryObject, RegistryPackage, Service, ServiceBinding, SpecificationLink, User

```
public interface ExtensibleObject

Implemented by: PostalAddress, RegistryObject

ExtensibleObject Slot Slot

See alsojavax.xml.registry.infomodel.Slot

An ExtensibleObject is one that allows itself to be extended by utilizing dynamically added Slots that add arbitrary attributes to the object on a per instance basis.

Author:
Farrukh S. Najmi

See Also:
Slot
```

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addSlot(Slot slot)</td>
<td>Adds a Slot to this object.</td>
</tr>
<tr>
<td>void addSlots(Collection slots)</td>
<td>Adds more Slots to this object.</td>
</tr>
<tr>
<td>Slot getSlot(String slotName)</td>
<td>Gets the slot specified by slotName.</td>
</tr>
<tr>
<td>Collection getSlots()</td>
<td>Returns the Slots associated with this object.</td>
</tr>
</tbody>
</table>
void `removeSlot(String slotName)`
Removes a Slot from this object.

void `removeSlots(Collection slotNames)`
Removes specified Slots from this object.

Method Detail

public void addSlot(Slot slot) throws `JAXRException`

Slot

0

`slot` Slot

Throws: `JAXRException`: JAXR

addSlot

void `addSlot(Slot slot)`
throws `JAXRException`

Adds a Slot to this object.

**Capability Level:** 0

**Parameters:**
- `slot` - the Slot object being added to this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

public void addSlots(java.util.Collection<E> slots) throws `JAXRException`

Slot
addSlots

```java
void addSlots(Collection slots)
throws JAXRException
```

Adds more Slots to this object.

**Capability Level:** 0

**Parameters:**
- `slots` - the Collection of Slot objects being added to this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

```java
public void removeSlot(String slotName) throws JAXRException
SlotSlot
```

---

```java
slotName Slot
Throws JAXRException: JAXR
```
removeSlot

```java
void removeSlot(String slotName)
    throws JAXRException
```

Removes a Slot from this object. The Slot is identified by its name.

**Capability Level: 0**

**Parameters:**
- `slotName` - the name for the Slot object being removed from this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

public void removeSlots(java.util.Collection<E> slotNames)
    throws JAXRException
SlotSlot

```java
0
```

<table>
<thead>
<tr>
<th>slotNames</th>
<th>Slot</th>
<th>Collection</th>
<th>String</th>
<th>Collection</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>supplierCardinality</td>
<td>JAXRException</td>
<td>JAXR</td>
<td>0..*</td>
<td></td>
</tr>
<tr>
<td>associates</td>
<td></td>
<td>&lt;={Slot}&gt;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>undirected</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>supplierRole</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>link</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>aggregationByValue</td>
<td></td>
<td></td>
<td>en</td>
<td></td>
</tr>
</tbody>
</table>

---

removeSlots

```java
void removeSlots(Collection slotNames)
    throws JAXRException
```

Removes specified Slots from this object. The Slots are identified by
its name.

**Capability Level: 0**

**Parameters:**
- `slotNames` - the Collection of names for Slot objects being removed from this object. Must be a Collection of Strings

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

```java
public Slot getSlot(String slotName) throws JAXRException
{
    slotName = slot
    0

    slotName Slot
    Throws JAXRException: JAXR
}
```

**getSlot**

```java
Slot getSlot(String slotName) throws JAXRException
{
    Gets the slot specified by slotName.

    **Capability Level: 0**

    **Parameters:**
    - `slotName` - the name of the desired Slot object

    **Throws:**
    - `JAXRException` - If the JAXR provider encounters an internal error
```
public java.util.Collection<E> getSlots() throws JAXRException

Slot

0

return Slot CollectionCollection null

Throws JAXRException: JAXR

supplierCardinality 0..*

associates <{Slot}>

undirected

supplierRole slots

link aggregationByValue

getSlots

Collection getSlots() throws JAXRException

Returns the Slots associated with this object.

Capability Level: 0

Returns: Collection of Slot instances. The Collection may be empty but not null.

Throws: JAXRException - If the JAXR provider encounters an internal error
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.context Class ExternalContext

java.lang.Object
  ↓ javax.faces.context.ExternalContext

public abstract class ExternalContext
  extends Object

Faces API  JavaServer Faces  Servlet  Portlet

  Servlet     Portlet

This class allows the Faces API to be unaware of the nature of its containing application environment. In particular, this class allows JavaServer Faces based applications to run in either a Servlet or a Portlet environment.

In the method descriptions below, paragraphs starting with Servlet: and Portlet: denote behavior that is specific to that particular environment.

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>String</td>
<td>BASIC_AUTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String identifier for BASIC authentication.</td>
</tr>
<tr>
<td>static</td>
<td>String</td>
<td>CLIENT_CERT_AUTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String identifier for CLIENT_CERT authentication.</td>
</tr>
<tr>
<td>static</td>
<td>String</td>
<td>DIGEST_AUTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String identifier for DIGEST authentication.</td>
</tr>
<tr>
<td>static</td>
<td>String</td>
<td>FORM_AUTH</td>
</tr>
<tr>
<td></td>
<td></td>
<td>String identifier for FORM authentication.</td>
</tr>
</tbody>
</table>
## Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ExternalContext()</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dispatch(String path)</code></td>
<td><code>abstract void</code> dispatch(String path)`</td>
<td>Dispatch a request to the specified resource to create output for this response.</td>
</tr>
<tr>
<td><code>encodeActionURL(String url)</code></td>
<td><code>abstract String</code> encodeActionURL(String url)`</td>
<td>Return the input URL, after performing any rewriting needed to ensure that it will correctly identify an addressable action in the current application.</td>
</tr>
<tr>
<td><code>encodeNamespace(String name)</code></td>
<td><code>abstract String</code> encodeNamespace(String name)`</td>
<td>Return the specified name, after prefixing it with a namespace that ensures that it will be unique within the context of a particular page.</td>
</tr>
<tr>
<td><code>encodeResourceURL(String url)</code></td>
<td><code>abstract String</code> encodeResourceURL(String url)`</td>
<td>Return the input URL, after performing any rewriting needed to ensure that it will correctly identify an addressable resource in the current application.</td>
</tr>
<tr>
<td><code>getApplicationMap()</code></td>
<td><code>abstract Map&lt;String, Object&gt;</code> getApplicationMap()`</td>
<td>Return a mutable <code>Map</code> representing the application scope attributes for the current application.</td>
</tr>
<tr>
<td><code>getAuthType()</code></td>
<td><code>abstract String</code> getAuthType()`</td>
<td>Return the name of the authentication scheme used to authenticate the current user, if any; otherwise, return <code>null</code>.</td>
</tr>
<tr>
<td><code>getContext()</code></td>
<td><code>abstract Object</code> getContext()`</td>
<td>Return the application environment object instance for the current application.</td>
</tr>
<tr>
<td><code>getInitParameter(String name)</code></td>
<td><code>abstract String</code> getInitParameter(String name)`</td>
<td>Return the value of the specified application initialization parameter (if any).</td>
</tr>
</tbody>
</table>
abstract Map getInitParameterMap()  
Return an immutable Map whose keys are the set of application initialization parameter names configured for this application, and whose values are the corresponding parameter values.

abstract String getRemoteUser()  
Return the login name of the user making the current request if any; otherwise, return null.

abstract Object getRequest()  
Return the environment-specific object instance for the current request.

String getRequestMethod()  
Return the character encoding currently being used to interpret this request.

String getContentType()  
Return the MIME Content-Type for this request.

abstract String getServletPath()  
Return the portion of the request URI that identifies the web application context for this request.

abstract Map<String, Object> getCookieMap()  
Return an immutable Map whose keys are the set of cookie names included in the current request, and whose values (of type javax.servlet.http.Cookie) are the first (or only) cookie for each cookie name returned by the underlying request.

abstract Map<String, String> getHeaderMap()  
Return an immutable Map whose keys are the set of request header names included in the current request, and whose values (of type String) are the first (or only) value for each header name returned by the underlying request.

abstract Map<String, String[]> getHeaderValuesMap()  
Return an immutable Map whose keys are the set of request header names included in the current request, and whose values (of type String[]) are all of the value for each header name returned by the
underlying request.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getRequestLocale()</code></td>
<td>Return the preferred <code>Locale</code> in which the client will accept content.</td>
</tr>
<tr>
<td><code>getLocale()</code></td>
<td></td>
</tr>
<tr>
<td><code>getRequestLocales()</code></td>
<td>Return an <code>Iterator</code> over the preferred <code>Locales</code> specified in the request, in decreasing order of preference.</td>
</tr>
<tr>
<td><code>getRequestMap()</code></td>
<td>Return a mutable <code>Map</code> representing the request scope attributes for the current application.</td>
</tr>
<tr>
<td><code>getRequestParameterMap()</code></td>
<td>Return an immutable <code>Map</code> whose keys are the set of request parameters names included in the current request, and whose values (of type <code>String</code>) are the first (or only) value for each parameter name returned by the underlying request.</td>
</tr>
<tr>
<td><code>getRequestParameterNames()</code></td>
<td>Return an <code>Iterator</code> over the names of all request parameters included in the current request.</td>
</tr>
<tr>
<td><code>getRequestParameterValuesMap()</code></td>
<td>Return an immutable <code>Map</code> whose keys are the set of request parameters names included in the current request, and whose values (of type <code>String[]</code>) are all of the values for each parameter name returned by the underlying request.</td>
</tr>
<tr>
<td><code>getRequestPathInfo()</code></td>
<td>Return the extra path information (if any) included in the request URI; otherwise, return <code>null</code>.</td>
</tr>
<tr>
<td><code>getRequestServletPath()</code></td>
<td>Return the servlet path information (if any) included in the request URI; otherwise, return <code>null</code>.</td>
</tr>
<tr>
<td><code>getResource(String path)</code></td>
<td>Return a <code>URL</code> for the application resource mapped to the specified path, if it exists; otherwise, return <code>null</code>.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getResourcesAsStream(String path)</code></td>
<td>Return an InputStream for an application resource mapped to the specified path, if it exists; otherwise, return null.</td>
</tr>
<tr>
<td><code>getResourcePaths(String path)</code></td>
<td>Return the set of resource paths for all application resources whose resource path starts with the specified argument.</td>
</tr>
<tr>
<td><code>getResponse()</code></td>
<td>Return the environment-specific object instance for the current response.</td>
</tr>
<tr>
<td><code>getResponseCharacterEncoding()</code></td>
<td>Returns the name of the character encoding (MIME charset) used for the body sent in this response.</td>
</tr>
<tr>
<td><code>getResponseContentType()</code></td>
<td>Return the MIME Content-Type for this response.</td>
</tr>
<tr>
<td><code>getSession(boolean create)</code></td>
<td>If the <code>create</code> parameter is true, create (if necessary) and return a session instance associated with the current request.</td>
</tr>
<tr>
<td><code>getSessionMap()</code></td>
<td>Return a mutable Map representing the session scope attributes for the current application.</td>
</tr>
<tr>
<td><code>getUserPrincipal()</code></td>
<td>Return the Principal object containing the name of the current authenticated user, if any; otherwise, return null.</td>
</tr>
<tr>
<td><code>isUserInRole(String role)</code></td>
<td>Return true if the currently authenticated user is included in the specified role.</td>
</tr>
<tr>
<td><code>log(String message)</code></td>
<td>Log the specified message to the application object.</td>
</tr>
</tbody>
</table>
abstract void log(String message, Throwable exception)
    Log the specified message and exception to the application object.

abstract void redirect(String url)
    Redirect a request to the specified URL, and cause the responseComplete() method to be called on the FacesContext instance for the current request.

void setRequest(Object request)
    Set the environment-specific request to be returned by subsequent calls to getRequest().

void setRequestCharacterEncoding(String encoding)
    Overrides the name of the character encoding used in the body of this request.

void setResponse(Object response)
    Set the environment-specific response to be returned by subsequent calls to getResponse().

void setResponseCharacterEncoding(String encoding)
    Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

BASIC_AUTH

public static final String BASIC_AUTH

String identifier for BASIC authentication.
CLIENT_CERT_AUTH

public static final String CLIENT_CERT_AUTH

String identifier for CLIENT_CERT authentication.

See Also:
Constant Field Values

DIGEST_AUTH

public static final String DIGEST_AUTH

String identifier for DIGEST authentication.

See Also:
Constant Field Values

FORM_AUTH

public static final String FORM_AUTH

String identifier for FORM authentication.

See Also:
Constant Field Values
### Constructor Detail

**public** `ExternalContext()`

**ExternalContext**

**public** `ExternalContext()`

### Method Detail

**abstract public** `void dispatch(String path)` throws `java.io.IOException`

**Servlet** `javax.servlet.ServletContext`

- `getRequestDispatcher(path)`
- `forward()`

**Portlet** `javax.portlet.PortletContext`

- `getRequestDispatcher()`
- `include()`

**Throws**
- `FacesException`: 
- `ServletException`: 
- `PortletException`: 
- `IllegalArgumentException`: 
- `IllegalStateException`: 
- `portlet ActionRequest RenderRequest`: 
- `java.io.IOException`: `/`
- `NullPointerException`: `path null`

**dispatch**

**public abstract** `void dispatch(String path)`

**Throws** `IOException`
Dispatch a request to the specified resource to create output for this response.

**Servlet**: This must be accomplished by calling the `javax.servlet.ServletContext` method `getRequestDispatcher(path)`, and calling the `forward()` method on the resulting object.

**Portlet**: This must be accomplished by calling the `javax.portlet.PortletContext` method `getRequestDispatcher()`, and calling the `include()` method on the resulting object.

**Parameters:**
- `path` - Context relative path to the specified resource, which must start with a slash ("/") character

**Throws:**
- `FacesException` - thrown if a `ServletException` or `PortletException` occurs
- `IllegalArgumentException` - if no request dispatcher can be created for the specified path
- `IllegalStateException` - if this method is called in a portlet environment, and the current request is an `ActionRequest` instead of a `RenderRequest`
- `IOException` - if an input/output error occurs
- `NullPointerException` - if `path` is null

abstract public String encodeActionURL(String url)

**URL**

**Servlet**

`javax.servlet.http.HttpServletResponse` `encodeURL(url)`

**Portlet**

`javax.portlet.PortletResponse` `encodeURL(url)`

**url**

**URL**

**Throws**

`NullPointerException`: `url` null
**encodeActionURL**

```java
public abstract String encodeActionURL(String url)
```

Return the input URL, after performing any rewriting needed to ensure that it will correctly identify an addressable action in the current application.

*Servlet:* This must be the value returned by the `javax.servlet.http.HttpServletResponse` method `encodeURL(url)`.

*Portlet:* This must be the value returned by the `javax.portlet.PortletResponse` method `encodeURL(url)`.

**Parameters:**

- `url` - The input URL to be encoded

**Throws:**

- `NullPointerException` - if `url` is null

---

**abstract public String encodeNamespace(String name)**

*Servlet*

*Portlet*  

```java
javax.portlet.RenderResponse getNamespace()
```

**name**

**Throws**

- `IllegalArgumentException` - `portlet` is null
- `IOException` - `RenderResponse` cannot be used for this type of request

**encodeNamespace**
public abstract String encodeNamespace(String name)

Return the specified name, after prefixing it with a namespace that ensures that it will be unique within the context of a particular page.

Servlet: The input value must be returned unchanged.

Portlet: The returned value must be the input value prefixed by the value returned by the javax.portlet.RenderResponse method getNamespace().

Parameters:
**name** - Name to be encoded

Throws:
- IllegalStateException - if this method is called in a portlet environment, and the current response is an ActionResponse instead of a RenderResponse
- NullPointerException - if name is null

abstract public String encodeResourceURL(String url)

URL

Servlet javax.servlet.http.HttpServletResponse encodeURL(url)

Portlet javax.portlet.PortletResponse encodeURL(url)

url URL

Throws

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>url</td>
<td>NullPointerException</td>
</tr>
</tbody>
</table>

encodeResourceURL

public abstract String encodeResourceURL(String url)
Return the input URL, after performing any rewriting needed to ensure that it will correctly identify an addressable resource in the current application.

*Servlet:* This must be the value returned by the `javax.servlet.http.HttpServletResponse` method `encodeURL(url)`.

*Portlet:* This must be the value returned by the `javax.portlet.PortletResponse` method `encodeURL(url)`.

**Parameters:**
- `url` - The input URL to be encoded

**Throws:**
- `NullPointerException` - if `url` is null

```java
abstract public java.util.Map<K, V> getApplicationMap()
```

```java
Map clear() remove() put() putAll() get()
```

```java
Map Bean Bean
javax.annotation.PreDestroy void Bean
PreDestroy
```

```java
Servlet javax.servlet.ServletContext
getAttribute() getAttributeNames() removeAttribute() setAttribute()
```

```java
Portlet javax.portlet.PortletContext
getAttribute() getAttributeNames() removeAttribute() setAttribute()
```

```java
getApplicationMap
```
public abstract Map<String, Object> getApplicationMap()

Return a mutable Map representing the application scope attributes for the current application. The returned Map must implement the entire contract for a modifiable map as described in the JavaDocs for java.util.Map. Modifications made in the Map must cause the corresponding changes in the set of application scope attributes. Particularly the clear(), remove(), put(), putAll(), and get() operations must take the appropriate action on the underlying data structure.

For any of the Map methods that cause an element to be removed from the underlying data structure, the following action regarding managed-beans must be taken. If the element to be removed is a managed-bean, and it has one or more public no-argument void return methods annotated with javax.annotation.PreDestroy, each such method must be called before the element is removed from the underlying data structure. Elements that are not managed-beans, but do happen to have methods with that annotation must not have those methods called on removal. Any exception thrown by the PreDestroy annotated methods must be caught and not rethrown. The exception may be logged.

Servlet: This must be the set of attributes available via the javax.servlet.ServletContext methods getAttribute(), getAttributeNames(), removeAttribute(), and setAttribute().

Portlet: This must be the set of attributes available via the javax.portlet.PortletContext methods getAttribute(), getAttributeNames(), removeAttribute(), and setAttribute().

abstract public String getAuthType()

null BASIC_AUTHCLIENT_CERT_AUTHDIGEST_AUTH FORM_AUTH

Servlet javax.servlet.http.HttpServletRequest getAuthType()
Portlet  javax.portlet.http.PortletRequest
getAuthType()  

getAuthType

public abstract String getAuthType()

Return the name of the authentication scheme used to authenticate the current user, if any; otherwise, return null. For standard authentication schemes, the returned value will match one of the following constants: BASIC_AUTH, CLIENT_CERT_AUTH, DIGEST_AUTH, or FORM_AUTH.

Servlet: This must be the value returned by the javax.servlet.http.HttpServletRequest method getAuthType().

Portlet: This must be the value returned by the javax.portlet.http.PortletRequest method getAuthType().

abstract public Object getContext()

Servlet  javax.servlet.ServletContext
Portlet  javax.portlet.PortletContext

getContext

public abstract Object getContext()

Return the application environment object instance for the current application.
abstract public String getInitParameter(String name)

**Servlet**   javax.servlet.ServletContext
getInitParameter(name)

**Portlet**  javax.portlet.PortletContext
getInitParameter(name)

  *name*
  *Throws*       NullPointerException: name null

getInitParameter

public abstract String getInitParameter(String name)

Return the value of the specified application initialization parameter (if any).

**Servlet**: This must be the result of the javax.servlet.ServletContext method getInitParameter(name).

**Portlet**: This must be the result of the javax.portlet.PortletContext method getInitParameter(name).

**Parameters:**
  *name* - Name of the requested initialization parameter

**Throws:**
  *NullPointeException* - if name is null
abstract public java.util.Map<K, V> getInitParameterMap()

Map	Map	java.util.Map JavaDoc

Servlet	javax.servlet.ServletContext
getInitParameterNames/

Portlet	javax.portlet.PortletContext
getInitParameterNames/

getInitParameterMap

classmap getInitParameterMap()

public abstract Map getInitParameterMap()

Return an immutable Map whose keys are the set of application initialization parameter names configured for this application, and whose values are the corresponding parameter values. The returned Map must implement the entire contract for an unmodifiable map as described in the JavaDocs for java.util.Map.

Servlet: This result must be as if it were synthesized by calling the javax.servlet.ServletContext method getInitParameterNames, and putting each configured parameter name/value pair into the result.

Portlet: This result must be as if it were synthesized by calling the javax.portlet.PortletContext method getInitParameterNames, and putting each configured parameter name/value pair into the result.

abstract public String getRemoteUser()

null

Servlet	javax.servlet.http.HttpServletRequest
getRemoteUser()

**Portlet**  javax.portlet.http.PortletRequest
getRemoteUser()

getRemoteUser

```java
public abstract String getRemoteUser()
```

Return the login name of the user making the current request if any; otherwise, return `null`.

**Servlet:** This must be the value returned by the `javax.servlet.http.HttpServletRequest` method `getRemoteUser()`.

**Portlet:** This must be the value returned by the `javax.portlet.http.PortletRequest` method `getRemoteUser()`.

---

abstract public Object getRequest()

**Servlet**  javax.servlet.http.HttpServletRequest

**Portlet**  javax.portlet.PortletRequest  ActionRequest
RenderRequest

getRequest

```java
public abstract Object getRequest()
```

Return the environment-specific object instance for the current request.
Servlet: This must be the current request's javax.servlet.http.HttpServletRequest instance.

Portlet: This must be the current request's javax.portlet.PortletRequest instance, which will be either an ActionRequest or a RenderRequest depending upon when this method is called.

```
public void setRequest(Object request)

(requestCode)

UnsupportedOperationException

since 1.2
```

setRequest

-public void setRequest(Object request)

Set the environment-specific request to be returned by subsequent calls to getRequest(). This may be used to install a wrapper for the request.

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend this class.

Since:
1.2

```
public void setRequestCharacterEncoding(String encoding) throws java.io.UnsupportedEncodingException
```
Reader   Stream   IllegalStateException

**Servlet**   
javax.servlet.ServletRequest
setCharacterEncoding()

**Portlet**   
javax.portlet.ActionRequest
setCharacterEncoding()

UnsupportedOperationException

**setRequestCharacterEncoding**

```java
public void setRequestCharacterEncoding(String encoding)
    throws UnsupportedEncodingException
```

Overrides the name of the character encoding used in the body of this request.

Calling this method after the request has been accessed will have no effect, unless a Reader or Stream has been obtained from the request, in which case an IllegalStateException is thrown.

**Servlet:** This must call through to the javax.servlet.ServletRequest method setCharacterEncoding().

**Portlet:** This must call through to the javax.portlet.ActionRequest method setCharacterEncoding().

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend this class.

**Throws:**

UnsupportedEncodingException - if this is not a valid encoding
abstract public String getRequestContextPath()

Web URI

Servlet javax.servlet.http.HttpServletRequest
getContextPath()

Portlet javax.portlet.PortletRequest getPortletContextPath()

getRequestContextPath

public abstract String getRequestContextPath()

Return the portion of the request URI that identifies the web application context for this request.

Servlet: This must be the value returned by the javax.servlet.http.HttpServletRequest methodgetContextPath().

Portlet: This must be the value returned by the javax.portlet.PortletRequest methodgetPortletContextPath().

abstract public java.util.Map<K, V> getRequestCookieMap()

Map cookie javax.servlet.http.Cookie
cookie cookie java.util.Map JavaDoc

Servlet javax.servlet.http.HttpServletRequest
null 0

**Portlet** Map

### getRequestCookieMap

```java
public abstract Map&lt;String, Object&gt; getRequestCookieMap()
```

Return an immutable Map whose keys are the set of cookie names included in the current request, and whose values (of type javax.servlet.http.Cookie) are the first (or only) cookie for each cookie name returned by the underlying request. The returned Map must implement the entire contract for an unmodifiable map as described in the JavaDocs for java.util.Map.

*Servlet:* This must be the value returned by the javax.servlet.http.HttpServletRequest method get Cookies(), unless null was returned, in which case this must be a zero-length array.

*Portlet:* This must be an empty Map.

### getRequestHeaderMap

```java
abstract public java.util.Map&lt;K, V&gt; getRequestHeaderMap()
```

JavaDoc

*Servlet*  
javax.servlet.http.HttpServletRequest  
getHeader() getHeaderNames()

*Portlet*  
javax.portlet.PortletRequest  
getProperty() getPropertyName() HTTP portlet
getRequestHeaderMap

public abstract Map<String, String> getRequestHeaderMap()

Return an immutable Map whose keys are the set of request header names included in the current request, and whose values (of type String) are the first (or only) value for each header name returned by the underlying request. The returned Map must implement the entire contract for an unmodifiable map as described in the JavaDocs for java.util.Map. In addition, key comparisons must be performed in a case insensitive manner.

Servlet: This must be the set of headers available via the javax.servlet.http.HttpServletRequest methods getHeader() and getHeaderNames().

Portlet: This must be the set of properties available via the javax.portlet.PortletRequest methods getProperty() and getPropertyNames(). As such, HTTP headers will only be included if they were provided by the portlet container, and additional properties provided by the portlet container may also be included.

abstract public java.util.Map<K, V> getRequestHeaderValuesMap()

Map String[] Map java.util.Map

Servlet javax.servlet.http.HttpServletRequest
getHeaders() getHeaderNames()

Portlet javax.portlet.PortletRequest getProperties() getPropertyNames() HTTP portlet

getRequestHeaderValuesMap

public abstract Map<String, String[]> getRequestHeaderValuesMap()

Return an immutable Map whose keys are the set of request header names included in the current request, and whose values (of type String[]) are all of the value for each header name returned by the underlying request. The returned Map must implement the entire contract for an unmodifiable map as described in the JavaDocs for java.util.Map. In addition, key comparisons must be performed in a case insensitive manner.

Servlet: This must be the set of headers available via the javax.servlet.http.HttpServletRequest methods getHeaders() and getHeaderNames().

Portlet: This must be the set of properties available via the javax.portlet.PortletRequest methods getProperties() and getPropertyNames(). As such, HTTP headers will only be included if they were provided by the portlet container, and additional properties provided by the portlet container may also be included.

abstract public java.util.Locale getRequestLocale()

Locale

Servlet      javax.servlet.ServletRequest    getLocale()

Portlet      javax.portlet.PortletRequest   getLocale()

ggetRequestLocale

public abstract Locale getRequestLocale()
Return the preferred `Locale` in which the client will accept content.

**Servlet:** This must be the value returned by the `javax.servlet.ServletRequest` method `getLocale()`.

**Portlet:** This must be the value returned by the `javax.portlet.PortletRequest` method `getLocale()`.

```java
abstract public java.util.Iterator<E> getRequestLocales()

Locale   Iterator

Servlet   javax.servlet.ServletRequest   getLocales()
Portlet   javax.portlet.PortletRequest   getLocales()
```

**getRequestLocales**

```java
public abstract Iterator<Locale> getRequestLocales()
```

Return an `Iterator` over the preferred `Locales` specified in the request, in decreasing order of preference.

**Servlet:** This must be an `Iterator` over the values returned by the `javax.servlet.ServletRequest` method `getLocales()`.

**Portlet:** This must be an `Iterator` over the values returned by the `javax.portlet.PortletRequest` method `getLocales()`.

```java
abstract public java.util.Map<K, V> getRequestMap()

Map   Map   java.util.Map   JavaDoc   Map
```
clear() remove() put() putAll() get()

Map Bean Bean
javax.annotation.PreDestroy void Bean
PreDestroy

Servlet javax.servlet.ServletRequest
getAttribute() getAttributeNames() removeAttribute()
setAttribute()

Portlet javax.portlet.PortletRequest
getAttribute() getAttributeNames() removeAttribute()
setAttribute()

ggetRequestMap

public abstract Map<String, Object> getRequestMap()

Return a mutable Map representing the request scope attributes for the current application. The returned Map must implement the entire contract for a modifiable map as described in the JavaDocs for java.util.Map. Modifications made in the Map must cause the corresponding changes in the set of request scope attributes. Particularly the clear(), remove(), put(), putAll(), and get() operations must take the appropriate action on the underlying data structure.

For any of the Map methods that cause an element to be removed from the underlying data structure, the following action regarding managed-beans must be taken. If the element to be removed is a managed-bean, and it has one or more public no-argument void return methods annotated with javax.annotation.PreDestroy, each such method must be called before the element is removed from the underlying data structure. Elements that are not managed-beans, but do happen to have methods with that annotation must not have
those methods called on removal. Any exception thrown by the
PreDestroy annotated methods must by caught and not rethrown.
The exception may be logged.

Servlet: This must be the set of attributes available via the
javax.servlet.ServletRequest methods getAttribute(),
ggetAttributeNames(), removeAttribute(), and setAttribute().

Portlet: This must be the set of attributes available via the
javax.portlet.PortletRequest methods getAttribute(),
ggetAttributeNames(), removeAttribute(), and setAttribute().

abstract public java.util.Map<K, V>
getRequestParameterMap()

Map String Map java.util.Map

Servlet javax.servlet.ServletRequest getParameter()
getParameterNames()

Portlet javax.portlet.PortletRequest getParameter()
getParameterNames()

getRequestParameterMap

public abstract Map<String, String> getRequestParameterMap()

Return an immutable Map whose keys are the set of request
parameters names included in the current request, and whose
values (of type String) are the first (or only) value for each parameter
name returned by the underlying request. The returned Map must
implement the entire contract for an unmodifiable map as described
in the JavaDocs for java.util.Map.
**Servlet:** This must be the set of parameters available via the `javax.servlet.ServletRequest` methods `getParameter()` and `getParameterNames()`.

**Portlet:** This must be the set of parameters available via the `javax.portlet.PortletRequest` methods `getParameter()` and `getParameterNames()`.

```java
abstract public java.util.Iterator<E>
getRequestParameterNames()
```

**Iterator**

**Servlet** `javax.servlet.ServletRequest` `getParameterNames()` `Iterator`

**Portlet** `javax.portlet.PortletRequest` `getParameterNames()` `Iterator`

**getRequestParameterNames**

```java
public abstract Iterator<String> getRequestParameterNames()
```

Return an `Iterator` over the names of all request parameters included in the current request.

**Servlet:** This must be an `Iterator` over the values returned by the `javax.servlet.ServletRequest` method `getParameterNames()`.

**Portlet:** This must be an `Iterator` over the values returned by the `javax.portlet.PortletRequest` method `getParameterNames()`.

```java
abstract public java.util.Map<K, V>
getRequestParameterValuesMap()
```
**Map String[]**  
**Map java.util.Map**

**JavaDoc**

```java
public abstract Map<String[], String[]> getRequestParameterValuesMap()
```

Return an immutable Map whose keys are the set of request parameters names included in the current request, and whose values (of type String[]) are all of the values for each parameter name returned by the underlying request. The returned Map must implement the entire contract for an unmodifiable map as described in the JavaDocs for java.util.Map.

**Servlet**: This must be the set of parameters available via the javax.servlet.ServletRequest methods `getParameterValues()` and `getParameterNames()`.

**Portlet**: This must be the set of parameters available via the javax.portlet.PortletRequest methods `getParameterValues()` and `getParameterNames()`.

---

**abstract public String getRequestPathInfo()**

```java
URI null
```

**Servlet** javax.servlet.http.HttpServletRequest

`getPathInfo()`
**Portlet**

null

---

**getRequestPathInfo**

```java
public abstract String getRequestPathInfo()
```

Return the extra path information (if any) included in the request URI; otherwise, return null.

*Servlet:* This must be the value returned by the `javax.servlet.http.HttpServletRequest` method `getPathInfo()`.

*Portlet:* This must be null.

---

**abstract public String getRequestServletPath()**

```java
URI servlet null
```

*Servlet*  
`javax.servlet.http.HttpServletRequest`  
`getServletPath()`

*Portlet*  
null

---

**getRequestServletPath**

```java
public abstract String getRequestServletPath()
```

Return the servlet path information (if any) included in the request URI; otherwise, return null.

*Servlet:* This must be the value returned by the `javax.servlet.http.HttpServletRequest` method `getServletPath()`.
Portlet: This must be **null**.

```java
public String getRequestCharacterEncoding()
```

**Servlet**  
*javax.servlet.ServletRequest*  
`getCharacterEncoding()`

**Portlet**  
*javax.portlet.ActionRequest*  
`getCharacterEncoding()`

`UnsupportedOperationException`

since 1.2

**getRequestCharacterEncoding**

```java
public String getRequestCharacterEncoding()
```

Return the character encoding currently being used to interpret this request.

**Servlet**: This must return the value returned by the *javax.servlet.ServletRequest* method `getCharacterEncoding()`.

**Portlet**: This must return the value returned by the *javax.portlet.ActionRequest* method `getCharacterEncoding()`.

The default implementation throws `UnsupportedOperationException` and is provided for the sole purpose of not breaking existing applications that extend this class.

**Since**:  
1.2
public String getRequestContentType()

MIME null

Servlet javax.servlet.ServletRequest getContentType()

Portlet null

UnsupportedOperationException

since 1.2

getRequestContentType

public String getRequestContentType()

Return the MIME Content-Type for this request. If not available, return null.

Servlet: This must return the value returned by the javax.servlet.ServletRequest method getContentType().

Portlet: This must return null.

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend this class.

Since:

1.2

public String getResponseCharacterEncoding()
**getResponseCharacterEncoding**

```java
public String getResponseCharacterEncoding()
```

Returns the name of the character encoding (MIME charset) used for the body sent in this response.

**Servlet:** This must return the value returned by the `javax.servlet.ServletResponse` method `getCharacterEncoding()`.

**Portlet:** This must return `null`.

The default implementation throws `UnsupportedOperationException` and is provided for the sole purpose of not breaking existing applications that extend this class.

**Since:**

1.2

---

**public String getResponseContentType()**

**MIME**

null

**Servlet**

`javax.servlet.ServletResponse` `getContentType()`

**Portlet**

null
UnsupportedOperationException

since 1.2

generateResponseContent
type

public String generateResponseContent()

Return the MIME Content-Type for this response. If not available, return null.

Servlet: This must return the value returned by the
javax.servlet.ServletResponse method getContentType().

Portlet: This must return null.

The default implementation throws UnsupportedOperationException
and is provided for the sole purpose of not breaking existing
applications that extend this class.

Since:
1.2

abstract public java.net.URL getResource(String path)
throws java.net.MalformedURLException

URL null

Servlet javax.servlet.ServletContext
getResources(path)

Portlet javax.portlet.PortletContext getResourc

path ("/")

Throws java.net.MalformedURLException:
public abstract URL getResource(String path) throws MalformedURLException

Return a URL for the application resource mapped to the specified path, if it exists; otherwise, return null.

Servlet: This must be the value returned by the javax.servlet.ServletContext method getResource(path).

Portlet: This must be the value returned by the javax.portlet.PortletContext method getResource(path).

Parameters:
- path - The path to the requested resource, which must start with a slash ("/" character)

Throws:
- MalformedURLException - if the specified path is not in the correct form
- NullPointerException - if path is null

abstract public java.io.InputStream getResourceAsStream(String path)

InputStream null

Servlet javax.servlet.ServletContext
getResourcesAsStream(path)

Portlet javax.portlet.PortletContext
getResourcesAsStream(path)

path ("/")
Throws NullPointerException: path null

getResourceAsStream

public abstract InputStream getResourceAsStream(String path)

Return an InputStream for an application resource mapped to the specified path, if it exists; otherwise, return null.

Servlet: This must be the value returned by the javax.servlet.ServletContext method getResourceAsStream(path).

Portlet: This must be the value returned by the javax.portlet.PortletContext method getResourceAsStream(path).

Parameters:
   path - The path to the requested resource, which must start with a slash ("/") character

Throws:
   NullPointerException - if path is null

abstract public java.util.Set<E> getResourcePaths(String path)

Set

Servlet javax.servlet.ServletContext getResourcePaths(path)

Portlet javax.portlet.PortletContext getResourcePaths(path)

path ("/")

Throws NullPointerException: path null
**getResourcePaths**

```java
public abstract Set<String> getResourcePaths(String path)
```

Return the set of resource paths for all application resources whose resource path starts with the specified argument.

*Servlet:* This must be the value returned by the `javax.servlet.ServletContext` method `getResourcePaths(path)`.

*Portlet:* This must be the value returned by the `javax.portlet.PortletContext` method `getResourcePaths(path)`.

**Parameters:**
- `path` - Partial path used to match resources, which must start with a slash (`"/"`) character

**Throws:**
- `NullPointerException` - if `path` is null

---

**abstract public Object getResponse()**

*Servlet*  
```java
javax.servlet.http.HttpServletResponse
```

*Portlet*  
```java
javax.portlet.PortletResponse
```

`ActionResponse`  
`RenderResponse`

---

**getResponse**

```java
public abstract Object getResponse()
```

Return the environment-specific object instance for the current response.

*Servlet:* This is the current request's

Portlet: This is the current request's javax.portlet.PortletResponse instance, which will be either an ActionResponse or a RenderResponse depending upon when this method is called.

public void setResponse(Object response)

getResponse

UnsupportedOperationException

since 1.2

setResponse

public void setResponse(Object response)

Set the environment-specific response to be returned by subsequent calls to getResponse(). This may be used to install a wrapper for the response.

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend this class.

Since:

1.2

MIME UTF-8

Servlet javax.servlet.ServletResponse
setCharacterEncoding()

**Portlet**

UnsupportedOperationException

since 1.2

setResponseCharacterEncoding

public void setResponseCharacterEncoding(String encoding)

Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8.

**Servlet:** This must call through to the `javax.servlet.ServletResponse` method `setCharacterEncoding()`.

**Portlet:** This method must take no action.

The default implementation throws `UnsupportedOperationException` and is provided for the sole purpose of not breaking existing applications that extend this class.

Since:

1.2

abstract public Object getSession(boolean create)

create true create false null

**Servlet** `javax.servlet.http.HttpServletRequest` `getSession(create)`

**Portlet** `javax.portlet.PortletRequest` `getPortletSession(create)`
getSession

public abstract Object getSession(boolean create)

    If the create parameter is true, create (if necessary) and return a session instance associated with the current request. If the create parameter is false return any existing session instance associated with the current request, or return null if there is no such session.

Servlet: This must return the result of calling getSession(create) on the underlying javax.servlet.http.HttpServletRequest instance.

Portlet: This must return the result of calling getPortletSession(create) on the underlying javax.portlet.PortletRequest instance.

Parameters:
create - Flag indicating whether or not a new session should be created if there is no session associated with the current request.

abstract public java.util.Map<K, V> getSessionMap()
**Portlet**  
`javax.portlet.PortletSession`  
`getAttribute()` `getAttributeNames()` `removeAttribute()`  
`setAttribute()`  
**PORTLET_SCOPE**

**getSessionMap**

```java
public abstract Map<String, Object> getSessionMap()
```

Return a mutable `Map` representing the session scope attributes for the current application. The returned `Map` must implement the entire contract for a modifiable map as described in the JavaDocs for `java.util.Map`. Modifications made in the `Map` must cause the corresponding changes in the set of session scope attributes. Particularly the `clear()`, `remove()`, `put()`, and `get()` operations must take the appropriate action on the underlying data structure. Accessing attributes via this `Map` must cause the creation of a session associated with the current request, if such a session does not already exist.

For any of the `Map` methods that cause an element to be removed from the underlying data structure, the following action regarding managed-beans must be taken. If the element to be removed is a managed-bean, and it has one or more public no-argument void return methods annotated with `javax.annotation.PreDestroy`, each such method must be called before the element is removed from the underlying data structure. Elements that are not managed-beans, but do happen to have methods with that annotation must not have those methods called on removal. Any exception thrown by the `PreDestroy` annotated methods must by caught and not rethrown. The exception may be logged.

**Servlet:** This must be the set of attributes available via the `javax.servlet.http.HttpServletSession` methods `getAttribute()`, `getAttributeNames()`, `removeAttribute()`, and `setAttribute()`.

**Portlet:** This must be the set of attributes available via the `javax.portlet.PortletSession` methods `getAttribute()`,
getAttributeNames(), removeAttribute(), and setAttribute(). All session attribute access must occur in PORTLET_SCOPE scope within the session.

abstract public java.security.Principal getUserPrincipal()

Principal null

Servlet javax.servlet.http.HttpServletRequest getUserPrincipal()

Portlet javax.portlet.http.PortletRequest getUserPrincipal()

g getUserPrincipal

public abstract Principal getUserPrincipal()

Return the Principal object containing the name of the current authenticated user, if any; otherwise, return null.

Servlet: This must be the value returned by the javax.servlet.http.HttpServletRequest method getUserPrincipal().

Portlet: This must be the value returned by the javax.portlet.http.PortletRequest method getUserPrincipal().

abstract public boolean isUserInRole(String role)

ture false

Servlet javax.servlet.http.HttpServletRequest
**isUserInRole**

```java
public abstract boolean isUserInRole(String role)
```

*Return true if the currently authenticated user is included in the specified role. Otherwise, return false.*

**Servlet:** This must be the value returned by the `javax.servlet.http.HttpServletRequest` method `isUserInRole(String role)`.

**Portlet:** This must be the value returned by the `javax.portlet.http.PortletRequest` method `isUserInRole(String role)`.

**Parameters:**
- `role` - Logical role name to be checked

**Throws:**
- `NullPointerException` - if `role` is `null`

---

**abstract public void log(String message)**

```java
Servlet javax.servlet.ServletContext log(String)
Portlet javax.portlet.PortletContext log(String)
```

*message*
Throws: NullPointerException: message null

log

```java
public abstract void log(String message)
```

Log the specified message to the application object.

**Servlet**: This must be performed by calling the `javax.servlet.ServletContext` method `log(String)`.

**Portlet**: This must be performed by calling the `javax.portlet.PortletContext` method `log(String)`.

**Parameters:**
- `message` - Message to be logged

**Throws:**
- `NullPointerException` - if message is null

abstract public void log(String message, Throwable exception)

**Servlet**
- `javax.servlet.ServletContext` log(String, Throwable)

**Portlet**
- `javax.portlet.PortletContext` log(String, Throwable)

- `message` exception
- **Throws** `NullPointerException`: message exception null
**Log**

```
public abstract void log(String message, Throwable exception)
```

Log the specified message and exception to the application object.

**Servlet:** This must be performed by calling the `javax.servlet.ServletContext` method `log(String, Throwable)`.

**Portlet:** This must be performed by calling the `javax.portlet.PortletContext` method `log(String, Throwable)`.

**Parameters:**
- message - Message to be logged
- exception - Exception to be logged

**Throws:**
- `NullPointerException` - if message or exception is null

---

**abstract public void redirect(String url) throws java.io.IOException**

**URL**

```
URL FacesContext responseComplete()
```

**Servlet**

```
javax.servlet.http.HttpServletResponse sendRedirect()
```

**Portlet**

```
javax.portlet.ActionResponse sendRedirect()
```

**Throws**
- `IllegalArgumentException`: url
- `IllegalStateException`: RenderResponse, ActionResponse
- `IllegalStateException`: portlet
- `IllegalStateException`: servlet
- `java.io.IOException`: /
public abstract void redirect(String url)
        throws IOException

Redirect a request to the specified URL, and cause the
responseComplete() method to be called on the FacesContext
instance for the current request.

Servlet: This must be accomplished by calling the
javax.servlet.http.HttpServletResponse method.sendRedirect().

Portlet: This must be accomplished by calling the
javax.portlet.ActionResponse method.sendRedirect().

Parameters:
url - Absolute URL to which the client should be redirected

Throws:
IllegalArgumentException - if the specified url is relative
IllegalStateException - if, in a portlet environment, the current
response object is a RenderResponse instead of an
ActionResponse
IllegalStateException - if, in a servlet environment, the current
response has already been committed
IOException - if an input/output error occurs
<table>
<thead>
<tr>
<th>Method</th>
<th>Field</th>
<th>Constructor</th>
<th>Nested</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method1</td>
<td>Field1</td>
<td>Constructor1</td>
<td>Nested1</td>
</tr>
<tr>
<td>Method2</td>
<td>Field2</td>
<td>Constructor2</td>
<td>Nested2</td>
</tr>
<tr>
<td>Method3</td>
<td>Field3</td>
<td>Constructor3</td>
<td>Nested3</td>
</tr>
<tr>
<td>Method4</td>
<td>Field4</td>
<td>Constructor4</td>
<td>Nested4</td>
</tr>
</tbody>
</table>
javax.xml.registry.infomodel  Interface ExternalIdentifier

All Superinterfaces:
   ExtensibleObject, RegistryObject

public interface ExternalIdentifier
   extends RegistryObject

Implements: RegistryObject

ExternalIdentifier  DUNS Social Security Number
RegistryObject  RegistryObject "DUNS" "Social Security Number"  DUNS  RegistryObject  0
ExternalIdentifier

See also  javax.xml.registry.infomodel.RegistryObject

ExternalIdentifier instances provide the additional identifier information to RegistryObjects such as DUNS number, Social Security Number, or an alias name of the organization. The attribute name inherited from RegistryObject is used to contain the identification scheme ("DUNS" "Social Security Number", etc.), and the attribute value contains the actual information (e.g. the actual DUNS number). Each RegistryObject may have 0 or more ExternalIdentifiers.

Author:
   Farrukh S. Najmi
See Also:
   RegistryObject

Method Summary

+ ClassificationScheme getIdentificationScheme()  
   Gets the ClassificationScheme that is used as the scheme for identifying this object.
**RegistryObject**

### Methods

- **getRegistryObject()**
  - Gets the parent RegistryObject for this ExternalId.

- **getValue()**
  - Gets the value of an ExternalIdentifier.

- **setIdentificationScheme(ClassificationScheme identificationScheme)**
  - Sets the ClassificationScheme that is used as the scheme for identifying this object.

- **setValue(String value)**
  - Sets the value of an ExternalIdentifier.

---

**Methods inherited from interface**

**javax.xml.registry.infomodel.RegistryObject**

- addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, removeClassifications, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML

---

**Methods inherited from interface**

**javax.xml.registry.infomodel.ExtensibleObject**

- addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

---

### Method Detail

**public RegistryObject getRegistryObject() throws JAXRException**

- `ExternalId` RegistryObject `registryObject`

**RegistryObject addExternalIdentifier**
getRegistryObject

```
public RegistryObject getRegistryObject() throws JAXRException
```

Gets the parent RegistryObject for this ExternalIdentifier. To set the registryObject call addExternalIdentifier on a RegistryObject.

**Capability Level: 0**

**Returns:**
- the RegistryObject that this object identifies

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

**See Also:**
- `RegistryObject.addExternalIdentifier(ExternalIdentifier ei)`

---

getValue

```
public String getValue() throws JAXRException
```

**getvalue**

```
String getValue()
```

**return**
- DUNS

**Throws**
- JAXRException: JAXR
throws `JAXRException`

Gets the value of an ExternalIdentifier.

**Capability Level: 0**

**Returns:**
the identification value defined by this object (e.g. a company's DUNS number)

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error

```java
public void setValue(String value) throws JAXRException

ExternalIdentifier

0

value    DUNS
Throws    JAXRException: JAXR
```

**setValue**

```java
void setValue(String value)
throws JAXRException
```

Sets the value of an ExternalIdentifier.

**Capability Level: 0**

**Parameters:**
value - the identification value defined by this object (e.g. a company's DUNS number)

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error
public ClassificationScheme getIdentificationScheme() throws JAXRException

ClassificationScheme 0

    return "DUNS" ClassificationScheme

Throws JAXRException: JAXR

supplierCardinality 0..*
clientCardinality 0..*
associationAsClass Classification

getIdentificationScheme

ClassificationScheme getIdentificationScheme() throws JAXRException

Gets the ClassificationScheme that is used as the identification scheme for identifying this object.

Capability Level: 0

Returns: the ClassificationScheme that is used as the identification scheme (e.g. "DUNS")

Throws: JAXRException - If the JAXR provider encounters an internal error

public void setIdentificationScheme(ClassificationScheme identificationScheme) throws JAXRException

ClassificationScheme 0
IdentificationScheme "DUNS" ClassificationScheme

Throws JAXRException: JAXR

setIdentificationScheme

void setIdentificationScheme(ClassificationScheme identificationScheme) throws JAXRException

Sets the ClassificationScheme that is used as the identification scheme for identifying this object.

Capability Level: 0

Parameters:
identificationScheme - the ClassificationScheme that is used as the identification scheme (e.g. "DUNS")

Throws:
JAXRException - If the JAXR provider encounters an internal error

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel Interface ExternalLink

All Superinterfaces:
   ExtensibleObject, RegistryObject, URIValidator

public interface ExternalLink
extends RegistryObject, URIValidator

Implements: RegistryObject, URIValidator

ExternalLink URIRegistryObject ExternalLink
RegistryObject

Submitting Organization DTD Submitting
Organization ExternalLink

See also javax.xml.registry.infomodel.RegistryObject

ExternalLink instances model a named URI to content that may reside
outside the registry. RegistryObject may be associated with any number
of ExternalLinks to annotate a RegistryObject with external links to
external content.

Consider the case where a Submitting Organization submits a repository
item (e.g. a DTD) and wants to associate some external content to that
object (e.g. the Submitting Organization's home page). The ExternalLink
enables this capability.

Author:
   Farrukh S. Najmi
See Also:
   RegistryObject
**getExternalURI()**

- Gets URI to the an external resource.

**getLinkedObjects()**

- Gets the collection of RegistryObjects that are annotated by this ExternalLink.

**setExternalURI(String uri)**

- Sets URI for an external resource.

---

### Methods inherited from interface javax.xml.registry.infomodel.RegistryObject

- addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML

---

### Methods inherited from interface javax.xml.registry.infomodel.ExtensibleObject

- addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

---

### Methods inherited from interface javax.xml.registry.infomodel.URIValidator

- getValidateURI, setValidateURI

---

### Method Detail

**public java.util.Collection<E> getLinkedObjects() throws JAXRException**

ExternalLink RegistryObject Collection
getLinkedObjects

`Collection getLinkedObjects()`

Throws `JAXRException`

Gets the collection of RegistryObjects that are annotated by this ExternalLink.

**Capability Level:** 0

**Returns:**
Collection of RegistryObjects. Return an empty Collection if no RegistryObjects are annotated by this object.

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error

---

public `String getExternalURI()`

Throws `JAXRException`

URI NULL

---

getExternalURI

`String getExternalURI()`

Throws `JAXRException`
Gets URI to the an external resource. Default is a NULL String.

**Capability Level: 0**

**Returns:**
the URI String for this object (e.g. "http://java.sun.com")

**Throws:**
  JAXRException - If the JAXR provider encounters an internal error

```java
public void setExternalURI(String uri) throws JAXRException
```

**setExternalURI**

Sets URI for an external resource.

**Capability Level: 0**

**Parameters:**
  uri - the URI String for this object (e.g. "http://java.sun.com")

**Throws:**
  JAXRException - If the JAXR provider encounters an internal error
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel Interface ExtrinsicObject

All Superinterfaces:
   ExtensibleObject, RegistryEntry, RegistryObject, Versionable

public interface ExtrinsicObject
extends RegistryEntry

Implements: RegistryEntry

ExtrinsicObject mime

ExtrinsicObject Collaboration Protocol Profiles (CPP)

ExtrinsicObjects provide metadata that describes submitted content whose type is not intrinsically known to the registry and therefore must be described by means of additional attributes (e.g., mime type).

Examples of content described by ExtrinsicObject include Collaboration Protocol Profiles (CPP), business process descriptions, and schemas.

Author:
   Farrukh S. Najmi

---

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.xml.registry.infomodel.RegistryEntry</th>
</tr>
</thead>
<tbody>
<tr>
<td>STABILITY_DYNAMIC, STABILITY_DYNAMIC_COMPATIBLE, STABILITY_STATIC,</td>
</tr>
<tr>
<td>STATUS_APPROVED, STATUS_DEPRECATED, STATUS_SUBMITTED,</td>
</tr>
<tr>
<td>STATUS_WITHDRAWN</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getMimeType()</code></td>
<td>Gets the mime type associated with this object.</td>
</tr>
<tr>
<td><code>DataHandler getRepositoryItem()</code></td>
<td>Gets the repository item for this object.</td>
</tr>
<tr>
<td><code>boolean isOpaque()</code></td>
<td>Determines whether the ExtrinsicObject is opaque (not readable) by the registry operator.</td>
</tr>
<tr>
<td><code>void setMimeType(String mimeType)</code></td>
<td>Sets the mime type associated with this object.</td>
</tr>
<tr>
<td><code>void setOpaque(boolean isOpaque)</code></td>
<td>Sets whether the ExtrinsicObject is opaque (not readable) by the registry.</td>
</tr>
<tr>
<td><code>void setRepositoryItem(DataHandler repositoryItem)</code></td>
<td>Sets the repository item for this object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.registry.infomodel.RegistryEntry`:
- `getExpiration`, `getStability`, `getStatus`, `setExpiration`, `setStability`

Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`:
- `addAssociation`, `addAssociations`, `addClassification`, `addClassifications`, `addExternalIdentifier`, `addExternalIdentifiers`, `addExternalLink`, `addExternalLinks`, `getAssociatedObjects`, `getAssociations`, `getAuditTrail`, `getClassifications`, `getDescription`, `getExternalIdentifiers`, `getExternalLinks`, `getKey`, `getLifeCycleManager`, `getName`, `getObjectType`, `getRegistryPackages`, `getSubmittingOrganization`, `removeAssociation`, `removeAssociations`, `removeClassification`, `removeClassifications`, `removeExternalIdentifier`, `removeExternalIdentifiers`, `removeExternalLink`, `removeExternalLinks`, `setAssociations`, `setClassifications`, `setDescription`, `setExternalIdentifiers`, `setExternalLinks`, `setKey`, `setName`, `toXML`

Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`
public String getMimeType() throws JAXRException

mime NULL

1

    return mime

Throws: JAXRException - JAXR

getMimeType

String getMimeType()

    throws JAXRException

Gets the mime type associated with this object. Default is a NULL String.

Capability Level: 1

Returns: the mime type associated with this object

Throws:
    JAXRException - If the JAXR provider encounters an internal error

public void setMimeType(String mimeType) throws
**JAXRException**

**mime**

1

```
   mimeType
   Returns
   JAXRException: JAXR
```

**setMimeType**

```java
void setMimeType(String mimeType)
throws JAXRException
```

Sets the mime type associated with this object.

**Capability Level:** 1

**Parameters:**
- `mimeType` - the mime type associated with this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**public boolean isOpaque() throws JAXRException**

**ExtrinsicObject**

**Submitting Organization**

1

```
   return ExtrinsicObject
   Returns JAXRException: JAXR
   true false
```

**isOpaque**
boolean `isOpaque()`
throws `JAXRException`

Determines whether the ExtrinsicObject is opaque (not readable) by the registry operator.

In some situations, a Submitting Organization may submit content that is encrypted and not even readable by the registry. This attribute allows the registry to know whether this is the case.

**Capability Level: 1**

**Returns:**
true if the ExtrinsicObject is readable by the registry operator;
false otherwise

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error

---

**public void setOpaque(boolean isOpaque) throws JAXRException**

**ExtrinsicObject**

1

`isOpaque`  ExtrinsicObject  true  boo

**Throws**
`JAXRException`: JAXR

---

**setOpaque**

**void setOpaque(boolean isOpaque)**
throws `JAXRException`

Sets whether the ExtrinsicObject is opaque (not readable) by the registry.

**Capability Level: 1**
Parameters:
- isOpaque - boolean value set to true if the ExtrinsicObject is readable by the registry operator; false otherwise

Throws:
- JAXRException - If the JAXR provider encounters an internal error

public DataHandler getRepositoryItem() throws JAXRException

null

1

return DataHandler

Throws: JAXRException: JAXR

getRepositoryItem

DataHandler getRepositoryItem()

throws JAXRException

Gets the repository item for this object. Must not return null.

Capability Level: 1

Returns:
- the DataHandler for the repository item

Throws:
- JAXRException - If the JAXR provider encounters an internal error

public void setRepositoryItem(DataHandler repositoryItem) throws JAXRException
void setRepositoryItem(DataHandler repositoryItem) throws JAXRException

Sets the repository item for this object.

**Capability Level: 1**

**Parameters:**
- repositoryItem - the DataHandler for the repository item. Must not be null

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error
javax.faces.context Class FacesContext

java.lang.Object
   - javax.faces.context.FacesContext

public abstract class FacesContext extends Object

FacesContext  JavaServer Faces

<table>
<thead>
<tr>
<th>Web</th>
<th>FacesContextFactory</th>
<th>getFacesContext()</th>
</tr>
</thead>
<tbody>
<tr>
<td>FacesContext</td>
<td>release()</td>
<td>FacesContext</td>
</tr>
<tr>
<td>servlet</td>
<td></td>
<td>Web</td>
</tr>
</tbody>
</table>

FacesContext contains all of the per-request state information related to the processing of a single JavaServer Faces request, and the rendering of the corresponding response. It is passed to, and potentially modified by, each phase of the request processing lifecycle.

A FacesContext instance is associated with a particular request at the beginning of request processing, by a call to the getFacesContext() method of the FacesContextFactory instance associated with the current web application. The instance remains active until its release() method is called, after which no further references to this instance are allowed. While a FacesContext instance is active, it must not be referenced from any thread other than the one upon which the servlet container executing this web application utilizes for the processing of this request.

Constructor Summary

<p>| FacesContext() |</p>
<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>abstract void</strong> addMessage(String clientId, FacesMessage message)</td>
</tr>
<tr>
<td>Append a FacesMessage to the set of messages associated with the specified client identifier, if clientId is not null.</td>
</tr>
<tr>
<td><strong>abstract Application</strong> getApplication()</td>
</tr>
<tr>
<td>Return the Application instance associated with this web application.</td>
</tr>
<tr>
<td><strong>abstract Iterator&lt;String&gt;</strong> getClientIdsWithMessages()</td>
</tr>
<tr>
<td>Return an Iterator over the client identifiers for which at least one FacesMessage has been queued.</td>
</tr>
<tr>
<td><strong>static FacesContext</strong> getCurrentInstance()</td>
</tr>
<tr>
<td>Return the FacesContext instance for the request that is being processed by the current thread, if any.</td>
</tr>
<tr>
<td><strong>ELContext</strong> getELContext()</td>
</tr>
<tr>
<td>Return the ELContext instance for this FacesContext instance.</td>
</tr>
<tr>
<td><strong>abstract ExternalContext</strong> getExternalContext()</td>
</tr>
<tr>
<td>Return the ExternalContext instance for this FacesContext instance.</td>
</tr>
<tr>
<td><strong>abstract FacesMessage.Severity</strong> getMaximumSeverity()</td>
</tr>
<tr>
<td>Return the maximum severity level recorded on any FacesMessages that has been queued, whether or not they are associated with any specific UIComponent.</td>
</tr>
<tr>
<td><strong>abstract Iterator&lt;FacesMessage&gt;</strong> getMessages()</td>
</tr>
<tr>
<td>Return an Iterator over the FacesMessages that have been queued, whether or not they are associated with any specific client identifier.</td>
</tr>
<tr>
<td><strong>getMessages(String clientId)</strong></td>
</tr>
</tbody>
</table>
| Return an Iterator over the FacesMessages that have been queued that are associated with the
### Getters and Setters

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iterator&lt;FacesMessage&gt;</td>
<td>Returns an iterator over the FacesMessages that have been queued that are not associated with any specific client identifier (if clientId is null).</td>
</tr>
<tr>
<td>getRenderKit()</td>
<td>Return the RenderKit instance for the render kit identifier specified on our UIViewRoot, if there is one.</td>
</tr>
<tr>
<td>getRenderResponse()</td>
<td>Return true if the renderResponse() method has been called for the current request.</td>
</tr>
<tr>
<td>getResponseComplete()</td>
<td>Return true if the responseComplete() method has been called for the current request.</td>
</tr>
<tr>
<td>getResponseStream()</td>
<td>Return the ResponseStream to which components should direct their binary output.</td>
</tr>
<tr>
<td>getResponseWriter()</td>
<td>Return the ResponseWriter to which components should direct their character-based output.</td>
</tr>
<tr>
<td>getViewRoot()</td>
<td>Return the root component that is associated with the this request.</td>
</tr>
<tr>
<td>release()</td>
<td>Release any resources associated with this FacesContext instance.</td>
</tr>
<tr>
<td>renderResponse()</td>
<td>Signal the JavaServer faces implementation that, as soon as the current phase of the request processing lifecycle has been completed, control should be passed to the Render Response phase, bypassing any phases that have not been executed yet.</td>
</tr>
<tr>
<td>responseComplete()</td>
<td>Signal the JavaServer Faces implementation</td>
</tr>
</tbody>
</table>
Abstract void setResponseStream(ResponseStream responseStream)
Set the ResponseStream to which components should direct their binary output.

Abstract void setResponseWriter(ResponseWriter responseWriter)
Set the ResponseWriter to which components should direct their character-based output.

Abstract void setViewRoot(UIViewRoot root)
Set the root component that is associated with this request.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public FacesContext()

FacesContext

public FacesContext()

Method Detail

Abstract public Application getApplication()
Web Application

Throws IllegalStateException:

getApplication

public abstract Application getApplication()

Return the Application instance associated with this web application.

Throws:

IllegalStateException - if this method is called after this instance has been released

abstract public java.util.Iterator<E> getClientIdsWithMessages()

javax.faces.application.FacesMessage Iterator
Iterator null Iterator #addMessage

Throws IllegalStateException:

ggetClientIdsWithMessages

public abstract Iterator<String> getClientIdsWithMessages()

Return an Iterator over the client identifiers for which at least one FacesMessage has been queued. If there are no such client identifiers, an empty Iterator is returned. If any messages have been queued that were not associated with any specific client identifier, a null value will be included in the iterated values. The elements in the Iterator must be returned in the order in which they were added with addMessage(java.lang.String, javax.faces.application.FacesMessage).
Throws:

- **IllegalStateException** - if this method is called after this instance has been released

```java
public ELContext getELContext()
```

Return the `ELContext` instance for this `FacesContext` instance. This `ELContext` instance has the same lifetime and scope as the `FacesContext` instance with which it is associated, and may be created lazily the first time this method is called for a given `FacesContext` instance. Upon creation of the `ELContext` instance, the implementation must take the following action:

- Call the `ELContext.putContext(java.lang.Class, FacesContext, ELContext)`
java.lang.Object method on the instance, passing in FacesContext.class and the this reference for the FacesContext instance itself.

- If the Collection returned by Application.getELContextListeners() is non-empty, create an instance of ELContextEvent and pass it to each ELContextListener instance in the Collection by calling the ELContextListener.contextCreated(javax.el.ELContextEvent) method.

The default implementation throws UnsupportedOperationException and is provided for the sole purpose of not breaking existing applications that extend this class.

**Throws:**
- IllegalStateException - if this method is called after this instance has been released

**Since:**
1.2

abstract public ExternalContext getExternalContext()

FacesContext ExternalContext

Throws IllegalStateException:

getExternalContext

public abstract ExternalContext getExternalContext()

Return the ExternalContext instance for this FacesContext instance.

**Throws:**
- IllegalStateException - if this method is called after this instance has been released
abstract public **FacesMessage.Severity** getMaximumSeverity()

```java
javax.faces.application.FacesMessage getMaximumSeverity()
```

Throws `IllegalStateException`: 

**getMaximumSeverity**

public abstract **FacesMessage.Severity** getMaximumSeverity()

Return the maximum severity level recorded on any **FacesMessage**s that has been queued, whether or not they are associated with any specific **UIComponent**. If no such messages have been queued, return `null`.

**Throws:**

`IllegalStateException` - if this method is called after this instance has been released

abstract public `java.util.Iterator<E>` getMessages()

```java
javax.faces.application.FacesMessage getMessages()
```

Throws `IllegalStateException`: 

**getMessages**

public abstract `Iterator<FacesMessage>` getMessages()

Return an `Iterator` over the **FacesMessage**s that have been queued, whether or not they are associated with any specific client identifier.
If no such messages have been queued, return an empty Iterator. The elements of the Iterator must be returned in the order in which they were added with calls to `addMessage(java.lang.String, javax.faces.application.FacesMessage)`.

**Throws:**

`IllegalStateException` - if this method is called after this instance has been released

abstract public java.util.Iterator<E> getMessages(String clientId)

```java
clientId null javax.faces.application.FacesMessage
clientId null javax.faces.application.FacesMessage
Iterator IteratorIterator #addMessage
```

`getMessages`

public abstract Iterator<FacesMessage> getMessages(String clientId)

Return an Iterator over the `FacesMessage`s that have been queued that are associated with the specified client identifier (if clientId is not null), or over the `FacesMessage`s that have been queued that are not associated with any specific client identifier (if clientId is null). If no such messages have been queued, return an empty Iterator. The elements of the Iterator must be returned in the order in which they were added with calls to `addMessage(java.lang.String, javax.faces.application.FacesMessage)`.

**Parameters:**

- `clientId` - The client identifier for which messages are requested, or null for messages not associated with any client identifier
abstract public RenderKit getRenderKit()

```java
UIViewRoot RenderKit UIViewRoot UIViewRoot
renderKitId RenderKit null
```

definition: getRenderKit

```java
public abstract RenderKit getRenderKit()
```

Return the RenderKit instance for the render kit identifier specified on our UIViewRoot, if there is one. If there is no current UIViewRoot, if the UIViewRoot does not have a specified renderKitId, or if there is no RenderKit for the specified identifier, return null instead.

abstract public boolean getRenderResponse()

```java
renderResponse() true
```

Throws: IllegalStateException

definition: getRenderResponse

```java
public abstract boolean getRenderResponse()
```

Return true if the renderResponse() method has been called for the current request.

Throws: IllegalStateException - if this method is called after this
abstract public boolean getResponseComplete()

responseComplete() true

Throws IllegalStateException:

getResponseComplete

public abstract boolean getResponseComplete()  

Return true if the responseComplete() method has been called for the current request.

Throws:  
_IllegalArgumentException_ - if this method is called after this instance has been released

abstract public ResponseStream getResponseStream()

ResponseStream ResponseStream ResponseWriter

Throws IllegalStateException:

getAddressStream

public abstract ResponseStream getResponseStream() 

Return the ResponseStream to which components should direct their binary output. Within a given response, components can use either the ResponseStream or the ResponseWriter, but not both.
abstract public void setResponseStream(ResponseStream responseStream)

Parameters:
responseStream - The new ResponseStream for this response

Throws:
NullPointerException - if responseStream is null
IllegalStateException - if this method is called after this instance has been released

abstract public ResponseWriter getResponseWriter()

Parameters:

Throws:
IllegalStateException:
getResponseWriter

public abstract ResponseWriter getResponseWriter()

Return the ResponseWriter to which components should direct their character-based output. Within a given response, components can use either the ResponseStream or the ResponseWriter, but not both.

**Throws:**

IllegalStateException - if this method is called after this instance has been released

---

abstract public void setResponseWriter(ResponseWriter responseWriter)

**Parameters:**

responseWriter - The new ResponseWriter for this response

**Throws:**

IllegalStateException: if this method is called after this instance has been released

NullPointerException: if responseWriter is null

---

setResponseWriter

public abstract void setResponseWriter(ResponseWriter responseWriter)

Set the ResponseWriter to which components should direct their character-based output.

**Parameters:**

responseWriter - The new ResponseWriter for this response

**Throws:**

IllegalStateException: if this method is called after this instance has been released

NullPointerException: if responseWriter is null
abstract public UIViewRoot getViewRoot()

Throws: IllegalStateException:

getViewRoot

public abstract UIViewRoot getViewRoot()

Return the root component that is associated with the this request.

Throws:

IllegalStateException - if this method is called after this instance has been released

abstract public void setViewRoot(UIViewRoot root)

Throws: IllegalStateException:

Throws: NullPointerException: root null

setViewRoot

public abstract void setViewRoot(UIViewRoot root)

Set the root component that is associated with this request. This method can only be called by the application handler (or a class that the handler calls), and only during the Invoke Application phase of the request processing lifecycle.

Parameters:

root - The new component UIViewRoot component
abstract public void addMessage(String clientId, FacesMessage message)

Parameters:
- clientId - The client identifier with which this message is associated (if any)
- message - The message to be appended

Throws:
- IllegalStateException - if this method is called after this instance has been released
- NullPointerException - if message is null
abstract public void release()

FacesContext Faces FacesContextFactory
FacesContext release() FacesContext
IllegalStateException

#setCurrentInstance null FacesContext

Throws IllegalStateException:

release

public abstract void release()

Release any resources associated with this FacesContext instance. Faces implementations may choose to pool instances in the associated FacesContextFactory to avoid repeated object creation and garbage collection. After release() is called on a FacesContext instance (until the FacesContext instance has been recycled by the implementation for re-use), calling any other methods will cause an IllegalStateException to be thrown.

The implementation must call
setCurrentInstance(javax.faces.context.FacesContext) passing null to remove the association between this thread and this dead FacesContext instance.

Throws:
IllegalStateException - if this method is called after this instance has been released

abstract public void renderResponse()

JavaServer faces

Throws IllegalStateException:
renderResponse

public abstract void renderResponse()

Signal the JavaServer faces implementation that, as soon as the current phase of the request processing lifecycle has been completed, control should be passed to the Render Response phase, bypassing any phases that have not been executed yet.

Throws:

IllegalStateException - if this method is called after this instance has been released

abstract public void responseComplete()

JavaServer Faces  HTTP  HTTP

Throws

IllegalStateException:

responseComplete

public abstract void responseComplete()

Signal the JavaServer Faces implementation that the HTTP response for this request has already been generated (such as an HTTP redirect), and that the request processing lifecycle should be terminated as soon as the current phase is completed.

Throws:

IllegalStateException - if this method is called after this instance has been released

public static FacesContext getCurrentInstance()
FacesContext

currentInstance

public static FacesContext getCurrentInstance()

Return the FacesContext instance for the request that is being processed by the current thread, if any.

protected static void setCurrentInstance(FacesContext context)

setCurrentInstance

protected static void setCurrentInstance(FacesContext context)

Set the FacesContext instance for the request that is being processed by the current thread.

Parameters:

context - The FacesContext instance for the current thread, or null if this thread no longer has a FacesContext instance.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMEs</td>
<td>NO FRAMEs</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.faces.context  **Class FacesContextFactory**

java.lang.Object  
  | javax.faces.context.FacesContextFactory

public abstract class **FacesContextFactory**
extends **Object**

**FacesContextFactory**

**FacesContext**  release()

JavaServer Faces Web  FacesContextFactory

FacesContextFactory factory = (FacesContextFactory)
FactoryFinder.getFactory(FactoryFinder.FACES_CONTEXT_FACTORY);

**FacesContextFactory** is a factory object that creates (if needed) and returns new **FacesContext** instances, initialized for the processing of the specified request and response objects. Implementations may take advantage of the calls to the release() method of the allocated **FacesContext** instances to pool and recycle them, rather than creating a new instance every time.

There must be one **FacesContextFactory** instance per web application that is utilizing JavaServer Faces. This instance can be acquired, in a portable manner, by calling:

```
FacesContextFactory factory = (FacesContextFactory)
FactoryFinder.getFactory(FactoryFinder.FACES_CONTEXT_FACTORY);
```

---

**Constructor Summary**
<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getFacesContext</strong>(Object context, Object request, Object response, Lifecycle lifecycle)</td>
</tr>
<tr>
<td>Create (if needed) and return a <code>FacesContext</code> instance that is initialized for the processing of the specified request and response objects, utilizing the specified <code>Lifecycle</code> instance, for this web application.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public <code>FacesContextFactory</code>()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract public <code>FacesContext</code> getFacesContext(Object context, Object request, Object response, Lifecycle lifecycle) throws <code>FacesException</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Lifecycle FacesContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>release() <code>FacesContext</code> getCurrentInstance()</td>
</tr>
</tbody>
</table>
**FacesContext**

<table>
<thead>
<tr>
<th>context</th>
<th>servlet</th>
<th>Web</th>
<th>ServletContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td>servlet</td>
<td>ServletRequest</td>
<td></td>
</tr>
<tr>
<td>response</td>
<td>servlet</td>
<td>ServletResponse</td>
<td></td>
</tr>
<tr>
<td>lifecycle</td>
<td>Lifecycle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**getFacesContext**

```java
public abstract FacesContext getFacesContext(Object context, Object request, Object response, Lifecycle lifecycle)
    throws FacesException
```

Create (if needed) and return a `FacesContext` instance that is initialized for the processing of the specified request and response objects, utilizing the specified `Lifecycle` instance, for this web application.

The implementation of this method must ensure that calls to the `getCurrentInstance()` method of `FacesContext`, from the same thread that called this method, will return the same `FacesContext` instance until the `release()` method is called on that instance.

**Parameters:**
- `context` - In servlet environments, the `ServletContext` that is associated with this web application
- `request` - In servlet environments, the `ServletRequest` that is to be processed
- `response` - In servlet environments, the `ServletResponse` that is to be processed
- `lifecycle` - The `Lifecycle` instance being used to process this request

**Throws:**
- `FacesException` - if a `FacesContext` cannot be constructed for the
specified parameters

NullPointerException - if any of the parameters are null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.event  Class FacesEvent

java.lang.Object
   └ java.util.EventObject
      └ javax.faces.event.FacesEvent

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   ActionEvent, ValueChangeEvent

public abstract class FacesEvent
extends EventObject

Extends: java.util.EventObject
Extended by: ActionEvent, ValueChangeEvent

FacesEvent  UIComponent  FacesEvent

FacesEvent is the base class for user interface and application events that can be fired by UIComponent instances. Concrete event classes must subclass FacesEvent in order to be supported by the request processing lifecycle.

See Also:
   Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
source
### Constructor Summary

**FacesEvent**

```java
(UICOMPONENT component)
```

Construct a new event object from the specified source component.

### Method Summary

#### `UICOMPONENT getComponent()`

Return the source `UICOMPONENT` that sent this event.

#### `PhaseId getPhaseId()`

Return the identifier of the request processing phase during which this event should be delivered.

#### `abstract boolean isAppropriateListener(FacesListener listener)`

Return `true` if this `FacesListener` is an instance of a listener class that this event supports.

#### `abstract void processListener(FacesListener listener)`

Broadcast this `FacesEvent` to the specified `FacesListener`, by whatever mechanism is appropriate.

#### `void queue()`

Convenience method to queue this event for broadcast at the end of the current request processing lifecycle phase.

#### `void setPhaseId(PhaseId phaseId)`

Set the `PhaseId` during which this event will be delivered.

### Methods inherited from class java.util.EventObject

- `getSource`, `toString`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`

### Constructor Detail
public FacesEvent(UIComponent component)

    component  UIComponent
    Throws    IllegalArgumentException: component null

FacesEvent
public FacesEvent(UIComponent component)

    Construct a new event object from the specified source component.

    Parameters:
    component - Source UIComponent for this event

    Throws:
    IllegalArgumentException - if component is null

Method Detail

public UIComponent getComponent()

    UIComponent

gGetComponent

public UIComponent getComponent()

    Return the source UIComponent that sent this event.

gGetComponent

public PhaseId getPhaseId()

    PhaseId PhaseId.ANY_PHASE
getPhasId

public PhaseId getPhaseId()

    Return the identifier of the request processing phase during which this event should be delivered. Legal values are the singleton instances defined by the PhaseId class, including PhaseId.ANY_PHASE, which is the default value.

public void setPhaseId(PhaseId phaseId)

    PhaseId

    Throws IllegalArgumentException: phaseId null

setPhaseId

public void setPhaseId(PhaseId phaseId)

    Set the PhaseId during which this event will be delivered.

    Throws: IllegalArgumentException - phaseId is null.

public void queue()

    Throws IllegalStateException: UIViewRoot
public void queue()

Convenience method to queue this event for broadcast at the end of
the current request processing lifecycle phase.

**Throws:**

*IllegalStateException* - if the source component for this event
is not a descendant of a *UIViewRoot*

---

abstract public boolean

**isAppropriateListener(FacesListener** listener)

**FacesListener** true "instanceof"

listener **FacesListener**

**isAppropriateListener**

public abstract boolean **isAppropriateListener(FacesListener** listener

Return true if this *FacesListener* is an instance of a listener class
that this event supports. Typically, this will be accomplished by an
"instanceof" check on the listener class.

**Parameters:**

listener - *FacesListener* to evaluate

---

abstract public void processListener(FacesListener** listener

**FacesEvent** **FacesListener** **FacesEvent**

listener **FacesEvent** **FacesListener**

**Throws** *AbortProcessingException*: JavaServer Face
processListener

public abstract void processListener(FacesListener listener)

Broadcast this FacesEvent to the specified FacesListener, by whatever mechanism is appropriate. Typically, this will be accomplished by calling an event processing method, and passing this FacesEvent as a parameter.

Parameters:

listener - FacesListener to send this FacesEvent to

Throws:

AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces Class FacesException

java.lang.Object
  ↓ java.lang.Throwable
    ↓ java.lang.Exception
      ↓ java.lang.RuntimeException
        ↓ javax.faces.FacesException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  AbortProcessingException, ConverterException, EvaluationException, ValidatorException, ViewExpiredException

public class FacesException
  extends RuntimeException

Extends: Throwable > Exception > RuntimeException
Extended by: AbortProcessingException, ConverterException, EvaluationException, ValidatorException, ViewExpiredException

JavaServer Faces

This class encapsulates general JavaServer Faces exceptions.

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FacesException()</td>
<td>Construct a new exception with no detail message or root cause.</td>
</tr>
</tbody>
</table>
### FacesException

**FacesException(String message)**

Construct a new exception with the specified detail message and no root cause.

**FacesException(String message, Throwable cause)**

Construct a new exception with the specified detail message and root cause.

**FacesException(Throwable cause)**

Construct a new exception with the specified root cause.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable getCause()</td>
<td>Return the cause of this exception, or null if the cause is nonexistent or unknown.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

**public FacesException()**

**FacesException**

**public FacesException()**
Construct a new exception with no detail message or root cause.

public FacesException(String message)

message

FacesException

public FacesException(String message)

Construct a new exception with the specified detail message and no root cause.

Parameters:
message - The detail message for this exception

public FacesException(Throwable cause)

(cause == null ? null : cause.toString())

cause

FacesException

public FacesException(Throwable cause)

Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

Parameters:
cause - The root cause for this exception
public FacesException(String message, Throwable cause)

Construct a new exception with the specified detail message and root cause.

Parameters:
message - The detail message for this exception
cause - The root cause for this exception

Method Detail

public Throwable getCause()

Return the cause of this exception, or null if the cause is nonexistent or unknown.

Overrides:
getCaue in class Throwable
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.event Interface FacesListener

All Superinterfaces:
EventListener

All Known Subinterfaces:
ActionListener, ValueChangeListener

All Known Implementing Classes:
MethodExpressionActionListener, MethodExpressionValueChangeListener

public interface FacesListener
extends EventListener

Implements: java.util.EventListener
Implemented by: ActionListener, ValueChangeListener

javax.faces.component.StateHolder

A generic base interface for event listeners for various types of FacesEvent s. All listener interfaces for specific FacesEvent event types must extend this interface.

Implementations of this interface must have a zero-args public constructor. If the class that implements this interface has state that needs to be saved and restored between requests, the class must also implement StateHolder.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.application Class FacesMessage

java.lang.Object
  └─javax.faces.application.FacesMessage

All Implemented Interfaces:
  Serializable

public class FacesMessage
  extends Object
  implements Serializable

Implements: java.io.Serializable
Inner classes: FacesMessage.Severity

FacesMessage messageId
javax.faces.application.FacesMessage
javax.faces.application.FacesMessage messageId

messageId FacesMessage

  Application#getMessageBundle null ResourceBundle
javax.faces.component.UIViewRoot Locale ResourceBundle
messageId FacesMessage summary
Application#getMessageBundle null #FACES_MESSAGES
ResourceBundle messageId FacesMessage summary
FacesMessage

{messageId} ResourceBundle {messageId}_detail
ResourceBundle FacesMessage detail

FacesMessage summary detail

**FacesMessage** represents a single validation (or other) message, which
is typically associated with a particular component in the view. A 
FacesMessage instance may be created based on a specific messageId. 
The specification defines the set of messageId s for which there must be 
FacesMessage instances.

The implementation must take the following steps when creating 
FacesMessage instances given a messageId:

Call Application.getMessageBundle(). If non-null, locate the named 
ResourceBundle, using the Locale from the current UIViewRoot and 
see if it has a value for the argument messageId. If it does, treat the 
value as the summary of the FacesMessage. If it does not, or if 
Application.getMessageBundle() returned null, look in the 
ResourceBundle named by the value of the constant FACES_MESSAGES 
and see if it has a value for the argument messageId. If it does, treat 
the value as the summary of the FacesMessage. If it does not, there is 
no initialization information for the FacesMessage instance.

In all cases, if a ResourceBundle hit is found for the {messageId}, look 
for further hits under the key {messageId}_detail. Use this value, if 
present, as the detail for the returned FacesMessage.

Make sure to perform any parameter substitution required for the 
summary and detail of the FacesMessage.

See Also: 
Serialized Form

### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>FacesMessage.Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class used to represent message severity levels in a typesafe enumeration.</td>
<td></td>
</tr>
</tbody>
</table>

### Field Summary

<table>
<thead>
<tr>
<th>FACES_MESSAGES</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceBundle identifier for messages</td>
</tr>
<tr>
<td>static String whose message identifiers are defined in the JavaServer Faces specification.</td>
</tr>
<tr>
<td>static FacesMessage.Severity SEVERITY_ERROR Message severity level indicating that an error has occurred.</td>
</tr>
<tr>
<td>static FacesMessage.Severity SEVERITY_FATAL Message severity level indicating that a serious error has occurred.</td>
</tr>
<tr>
<td>static FacesMessage.Severity SEVERITY_INFO Message severity level indicating an informational message rather than an error.</td>
</tr>
<tr>
<td>static FacesMessage.Severity SEVERITY_WARN Message severity level indicating that an error might have occurred.</td>
</tr>
<tr>
<td>static List VALUES Immutable List of valid FacesMessage.Severity instances, in ascending order of their ordinal value.</td>
</tr>
<tr>
<td>static Map VALUES_MAP Immutable Map of valid FacesMessage.Severity instances, keyed by name.</td>
</tr>
</tbody>
</table>

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FacesMessage()</td>
<td>Construct a new FacesMessage with no initial values.</td>
</tr>
<tr>
<td>FacesMessage(FacesMessage.Severity severity, String summary, String detail)</td>
<td>Construct a new FacesMessage with the specified initial values.</td>
</tr>
<tr>
<td>FacesMessage(String summary)</td>
<td>Construct a new FacesMessage with just a summary.</td>
</tr>
<tr>
<td>FacesMessage(String summary, String detail)</td>
<td>Construct a new FacesMessage with the specified initial values.</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><code>getDetail()</code></td>
<td>Return the localized detail text.</td>
</tr>
<tr>
<td>FacesMessage.Severity</td>
<td><code>getSeverity()</code></td>
<td>Return the severity level.</td>
</tr>
<tr>
<td>String</td>
<td><code>getSummary()</code></td>
<td>Return the localized summary text.</td>
</tr>
<tr>
<td>void</td>
<td><code>setDetail(String detail)</code></td>
<td>Set the localized detail text.</td>
</tr>
<tr>
<td>void</td>
<td><code>setSeverity(FacesMessage.Severity severity)</code></td>
<td>Set the severity level.</td>
</tr>
<tr>
<td>void</td>
<td><code>setSummary(String summary)</code></td>
<td>Set the localized summary text.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

**FACES_MESSAGES**

```java
public static final String FACES_MESSAGES
```

ResourceBundle identifier for messages whose message identifiers are defined in the JavaServer Faces specification.

**See Also:**

Constant Field Values
SEVERITY_INFO

public static final FacesMessage.Severity SEVERITY_INFO

Message severity level indicating an informational message rather than an error.

SEVERITY_WARN

public static final FacesMessage.Severity SEVERITY_WARN

Message severity level indicating that an error might have occurred.

SEVERITY_ERROR

public static final FacesMessage.Severity SEVERITY_ERROR

Message severity level indicating that an error has occurred.

SEVERITY_FATAL

public static final FacesMessage.Severity SEVERITY_FATAL

Message severity level indicating that a serious error has occurred.

VALUES

public static final List VALUES
Immutable List of valid `FacesMessage.Severity` instances, in ascending order of their ordinal value.

VALUES_MAP

```java
public static final Map VALUES_MAP
```

Immutable Map of valid `FacesMessage.Severity` instances, keyed by name.

Constructor Detail

```java
public FacesMessage()
```

```java
javax.faces.application.FacesMessage Severity.INFO
```

FacesMessage

```java
public FacesMessage()
```

Construct a new `FacesMessage` with no initial values. The severity is set to Severity.INFO.

```java
javax.faces.application.FacesMessage null Severity.INFO
```

FacesMessage
public FacesMessage(String summary)

Construct a new FacesMessage with just a summary. The detail is null, the severity is set to Severity.INFO.

public FacesMessage(String summary, String detail)

javax.faces.application.FacesMessage Severity.INFO

summary
detail

Throws IllegalArgumentException:

FacesMessage

public FacesMessage(String summary, String detail)

Construct a new FacesMessage with the specified initial values. The severity is set to Severity.INFO.

Parameters:

summary - Localized summary message text
detail - Localized detail message text

Throws:

IllegalArgumentException - if the specified severity level is not one of the supported values

public FacesMessage(FacesMessage.Severity severity, String summary, String detail)

FacesMessage

severity
summary
detail

Throws IllegalArgumentException:

**FacesMessage**

```java
public FacesMessage(FacesMessage.Severity severity,
                     String summary,
                     String detail)
```

Construct a new FacesMessage with the specified initial values.

**Parameters:**
- severity - the severity
- summary - Localized summary message text
- detail - Localized detail message text

**Throws:**
- IllegalArgumentException - if the specified severity level is not one of the supported values

**Method Detail**

```java
public String getDetail()
```

**getDetail**

```java
public String getDetail()
```

Return the localized detail text. If no localized detail text has been defined for this message, return the localized summary text instead.

```java
public void setDetail(String detail)
```
setDetail

public void setDetail(String detail)

Set the localized detail text.

Parameters:
    detail - The new localized detail text

getSeverity

public FacesMessage.Severity getSeverity()

getSeverity

public FacesMessage.Severity getSeverity()

Return the severity level.

setSeverity

public void setSeverity(FacesMessage.Severity severity)

    severity
    Throws       IllegalArgumentException:
public void setSeverity(FacesMessage.Severity severity)

    Set the severity level.

    Parameters:
    severity - The new severity level

    Throws:
    IllegalArgumentException - if the specified severity level is not one of the supported values

public String getSummary()

getSummary

public String getSummary()

    Return the localized summary text.

public void setSummary(String summary)

    Set the localized summary text.

    Parameters:
    summary - The new localized summary text
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.application  **Class FacesMessage.Severity**

java.lang.Object
  ▼ javax.faces.application.FacesMessage.Severity

**All Implemented Interfaces:**  
  Comparable

**Enclosing class:**  
  FacesMessage

---

```java
public static class FacesMessage.Severity
  extends Object
  implements Comparable

Implements: Comparable<T>
Contained within: FacesMessage
```

Class used to represent message severity levels in a typesafe enumeration.

---

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int compareTo(Object other)</code></td>
<td>Compare this <code>FacesMessage.Severity</code> instance to the specified one.</td>
</tr>
<tr>
<td><code>int getOrdinal()</code></td>
<td>Return the ordinal value of this <code>FacesMessage.Severity</code> instance.</td>
</tr>
<tr>
<td><code>toString()</code></td>
<td></td>
</tr>
</tbody>
</table>
Return a String representation of this `FacesMessage.Severity` instance.

Methods inherited from class `java.lang.Object`
`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Method Detail

public int compareTo(Object other)

```
javax.faces.application.FacesMessage.Severity
0
other
```

`compareTo`

public int compareTo(Object other)

Compare this `FacesMessage.Severity` instance to the specified one. Returns a negative integer, zero, or a positive integer if this object is less than, equal to, or greater than the specified object.

Specified by: `compareTo` in interface `Comparable`
Parameters:
other - The other object to be compared to

public int getOrdinal()

`FacesMessage.Severity`
getOrdinal

```java
public int getOrdinal()
```

Return the ordinal value of this `FacesMessage.Severity` instance.

---

toString

```java
public String toString()
```

Return a String representation of this `FacesMessage.Severity` instance.

**Overrides:**

`toString` in class `Object`
javax.faces.webapp  Class FacesServlet

java.lang.Object
  └─javax.faces.webapp.FacesServlet

All Implemented Interfaces:
  Servlet

public final class FacesServlet
extends Object
implements Servlet

Implements: Servlet

FacesServlet Web servlet JavaServer Faces

FacesServlet is a servlet that manages the request processing lifecycle for web applications that are utilizing JavaServer Faces to construct the user interface.

### Field Summary

<table>
<thead>
<tr>
<th>Static</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String</strong></td>
<td><strong>CONFIG_FILES_ATTR</strong></td>
</tr>
<tr>
<td></td>
<td>Context initialization parameter name for a comma delimited list of context-relative resource paths (in addition to /WEB-INF/faces-config.xml which is loaded automatically if it exists) containing JavaServer Faces configuration information.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><strong>LIFECYCLE_ID_ATTR</strong></td>
</tr>
<tr>
<td></td>
<td>Context initialization parameter name for the lifecycle identifier of the Lifecycle instance to be utilized.</td>
</tr>
</tbody>
</table>
## Constructor Summary

| FacesServlet() |

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void destroy()</code></td>
<td>Release all resources acquired at startup time.</td>
</tr>
<tr>
<td><code>ServletConfig getServletConfig()</code></td>
<td>Return the ServletConfig instance for this servlet.</td>
</tr>
<tr>
<td><code>String getServletInfo()</code></td>
<td>Return information about this Servlet.</td>
</tr>
<tr>
<td><code>void init(ServletConfig servletConfig)</code></td>
<td>Acquire the factory instances we will require.</td>
</tr>
<tr>
<td><code>void service(ServletRequest request, ServletResponse response)</code></td>
<td>Process an incoming request, and create the corresponding response, by executing the request processing lifecycle.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
config.xml which is loaded automatically if it exists) containing JavaServer Faces configuration information.

See Also:
Constant Field Values

---

LIFECYCLE_ID_ATTR

public static final String LIFECYCLE_ID_ATTR

Context initialization parameter name for the lifecycle identifier of the Lifecycle instance to be utilized.

See Also:
Constant Field Values

---

**Constructor Detail**

public FacesServlet()

FacesServlet

public FacesServlet()

---

**Method Detail**

public void destroy()
**destroy**

```java
public void destroy()
```

Release all resources acquired at startup time.

**Specified by:**
```
destroy in interface Servlet
```

---

**public ServletConfig getServletConfig()**

```java
servlet ServletConfig
```

**getServletConfig**

```java
public ServletConfig getServletConfig()
```

Return the ServletConfig instance for this servlet.

**Specified by:**
```
getServletConfig in interface Servlet
```

**Returns:**
the ServletConfig object that initializes this servlet

**See Also:**
```
Servlet.init(javax.servlet.ServletConfig)
```

---

**public String getServletInfo()**

```java
Servlet
```

**getServletInfo**

```java
public String getServletInfo()
```
Return information about this Servlet.

Specified by:
getServletInfo in interface Servlet

Returns:
a String containing servlet information

```
public void init(ServletConfig servletConfig) throws ServletException
```

Throws ServletException: Faces

init

```
public void init(ServletConfig servletConfig)
    throws ServletException
```

Acquire the factory instances we will require.

Specified by: init in interface Servlet

Parameters:
servletConfig - a ServletConfig object containing the servlet's configuration and initialization parameters

Throws:
ServletException - if, for any reason, the startup of this Faces application failed. This includes errors in the config file that is parsed before or during the processing of this init() method.

See Also:
UnavailableException, Servlet.getServletContext()

```
public void service(ServletRequest request, ServletResponse response) throws java.io.IOException,
```
ServletException

request  response  HttpServletRequest
HttpServletResponse

HttpServletResponse  sendError
HttpServletRequest.SC_NOT_FOUND

/WEB-INF/
WEB-INF
/META-INF/
META-INF

request  servlet
response  servlet
Throws  java.io.IOException: /
Throws  ServletException: servlet

service

public void service(HttpServletRequest request,
  HttpServletResponse response)
  throws IOException,
  ServletException

Process an incoming request, and create the corresponding response, by executing the request processing lifecycle.

If the request and response arguments to this method are not instances of HttpServletRequest and HttpServletResponse, respectively, the results of invoking this method are undefined.

This method must respond to requests that start with the following strings by invoking the sendError method on the response argument (cast to HttpServletResponse), passing the code
HttpServletResponse.SC_NOT_FOUND as the argument.

/WEB-INF/
/WEB-INF
/META-INF/
/META-INF

Specified by:
service in interface Servlet

Parameters:
request - The servlet request we are processing
response - The servlet response we are creating

Throws:
IOException - if an input/output error occurs during processing
ServletException - if a servlet error occurs during processing
javax.faces.webapp Class FacetTag

java.lang.Object
   ▼ javax.servlet.jsp.tagext.TagSupport
      ▼ javax.faces.webapp.FacetTag

All Implemented Interfaces:
   Serializable, IterationTag, JspTag, Tag

public class FacetTag

extends TagSupport

Extends: TagSupport

FacetTag JSP javax.faces.component.UIComponent facet

FacetTag javax.faces.component.UIComponent UIComponentTag

FacetTag is the JSP mechanism for denoting a UIComponent is to be added as a facet to the component associated with its parent.

A FacetTag must have one and only one child. This child must be a UIComponentTag instance representing a single UIComponent instance.

See Also:
   Serialized Form

---

Field Summary

| Fields inherited from class javax.servlet.jsp.tagext.TagSupport | |
### Fields inherited from interface `javax.servlet.jsp.tagext.IterationTag`
- EVAL_BODY_AGAIN

### Fields inherited from interface `javax.servlet.jsp.tagext.Tag`
- EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

## Constructor Summary

**FacetTag()**

## Method Summary

### `doStartTag()`
int

Return EVAL_BODY_INCLUDE to cause nested body content to be evaluated.

### `getName()`
String

Return the name to be assigned to this facet.

### `release()`
void

Release any resources allocated by this tag instance.

### `setName(String name)`
void

Set the name to be assigned to this facet.

### Methods inherited from class `javax.servlet.jsp.tagext.TagSupport`
- doAfterBody, doEndTag, findAncestorWithClass, getId, getParent, getValue, getValues, removeValue, setId, setPageContext, setParent, setValue

### Methods inherited from class `java.lang.Object`
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
public FacetTag()

FacetTag

public FacetTag()

Method Detail

public String getName()

getName

public String getName()

Return the name to be assigned to this facet.

public void setName(String name)

setName

public void setName(String name)

Set the name to be assigned to this facet.
Parameters:
- name - The new facet name

public void release()

release

public void release()

Release any resources allocated by this tag instance.

Specified by:
- release in interface Tag

Overrides:
- release in class TagSupport

See Also:
- Tag.release()

public int doStartTag() throws JspException

EVAL_BODY_INCLUDE

doStartTag

public int doStartTag()
throws JspException

Return EVAL_BODY_INCLUDE to cause nested body content to be evaluated.

Specified by:
doStartTag in interface Tag

Overrides:
doStartTag in class TagSupport

Returns:
SKIP_BODY

Throws:
JspException - if an error occurs while processing this tag

See Also:
Tag.doStartTag()
javax.xml.stream   Class FactoryConfigurationError

java.lang.Object
   └ java.lang.Throwable
      └ java.lang.Error
         └ javax.xml.stream.FactoryConfigurationError

All Implemented Interfaces:
   Serializable

public class FactoryConfigurationError
   extends Error

Extends: Throwable > Error

version          1.0

An error class for reporting factory configuration errors.

Version:
   1.0

Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

See Also:
   Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryConfigurationError()</td>
<td>Default constructor</td>
</tr>
<tr>
<td>FactoryConfigurationError(Exception e)</td>
<td>Construct an exception with a nested inner exception</td>
</tr>
<tr>
<td>FactoryConfigurationError(Exception e, String msg)</td>
<td></td>
</tr>
</tbody>
</table>
Construct an exception with a nested inner exception and a message

```java
FactoryConfigurationError(String msg)
```

Construct an exception with associated message

```java
FactoryConfigurationError(String msg, Exception e)
```

### Method Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th><code>getException()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the nested exception (if any)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>getMessage()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Report the message associated with this error</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public FactoryConfigurationError(Exception e)

    e

FactoryConfigurationError

public FactoryConfigurationError(Exception e)

    Construct an exception with a nested inner exception

    Parameters:
    e - the exception to nest

-----------------------------

public FactoryConfigurationError(Exception e, String msg)

    e

    msg

FactoryConfigurationError

public FactoryConfigurationError(Exception e, String msg)

    Construct an exception with a nested inner exception and a message

    Parameters:
    e - the exception to nest
    msg - the message to report

-----------------------------

public FactoryConfigurationError(String msg, Exception e)
FactoryConfigurationError

public FactoryConfigurationError(String msg, Exception e)

Construct an exception with a nested inner exception and a message

Parameters:
   msg - the message to report
   e - the exception to nest

public FactoryConfigurationError(String msg)

Construct an exception with associated message

Parameters:
   e - the exception to nest
   msg - the message to report

Method Detail

public Exception getException()
public Exception getException()

Return the nested exception (if any)

Returns:
the nested exception or null

public String getMessage()

return

ggetMessage

Report the message associated with this error

Overrides:
getMessage in class Throwable

Returns:
the string value of the message
PS:
javax.faces **Class FactoryFinder**

`java.lang.Object`
- `javax.faces.FactoryFinder`

```java
public final class FactoryFinder
extends Object
```

**FactoryFinder**  JavaServer Faces API

- **JavaServer Faces** Web  WEB-INF  factory
- `javax.faces.CONFIG_FILES`  `ServletContext`  `init` **JavaServer Faces**  factory
- **JavaServer Faces**  `ServletContext`  `jar`  **META-INF**  factory
- `META-INF/services/{factory-class-name}`  **Web**  **JAR**
- **JavaServer Faces**
  ```java
  javax.faces.context.FacesContextFactory
  Web  ClassLoader  jar  **META-INF/services/javax.faces.context.FacesContextFactory**
  FacesContextFactory  FactoryFinder  **JavaServer Faces**
  FacesContextFactory
  Factory
  Web  **JavaServer Faces**  Web  Web
  ```

**FactoryFinder** implements the standard discovery algorithm for all factory objects specified in the JavaServer Faces APIs. For a given factory class name, a corresponding implementation class is searched for
based on the following algorithm. Items are listed in order of decreasing search precedence:

- If the JavaServer Faces configuration file bundled into the `WEB-INF` directory of the webapp contains a `factory` entry of the given factory class name, that factory is used.
- If the JavaServer Faces configuration files named by the `javax.faces.CONFIG_FILES` ServletContext init parameter contain any `factory` entries of the given factory class name, those factories are used, with the last one taking precedence.
- If there are any JavaServer Faces configuration files bundled into the `META-INF` directory of any jars on the ServletContext's resource paths, the `factory` entries of the given factory class name in those files are used, with the last one taking precedence.
- If a `META-INF/services/{factory-class-name}` resource is visible to the web application class loader for the calling application (typically as a result of being present in the manifest of a JAR file), its first line is read and assumed to be the name of the factory implementation class to use.
- If none of the above steps yield a match, the JavaServer Faces implementation specific class is used.

If any of the factories found on any of the steps above happen to have a one-argument constructor, with argument the type being the abstract factory class, that constructor is invoked, and the previous match is passed to the constructor. For example, say the container vendor provided an implementation of `FacesContextFactory`, and identified it in `META-INF/services/javax.faces.context.FacesContextFactory` in a jar on the webapp ClassLoader. Also say this implementation provided by the container vendor had a one argument constructor that took a `FacesContextFactory` instance. The FactoryFinder system would call that one-argument constructor, passing the implementation of `FacesContextFactory` provided by the JavaServer Faces implementation.

If a Factory implementation does not provide a proper one-argument constructor, it must provide a zero-arguments constructor in order to be successfully instantiated.

Once the name of the factory implementation class is located, the web
application class loader for the calling application is requested to load this class, and a corresponding instance of the class will be created. A side effect of this rule is that each web application will receive its own instance of each factory class, whether the JavaServer Faces implementation is included within the web application or is made visible through the container’s facilities for shared libraries.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String APPLICATION_FACTORY</td>
</tr>
<tr>
<td>static String FACES_CONTEXT_FACTORY</td>
</tr>
<tr>
<td>static String LIFECYCLE_FACTORY</td>
</tr>
<tr>
<td>static String RENDER_KIT_FACTORY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static Object getFactory(String factoryName)</td>
</tr>
<tr>
<td>static void releaseFactories()</td>
</tr>
<tr>
<td>static void setFactory(String factoryName, String implName)</td>
</tr>
</tbody>
</table>
getFactory(java.lang.String) will find this mapping when searching for a match.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

APPLICATION_FACTORY
public static final String APPLICATION_FACTORY

The property name for the ApplicationFactory class name.

See Also:
Constant Field Values

FACES_CONTEXT_FACTORY
public static final String FACES_CONTEXT_FACTORY

The property name for the FacesContextFactory class name.

See Also:
Constant Field Values

LIFECYCLE_FACTORY
public static final String LIFECYCLE_FACTORY

The property name for the LifecycleFactory class name.

See Also:
Constant Field Values

public static final String RENDER_KIT_FACTORY

The property name for the RenderKitFactory class name.

See Also:
Constant Field Values

Method Detail

public static Object getFactory(String factoryName) throws FacesException

JavaServer Faces  Web

factoryName  JavaServer Faces
Throws  FacesException:  Web
Throws  FacesException:
Throws  FacesException:
Throws  IllegalArgumentException:  factoryName  JavaServer
Faces
Throws  IllegalStateException:
Throws  NullPointerException:  factoryname  null

getFactory
public static Object getFactory(String factoryName) throws FacesException

Create (if necessary) and return a per-web-application instance of the appropriate implementation class for the specified JavaServer Faces factory class, based on the discovery algorithm described in the class description.

Parameters:
  factoryName - Fully qualified name of the JavaServer Faces factory for which an implementation instance is requested

Throws:
  FacesException - if the web application class loader cannot be identified
  FacesException - if an instance of the configured factory implementation class cannot be loaded
  FacesException - if an instance of the configured factory implementation class cannot be instantiated
  IllegalArgumentException - if factoryName does not identify a standard JavaServer Faces factory name
  IllegalStateException - if there is no configured factory implementation class for the specified factory name
  NullPointerException - if factoryname is null

public static void setFactory(String factoryName, String implName)

getFactory()  factoryName

Faces

Throws  IllegalArgumentException:   factoryName  JavaServer Faces
  NullPointerException:   factoryname  null
**setFactory**

```java
public static void setFactory(String factoryName, String implName)
```

This method will store the argument `factoryName/implName` mapping in such a way that `getFactory(java.lang.String)` will find this mapping when searching for a match.

This method has no effect if `getFactory()` has already been called looking for a factory for this `factoryName`.

This method can be used by implementations to store a factory mapping while parsing the Faces configuration file.

**Throws:**

- `IllegalArgumentException` - if `factoryName` does not identify a standard JavaServer Faces factory name
- `NullPointerException` - if `factoryName` is null

---

**public static void releaseFactories() throws FacesException**

**Web**  
JavaServer Faces API  
Web

Throws  

`FacesException`: Web

---

**releaseFactories**

```java
public static void releaseFactories()
throws FacesException
```

Release any references to factory instances associated with the class loader for the calling web application. This method should be called as apart of web application shutdown in a container where the JavaServer Faces API classes are part of the container itself, rather
than being included inside the web application.

**Throws:**

*FacesException* - if the web application class loader cannot be identified
javax.xml.registry Interface FederatedConnection

All Superinterfaces: 
  Connection

public interface FederatedConnection
extends Connection

Implements: Connection

JAXR API

Represents a single logical connection to a federation or group of registry providers. This interface is used in support of the distributed query feature of the JAXR API.

Author:
  Farrukh S. Najmi

Method Summary

Methods inherited from interface javax.xml.registry.Connection
close, getCredentials, getRegistryService, isClosed, isSynchronous, setCredentials, setSynchronous

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
Clients use a FetchProfile to list the Message attributes that it wishes to prefetch from the server for a range of messages.

Messages obtained from a Folder are light-weight objects that typically start off as empty references to the actual messages. Such a Message object is filled in "on-demand" when the appropriate get*() methods are invoked on that particular Message. Certain server-based message
access protocols (Ex: IMAP) allow batch fetching of message attributes for a range of messages in a single request. Clients that want to use message attributes for a range of Messages (Example: to display the top-level headers in a headerlist) might want to use the optimization provided by such servers. The FetchProfile allows the client to indicate this desire to the server.

Note that implementations are not obligated to support FetchProfiles, since there might be cases where the backend service does not allow easy, efficient fetching of such profiles.

Sample code that illustrates the use of a FetchProfile is given below:

```java
Message[] msgs = folder.getMessages();

FetchProfile fp = new FetchProfile();
fp.add(FetchProfile.Item.ENVELOPE);
fp.add("X-mailer");
folder.fetch(msgs, fp);
```

**Author:**
John Mani, Bill Shannon

**See Also:**
Folder.fetch(javax.mail.Message[], javax.mail.FetchProfile)

## Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>FetchProfile.Item</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This inner class is the base class of all items that can be requested in a FetchProfile.</td>
</tr>
</tbody>
</table>

## Constructor Summary

<table>
<thead>
<tr>
<th>FetchProfile()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an empty FetchProfile.</td>
</tr>
</tbody>
</table>

## Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void add(FetchProfile.Item item)</code></td>
<td>Add the given special item as one of the attributes to be prefetched.</td>
</tr>
<tr>
<td><code>void add(String headerName)</code></td>
<td>Add the specified header-field to the list of attributes to be prefetched.</td>
</tr>
<tr>
<td><code>boolean contains(FetchProfile.Item item)</code></td>
<td>Returns true if the fetch profile contains given special item.</td>
</tr>
<tr>
<td><code>boolean contains(String headerName)</code></td>
<td>Returns true if the fetch profile contains given header name.</td>
</tr>
<tr>
<td><code>String[] getHeaderNames()</code></td>
<td>Get the names of the header-fields set in this profile.</td>
</tr>
<tr>
<td><code>FetchProfile.Item[] getItems()</code></td>
<td>Get the items set in this profile.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public void add(Envelope.Item item)

Add the given special item as one of the attributes to be prefetched.

Parameters:
item - the special item to be fetched
See Also: ENVELOPE, CONTENT_INFO, FLAGS

public void add(String headerName)

Add the specified header-field to the list of attributes to be prefetched.

Parameters:
headerName - header to be prefetched
public boolean contains(FetchProfile.Item item)
    
contains

public boolean contains(FetchProfile.Item item)
    
    Returns true if the fetch profile contains given special item.

public boolean contains(String headerName)
    
contains

public boolean contains(String headerName)
    
    Returns true if the fetch profile contains given header name.

public FetchProfile.Item[] getItems()
    
getItems

public FetchProfile.Item[] getItems()
    
    Get the items set in this profile.

    Returns:
    items set in this profile
public String[] getHeaderNames()

return

g.getHeaderNames

public String[] getHeaderNames()

Get the names of the header-fields set in this profile.

Returns:
headers set in this profile
This inner class is the base class of all items that can be requested in a FetchProfile. The items currently defined here are ENVELOPE, CONTENT_INFO and FLAGS. The UIDFolder interface defines the UID Item as well.

Note that this class only has a protected constructor, thereby restricting new Item types to either this class or subclasses. This effectively implements a enumeration of allowed Item types.
Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static FetchProfile.Item</td>
<td>CONTENT_INFO</td>
</tr>
<tr>
<td>static FetchProfile.Item</td>
<td>ENVELOPE</td>
</tr>
<tr>
<td>static FetchProfile.Item</td>
<td>FLAGS</td>
</tr>
</tbody>
</table>

Constructor Summary

protected FetchProfile.Item(String name)

Constructor for an item.

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

ENVELOPE

public static final FetchProfile.Item ENVELOPE

This is the Envelope item.

The Envelope is an aggregation of the common attributes of a Message. Implementations should include the following attributes: From, To, Cc, Bcc, ReplyTo, Subject and Date. More items may be included as well.
For implementations of the IMAP4 protocol (RFC 2060), the Envelope should include the ENVELOPE data item. More items may be included too.

CONTENT_INFO

public static final FetchProfile.Item CONTENT_INFO

This item is for fetching information about the content of the message.

This includes all the attributes that describe the content of the message. Implementations should include the following attributes: ContentType, ContentDisposition, ContentDescription, Size and LineCount. Other items may be included as well.

FLAGS

public static final FetchProfile.Item FLAGS

This is the Flags item.

Constructor Detail

protected FetchProfile.Item(String name)

FetchProfile.Item

protected FetchProfile.Item(String name)
Constructor for an item. The name is used only for debugging.
Description: Enum

**javax.persistence**

**Enum**

**FetchType**

**java.lang.Object**

**java.lang.Enum<FetchType>**

**javax.persistence.FetchType**

**All Implemented Interfaces:**

*Serializable, Comparable<FetchType>*

---

```java
public enum FetchType
    extends Enum<FetchType>

    Extends: Enum<E>

    EAGER      LAZY      LAZY

    @Basic(fetch=LAZY)
    protected String getName() { return name; }
```

**since**

Java Persistence 1.0

---

Defines strategies for fetching data from the database. The **EAGER** strategy is a requirement on the persistence provider runtime that data must be eagerly fetched. The **LAZY** strategy is a hint to the persistence provider runtime that data should be fetched lazily when it is first accessed. The implementation is permitted to eagerly fetch data for which the **LAZY** strategy hint has been specified. In particular, lazy fetching might only be available for **Basic** mappings for which property-based access is used.

Example:

```java
    @Basic(fetch=LAZY)
    protected String getName() { return name; }
```

**Since:**
Enum Constant Summary

**EAGER**
Defines that data must be eagerly fetched

**LAZY**
Defines that data can be lazily fetched

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Enum**

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, `valueOf`

Methods inherited from class java.lang.**Object**

finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

**LAZY**

public static final **FetchType** **LAZY**

Defines that data can be lazily fetched
EAGER

public static final FetchType EAGER

Defines that data must be eagerly fetched

**Method Detail**

**final public static FetchType[] values()**

```java
public static final FetchType[] values()
```

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(FetchType c : FetchType.values())
    System.out.println(c);
```

**Returns:**

an array containing the constants of this enum type, in the order they're declared

**public static FetchType.valueOf(String name)**

```java
public static FetchType.valueOf(String name)
```

Returns the enum constant of this type with the specified name. The
string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:

   name - the name of the enum constant to be returned.

Returns:

   the enum constant with the specified name

Throws:

   IllegalArgumentException - if this enum type has no constant with the specified name
javax.persistence  Annotation Type FieldResult

@Target(value={})
@Retention(value=RUNTIME)
public @interface FieldResult

Implements: Annotation
@Target(value={})
@Retention(value=RUNTIME)

SELECT

Query q = em.createNativeQuery(
   "SELECT o.id AS order_id, " +
   "o.quantity AS order_quantity, " +
   "o.item AS order_item, " +
   "FROM Order o, Item i " +
   "WHERE (order_quantity > 25) AND (order_item = i.id)",
   "OrderResults");

@SqlResultSetMapping(name="OrderResults",
   entities={
      @EntityResult(entityClass=com.acme.Order.class,
         fields={
            @FieldResult(name="id", column="order_id"),
            @FieldResult(name="quantity", column="order_quantity",
            @FieldResult(name="item", column="order_item"))
         }
      })
)

since Java Persistence 1.0  

Is used to map the columns specified in the SELECT list of the query to the properties or fields of the entity class.

Example:
Query q = em.createNativeQuery(
   "SELECT o.id AS order_id, " +
   "o.quantity AS order_quantity, " +
   "o.item AS order_item, " +
   "FROM Order o, Item i " +
   "WHERE (order_quantity > 25) AND (order_item = i.id)",
   "OrderResults"),

   @ resultListMapping(name="OrderResults",
      entities={
         @EntityResult(entityClass=com.acme.Order.class,
            fields={
               @FieldResult(name="id", column="order_id"),
               @FieldResult(name="quantity", column="order_quantity",
               @FieldResult(name="item", column="order_item"))
            }
         })
   )
)
"OrderResults";

@SqlResultSetMapping(name="OrderResults",
   entities=
       @EntityResult(entityClass=com.acme.Order.class,
             fields={
                @FieldResult(name="id", column="order_id"),
                @FieldResult(name="quantity", column="order_quantity",
                @FieldResult(name="item", column="order_item"))
           })
   } }

Since:
Java Persistence 1.0

---

**Required Element Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>column</td>
</tr>
<tr>
<td></td>
<td>Name of the column in the SELECT clause - i.e., column aliases, if applicable.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
</tr>
<tr>
<td></td>
<td>Name of the persistent field or property of the class.</td>
</tr>
</tbody>
</table>

---

**Element Detail**

abstract public String name()

name

public abstract String name

    Name of the persistent field or property of the class.

abstract public String column()

SELECT
public abstract String column

Name of the column in the SELECT clause - i.e., column aliases, if applicable.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.activation Class FileDataSource

java.lang.Object
  javax.activation.FileDataSource

All Implemented Interfaces:
  DataSource

public class FileDataSource
  extends Object
  implements DataSource

Implements: DataSource

FileDataSource  DataSource  FileTypeMap

FileDataSource

FileDataSource  FileTypeMap  setFileTypeMap
FileDataSource  FileTypeMap  FileTypeMap  FileDataSource
FileTypeMap  getDefaultFileTypeMap  FileTypeMap

See  javax.activation.DataSource,  javax.activation.FileTypeMap,
also  javax.activation.MimetypesFileTypeMap

The FileDataSource class implements a simple DataSource object that
encapsulates a file. It provides data typing services via a FileTypeMap
object.

FileDataSource Typing Semantics

The FileDataSource class delegates data typing of files to an object
subclassed from the FileTypeMap class. The setFileTypeMap method can
be used to explicitly set the FileTypeMap for an instance of
FileDataSource. If no FileTypeMap is set, the FileDataSource will call the
FileTypeMap's getDefaultFiletypeMap method to get the System's
default FileTypeMap.

See Also:

DataSource, FileTypeMap, MimetypesFileTypeMap

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
<th>Method Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FileDataSource(File file)</td>
<td></td>
<td>Creates a FileDataSource from a File object.</td>
</tr>
<tr>
<td>FileDataSource(String name)</td>
<td></td>
<td>Creates a FileDataSource from the specified path name.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Method Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getContentType()</td>
<td>This method returns the MIME type of the data in the form of a string.</td>
</tr>
<tr>
<td>File</td>
<td>getFile()</td>
<td>Return the File object that corresponds to this FileDataSource.</td>
</tr>
<tr>
<td>InputStream</td>
<td>getInputStream()</td>
<td>This method will return an InputStream representing the data and will throw an IOException if it cannot do so.</td>
</tr>
<tr>
<td>String</td>
<td>getName()</td>
<td>Return the name of this object.</td>
</tr>
<tr>
<td>OutputStream</td>
<td>getOutputStream()</td>
<td>This method will return an OutputStream representing the data and will throw an IOException if it cannot do so.</td>
</tr>
<tr>
<td>void</td>
<td>setFileTypeMap(FileTypeMap map)</td>
<td>Set the FileTypeMap to use with this FileDataSource</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detail

public FileDataSource(java.io.File file)

File DataSource

file

FileDataSource

public FileDataSource(File file)

Creates a FileDataSource from a File object. Note: The file will not actually be opened until a method is called that requires the file to be opened.

Parameters:
file - the file

FileDataSource

public FileDataSource(String name)

FileDataSource

name

FileDataSource

public FileDataSource(String name)

Creates a FileDataSource from the specified path name. Note: The file will not actually be opened until a method is called that requires the file to be opened.

Parameters:
name - the system-dependent file name.
public java.io.InputStream getInputStream() throws java.io.IOException
InputStream IOException InputStream

public InputStream getInputStream() throws IOException

This method will return an InputStream representing the data and will throw an IOException if it cannot do so. This method will return a new instance of InputStream with each invocation.

Specified by:
getInputStream in interface DataSource

Returns:
an InputStream

 Throws:
IOException

public java.io.OutputStream getOutputStream() throws java.io.IOException
OutputStream IOException OutputStream

public OutputStream getOutputStream() throws IOException

This method will return an OutputStream representing the data and will throw an IOException if it cannot do so. This method will return a new instance of OutputStream with each invocation.
public String getContentType()

public String getName()
getName

public String getName()

    Return the name of this object. The FileDataSource will return the file name of the object.

    Specified by: getName in interface DataSource

    Returns: the name of the object.

    See Also: DataSource

getFile

public java.io.File getFile()

    Return the File object that corresponds to this FileDataSource.

    Returns: the File object for the file represented by this object.

setFileTypeMap

public void setFileTypeMap(FileTypeMap map)

    Set the type map for this FileDataSource.
public void setFileTypeMap(FileTypeMap map)

Set the FileTypeMap to use with this FileDataSource

Parameters:
map - The FileTypeMap for this object.
javax.activation Class FileTypeMap

java.lang.Object
  \--- javax.activation.FileTypeMap

Direct Known Subclasses:
  MimetypesFileTypeMap

public abstract class FileTypeMap extends Object

Extended by: MimetypesFileTypeMap

FileTypesMap getContentType File FileTypeMap
MimetypesFileTypeMap FileDataSource FileTypeMap
MimetypesFileTypeMap

See also
javax.activation.FileTypeMap,
javax.activation.FileDataSource,
javax.activation.MimetypesFileTypeMap

The FileTypeMap is an abstract class that provides a data typing interface for files. Implementations of this class will implement the
getContentType methods which will derive a content type from a file
name or a File object. FileTypeMaps could use any scheme to determine
the data type, from examining the file extension of a file (like the
MimetypesFileTypeMap) to opening the file and trying to derive its type
from the contents of the file. The FileDataSource class uses the default
FileTypeMap (a MimetypesFileTypeMap unless changed) to determine
the content type of files.

See Also:
FileTypesMap, FileDataSource, MimetypesFileTypeMap

Constructor Summary


### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract <code>String getContentType(File file)</code></td>
<td>Return the type of the file object.</td>
</tr>
<tr>
<td>abstract <code>String getContentType(String filename)</code></td>
<td>Return the type of the file passed in.</td>
</tr>
<tr>
<td>static <code>FileTypeMap getDefaultFileTypeMap()</code></td>
<td>Return the default <code>FileTypeMap</code> for the system.</td>
</tr>
<tr>
<td>static <code>void setDefaultFileTypeMap(FileTypeMap map)</code></td>
<td>Sets the default <code>FileTypeMap</code> for the system.</td>
</tr>
</tbody>
</table>

### Constructor Detail

public `FileTypeMap()`

The default constructor.

### Method Detail

abstract public `String getContentType(java.io.File file)`
MIME

public abstract String getContentType(File file)

Return the type of the file object. This method should always return a valid MIME type.

Parameters:
  file - A file to be typed.

Returns:
The content type.

abstract public String getContentType(String filename)
MIME

public abstract String getContentType(String filename)

Return the type of the file passed in. This method should always return a valid MIME type.

Parameters:
  filename - the pathname of the file.

Returns:
The content type.

public static void setDefaultFileTypeMap(FileTypeMap map)
**File**

**TypeMap**

**get**

**Default**

**File**

**TypeMap**

\[
\text{map} \quad \text{FileTypeMap}
\]

**Throws**

SecurityException:

**setDefault**

**FileTypeMap**

public static void **setDefault**

**FileTypeMap**

(FileTypeMap map)

Sets the default FileTypeMap for the system. This instance will be returned to callers of getDefaultFileTypeMap.

Parameters:

map - The FileTypeMap.

Throws:

SecurityException - if the caller doesn't have permission to change the default

---

public static **FileTypeMap**

get**Default**FileTypeMap()

**FileTypeMap**

set**Default**FileTypeMap

MimetypesFileTypeMap

return

See also

**setDefault**FileTypeMap

---

**get**

**Default**

**File**

**TypeMap**

public static **FileTypeMap**

get**Default**FileTypeMap()

Return the default FileTypeMap for the system. If setDefaultFileTypeMap was called, return that instance, otherwise return an instance of MimetypesFileTypeMap.

Returns:

The default FileTypeMap

See Also:

setDefaultFileTypeMap(javax.activation.FileTypeMap)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
A filter is an object that performs filtering tasks on either the request to a resource (a servlet or static content), or on the response from a resource, or both.

Filters perform filtering in the `doFilter` method. Every Filter has access to a FilterConfig object from which it can obtain its initialization parameters, a reference to the ServletContext which it can use, for example, to load resources needed for filtering tasks.

Filters are configured in the deployment descriptor of a web application.
Examples that have been identified for this design are
1) Authentication Filters
2) Logging and Auditing Filters
3) Image conversion Filters
4) Data compression Filters
5) Encryption Filters
6) Tokenizing Filters
7) Filters that trigger resource access events
8) XSL/T filters
9) Mime-type chain Filter

Since:
Servlet 2.3

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>voiddestroy()</strong></td>
</tr>
<tr>
<td><strong>voiddoFilter(ServletRequest request, ServletResponse response, FilterChain chain)</strong></td>
</tr>
<tr>
<td><strong>voidinit(FilterConfig filterConfig)</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>public void init(FilterConfig filterConfig) throws ServletException</strong></td>
</tr>
<tr>
<td>Web servlet init init</td>
</tr>
</tbody>
</table>
void init(FilterConfig filterConfig) throws ServletException

Called by the web container to indicate to a filter that it is being placed into service. The servlet container calls the init method exactly once after instantiating the filter. The init method must complete successfully before the filter is asked to do any filtering work.

The web container cannot place the filter into service if the init method either
1. Throws a ServletException
2. Does not return within a time period defined by the web container

Throws:
ServletException

public void doFilter(HttpServletRequest request, HttpServletResponse response, FilterChain chain) throws java.io.IOException, ServletException, ServletException
/ Filter

- 1.
  2.
  3.
4. a) FilterChain (chain.doFilter())
4. b) /
5.
doFilter

```java
doFilter(ServletRequest request, ServletResponse response, FilterChain chain)
throws IOException, ServletException
```

The *doFilter* method of the Filter is called by the container each time a request/response pair is passed through the chain due to a client request for a resource at the end of the chain. The FilterChain passed in to this method allows the Filter to pass on the request and response to the next entity in the chain.

A typical implementation of this method would follow the following pattern:-
1. Examine the request
2. Optionally wrap the request object with a custom implementation to filter content or headers for input filtering
3. Optionally wrap the response object with a custom implementation to filter content or headers for output filtering
4. a) *Either* invoke the next entity in the chain using the FilterChain object (chain.doFilter()),
4. b) *or* not pass on the request/response pair to the next entity in the filter chain to block the request processing
5. Directly set headers on the response after invocation of the next entity in the filter chain.

`Throws:`
- IOException
- ServletException

```
public void destroy()
```

Web doFilter Web doFilter
void **dest**roy()

Called by the web container to indicate to a filter that it is being taken out of service. This method is only called once all threads within the filter's doFilter method have exited or after a timeout period has passed. After the web container calls this method, it will not call the doFilter method again on this instance of the filter.

This method gives the filter an opportunity to clean up any resources that are being held (for example, memory, file handles, threads) and make sure that any persistent state is synchronized with the filter's current state in memory.
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAME: NO FRAMES</td>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
javax.servlet Interface FilterChain

public interface FilterChain

FilterChain servlet FilterChain

since Servlet 2.3

See also javax.servlet.Filter

A FilterChain is an object provided by the servlet container to the developer giving a view into the invocation chain of a filtered request for a resource. Filters use the FilterChain to invoke the next filter in the chain, or if the calling filter is the last filter in the chain, to invoke the resource at the end of the chain.

Since: Servlet 2.3

See Also: Filter

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void doFilter(ServletRequest request, ServletResponse response)</td>
<td>Causes the next filter in the chain to be invoked, or if the calling filter is the last filter in the chain, causes the resource at the end of the chain to be invoked.</td>
</tr>
</tbody>
</table>

Method Detail

public void doFilter(ServletRequest request, ServletResponse response) throws java.io.IOException,
ServletException

request
response
since 2.3

doFilter

void doFilter(ServletRequest request, ServletResponse response)
throws IOException, ServletException

Causes the next filter in the chain to be invoked, or if the calling filter is the last filter in the chain, causes the resource at the end of the chain to be invoked.

Parameters:
request - the request to pass along the chain.
response - the response to pass along the chain.

Throws:
IOException
ServletException

Since:
2.3
javax.servlet Interface FilterConfig

public interface FilterConfig

since Servlet 2.3

See also javax.servlet.Filter

A filter configuration object used by a servlet container to pass information to a filter during initialization.

Since:
Servlet 2.3
See Also:
Filter

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getFilterName()</code></td>
<td>Returns the filter-name of this filter as defined in the deployment descriptor.</td>
</tr>
<tr>
<td><code>String getInitParameter(String name)</code></td>
<td>Returns a string containing the value of the named initialization parameter, or null if the parameter does not exist.</td>
</tr>
<tr>
<td><code>Enumeration getInitParameterNames()</code></td>
<td>Returns the names of the filter's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the filter has no initialization parameters.</td>
</tr>
<tr>
<td><code>ServletContext getServletContext()</code></td>
<td>Returns a reference to the ServletContext in which</td>
</tr>
</tbody>
</table>
Method Detail

public String getFilterName()

getFilterName

String getFilterName()

Returns the filter-name of this filter as defined in the deployment descriptor.

public ServletContext getServletContext()

ServletContext return ServletContext servlet

See also javax.servlet.ServletContext

getServletContext

ServletContext getServletContext()

Returns a reference to the ServletContext in which the caller is executing.

Returns: a ServletContext object, used by the caller to interact with its servlet container

See Also: ServletContext
public String getInitParameter(String name)
    String null

    name String
    return String

getInitParameter

String getInitParameter(String name)

    Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist.

Parameters:
    name - a String specifying the name of the initialization parameter

Returns:
    a String containing the value of the initialization parameter

public java.util Enumeration<E> getInitParameterNames()
    String Enumeration Enumeration
    return String Enumeration

getInitParameterNames

Enumeration getInitParameterNames()

    Returns the names of the filter's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the filter has no initialization parameters.

Returns:
    an Enumeration of String objects containing the names of the filter's initialization parameters
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  **Class FinderException**

**java.lang.Object**  
  **java.lang.Throwable**  
  **java.lang.Exception**  
  **javax.ejb.FinderException**

**All Implemented Interfaces:**  
  **Serializable**

**Direct Known Subclasses:**  
  **ObjectNotFoundException**

---

```
public class FinderException
extends Exception

Extends: Throwable > Exception
Extended by: ObjectNotFoundException
```

Bean home findMETHOD(...) throws FinderException

EJB

The FinderException exception must be included in the throws clause of every findMETHOD(...) method of an entity Bean's home interface.

The exception is used as a standard application-level exception to report a failure to find the requested EJB object(s).

**See Also:**  
  **Serialized Form**

---

**Constructor Summary**
FinderException()

Constructs an FinderException with no detail message.

FinderException(String message)

Constructs an FinderException with the specified detail message.

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public FinderException()
FinderException

FinderException

public FinderException()

Constructs an FinderException with no detail message.

public FinderException(String message)
FinderException

FinderException
public FinderException(String message)

Constructs an FinderException with the specified detail message.
javax.xml.registry Class FindException

java.lang.Object
   └ java.lang.Throwable
      └ java.lang.Exception
         └ javax.xml.registry.JAXRException
            └ javax.xml.registry.RegistryException
               └ javax.xml.registry.FindException

All Implemented Interfaces:
    Serializable, JAXRResponse

public class FindException
extends RegistryException

Extends: Throwable > Exception > JAXRException > RegistryException

find RegistryException

A RegistryException that occurs during a find action.

Author:
    Farrukh S. Najmi

See Also:
    Serialized Form

Field Summary

| Fields inherited from class javax.xml.registry.JAXRException |
| cause |

| Fields inherited from interface javax.xml.registry.JAXRResponse |
| STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING |
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FindException()</strong></td>
<td>Constructs a JAXRException object with no reason or embedded Throwable.</td>
</tr>
<tr>
<td><strong>FindException(String reason)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><strong>FindException(String reason, Throwable cause)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.</td>
</tr>
<tr>
<td><strong>FindException(Throwable cause)</strong></td>
<td>Constructs a JAXRException object initialized with the given Throwable object.</td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class javax.xml.registry.RegistryException
- [getErrorObjectKey](#), [setErrorObjectKey](#)

### Methods inherited from class javax.xml.registry.JAXRException
- [getCause](#), [getMessage](#), [getRequestId](#), [getStatus](#), [initCause](#), [isAvailable](#)

### Methods inherited from class java.lang.ThrowException
- [fillInStackTrace](#), [getLocalizedMessage](#), [getStackTrace](#), [printStackTrace](#), [printStackTrace](#), [printStackTrace](#), [setStackTrace](#), [toString](#)

### Methods inherited from class java.lang.Object
- [clone](#), [equals](#), [finalize](#), [getClass](#), [hashCode](#), [notify](#), [notifyAll](#), [wait](#), [wait](#), [wait](#)
public FindException()
    Throwable JAXRException

FindException

public FindException()
    Constructs a JAXRException object with no reason or embedded Throwable.

public FindException(String reason)
    JAXRException String reason

FindException

public FindException(String reason)
    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

    Parameters:
        reason - a description of what caused the exception

public FindException(String reason, Throwable cause)
    JAXRException String Throwable
    Throwable
        reason
        cause
FindException

public FindException(String reason, Throwable cause)

Constructs a JAXRException object with the given string as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
  reason - a description of what caused the exception
  cause - a Throwable object that is to be embedded in this JAXRException object

public FindException(Throwable cause)

    Throwable JAXRException
    cause Exception Throwable

FindException

public FindException(Throwable cause)

Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
  cause - the Throwable that caused this Exception

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY:NESTED | FIELD | CONSTR | METHOD
FRAMES  NO FRAMES
DETAIL:FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.registry Interface FindQualifier

public interface FindQualifier

FindQualifier find JAXR JAXR

See also javax.xml.registry.BusinessQueryManager

FindQualifier provides various constants that identify options that effect find method behavior. A JAXR provider may silently ignore any qualifiers marked as optional. A JAXR provider must support qualifiers not marked optional.

See Also: BusinessQueryManager

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String AND_ALL_KEYS</td>
</tr>
<tr>
<td>static String CASE_SENSITIVE_MATCH</td>
</tr>
<tr>
<td>static String COMBINE_CLASSIFICATIONS</td>
</tr>
<tr>
<td>Maps to UDDI combineCategoryBags.</td>
</tr>
<tr>
<td>static String EXACT_NAME_MATCH</td>
</tr>
<tr>
<td>static String OR_ALL_KEYS</td>
</tr>
<tr>
<td>static String OR_LIKE_KEYS</td>
</tr>
<tr>
<td>static String SERVICE_SUBSET</td>
</tr>
<tr>
<td>Maps to UDDI serviceSubset.</td>
</tr>
<tr>
<td>static String SORT_BY_DATE_ASC</td>
</tr>
</tbody>
</table>
### Field Detail

**EXACT_NAME_MATCH**

`static final String EXACT_NAME_MATCH`

See Also:

- [Constant Field Values](#)

---

**CASE_SENSITIVE_MATCH**

`static final String CASE_SENSITIVE_MATCH`

See Also:

- [Constant Field Values](#)

---

**SORT_BY_NAME_ASC**
static final String SORT_BY_NAME_ASC

See Also:
    Constant Field Values

SORT_BY_NAME_DESC
static final String SORT_BY_NAME_DESC

See Also:
    Constant Field Values

SORT_BY_DATE_ASC
static final String SORT_BY_DATE_ASC

See Also:
    Constant Field Values

SORT_BY_DATE_DESC
static final String SORT_BY_DATE_DESC

See Also:
    Constant Field Values

ORLIKEKEYS
static final String OR_LIKE_KEYS

See Also:
Constant Field Values

---

OR_ALL_KEYS

static final String OR_ALL_KEYS

See Also:
Constant Field Values

---

COMBINE_CLASSIFICATIONS

static final String COMBINE_CLASSIFICATIONS

Maps to UDDI combineCategoryBags. Required for JAXR UDDI providers, optional for all others.

See Also:
Constant Field Values

---

SERVICE_SUBSET

static final String SERVICE_SUBSET

Maps to UDDI serviceSubset. Required for JAXR UDDI providers, optional for all others.

See Also:
Constant Field Values
AND_ALL_KEYS

static final String AND_ALL_KEYS

See Also:
Constant Field Values

SOUNDEX

static final String SOUNDEX

Optional qualifier that allows matching strings by their sounds.

See Also:
Constant Field Values
javax.mail **Class Flags**

```
java.lang.Object
  \_ javax.mail.Flags

All Implemented Interfaces:
    Serializable, Cloneable
```

defined public class **Flags** extends **Object**
implements **Cloneable, Serializable**

**Implements:** Cloneable, java.io.Serializable  
**Inner classes:** **Flags.Flag**

```
Flags  Message Flags
      Flags.Flag  String

Folder

Flags  Flags
      JavaMail API
```

```java
Message m = folder.getMessage(1);
m.setFlag(Flags.Flag.DELETED, true); // set the DELETED flag

// Check if DELETED flag is set of this message
if (m.isSet(Flags.Flag.DELETED))
    System.out.println("DELETED message");

// Examine ALL system flags for this message
Flags flags = m.getFlags();
Flags.Flag[] sf = flags.getSystemFlags();
for (int i = 0; i < sf.length; i++) {
    if (sf[i] == Flags.Flag.DELETED)
        System.out.println("DELETED message");
```

else if (sf[i] == Flags.Flag.SEEN)
System.out.println("SEEN message");
......
......
}

See also

getPermanentFlags

The Flags class represents the set of flags on a Message. Flags are composed of predefined system flags, and user defined flags.

A System flag is represented by the Flags.Flag inner class. A User defined flag is represented as a String. User flags are case-independent.

A set of standard system flags are predefined. Most folder implementations are expected to support these flags. Some implementations may also support arbitrary user-defined flags. The getPermanentFlags method on a Folder returns a Flags object that holds all the flags that are supported by that folder implementation.

A Flags object is serializable so that (for example) the use of Flags objects in search terms can be serialized along with the search terms.

Warning: Serialized objects of this class may not be compatible with future JavaMail API releases. The current serialization support is appropriate for short term storage.

The below code sample illustrates how to set, examine and get the flags for a message.

```java
Message m = folder.getMessage(1);
m.setFlag(Flags.Flag.DELETED, true); // set the DELETED flag

// Check if DELETED flag is set of this message
if (m.isSet(Flags.Flag.DELETED))
    System.out.println("DELETED message");

// Examine ALL system flags for this message
Flags flags = m.getFlags();
Flags.Flag[] sf = flags.getSystemFlags();
```
for (int i = 0; i < sf.length; i++) {
    if (sf[i] == Flags.Flag.DELETED) {
        System.out.println("DELETED message");
    } else if (sf[i] == Flags.Flag.SEEN) {
        System.out.println("SEEN message");
    }
}

Author:
John Mani, Bill Shannon

See Also:
Folder.getPermanentFlags(), Serialized Form

## Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>Flags.Flag</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This inner class represents an individual system flag.</td>
</tr>
</tbody>
</table>

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flags()</td>
<td>Construct an empty Flags object.</td>
</tr>
<tr>
<td>Flags(Flags.Flag flag)</td>
<td>Construct a Flags object initialized with the given system flag.</td>
</tr>
<tr>
<td>Flags(Flags flags)</td>
<td>Construct a Flags object initialized with the given flags.</td>
</tr>
<tr>
<td>Flags(String flag)</td>
<td>Construct a Flags object initialized with the given user flag.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void add(Flags.Flag flag)</td>
<td>Add the specified system flag to this Flags object.</td>
</tr>
<tr>
<td>void add(Flags f)</td>
<td>Add all the flags in the given Flags object to this Flags object.</td>
</tr>
<tr>
<td>Method</td>
<td>Signature</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>add</strong></td>
<td><code>String flag</code></td>
</tr>
<tr>
<td><strong>clone</strong></td>
<td></td>
</tr>
<tr>
<td><strong>contains</strong></td>
<td><code>Flags.Flag flag</code></td>
</tr>
<tr>
<td><strong>contains</strong></td>
<td><code>Flags f</code></td>
</tr>
<tr>
<td><strong>contains</strong></td>
<td><code>String flag</code></td>
</tr>
<tr>
<td><strong>equals</strong></td>
<td><code>Object obj</code></td>
</tr>
<tr>
<td><strong>getSystemFlags</strong></td>
<td></td>
</tr>
<tr>
<td><strong>getUserFlags</strong></td>
<td></td>
</tr>
<tr>
<td><strong>hashCode</strong></td>
<td></td>
</tr>
<tr>
<td><strong>remove</strong></td>
<td><code>Flags.Flag flag</code></td>
</tr>
<tr>
<td><strong>remove</strong></td>
<td><code>Flags f</code></td>
</tr>
<tr>
<td><strong>remove</strong></td>
<td><code>String flag</code></td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

- `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public Flags()
Flags

Construct an empty Flags object.

public Flags(Flags flags)
Flags

flags

Construct a Flags object initialized with the given flags.

Parameters:
flags - the flags for initialization

public Flags(Flags.Flag flag)
Flags

flag

public Flags(Flags.Flag flag)
Construct a Flags object initialized with the given system flag.

Parameters:
flag - the flag for initialization

public Flags(String flag)

Construct a Flags object initialized with the given user flag.

Parameters:
flag - the flag for initialization

Method Detail

public void add(Flags.Flag flag)

Add the specified system flag to this Flags object.

Parameters:
flag - the flag to add
public void add(String flag)

Add the specified user flag to this Flags object.

Parameters:
flag - the flag to add

public void add(Flags f)

Add all the flags in the given Flags object to this Flags object.

Parameters:
f - Flags object

public void remove(Flags.Flag flag)

remove
public void remove(Flags.Flag flag)

    Remove the specified system flag from this Flags object.

    **Parameters:**
    flag - the flag to be removed

public void remove(String flag)

    Remove the specified user flag from this Flags object.

    **Parameters:**
    flag - the flag to be removed

public void remove(Flags f)

    Remove all flags in the given Flags object from this Flags object.

    **Parameters:**
    f - the flag to be removed
public boolean contains(Flags.Flag flag)
Flags
    return true false
contains

public boolean contains(Flags.Flag flag)
    Check whether the specified system flag is present in this Flags object.
    
    Returns:
    true of the given flag is present, otherwise false.

public boolean contains(String flag)
Flags
    return true false
contains

public boolean contains(String flag)
    Check whether the specified user flag is present in this Flags object.
    
    Returns:
    true of the given flag is present, otherwise false.

public boolean contains(Flags f)
Flags Flags
    return Flags true false
contains
public boolean contains(Flags f)

Check whether all the flags in the specified Flags object are present in this Flags object.

Returns:
true if all flags in the given Flags object are present, otherwise false.

public boolean equals(Object obj)

Check whether the two Flags objects are equal.

Overrides:
equals in class Object

Returns:
ture if they're equal

public int hashCode()

Compute a hash code for this Flags object.
Overrides:

- `hashCode` in class `Object`

Returns:

- the hash code

---

```java
public Flags.Flag[] getSystemFlags()
Flags 0
    return Flags.Flag
```

**getSystemFlags**

```java
public Flags.Flag[] getSystemFlags()
```

Return all the system flags in this Flags object. Returns an array of size zero if no flags are set.

**Returns:**

- array of Flags.Flag objects representing system flags

---

```java
public String[] getUserFlags()
Flags 0
    return String String
```

**getUserFlags**

```java
public String[] getUserFlags()
```

Return all the user flags in this Flags object. Returns an array of size zero if no flags are set.

**Returns:**

- array of Strings, each String represents a flag.
public Object clone()  
Flags

clone

public Object clone()

Returns a clone of this Flags object.

Overrides:
   clone in class Object
javax.mail Class Flags.Flag

java.lang.Object
  - javax.mail.Flags.Flag

Enclosing class:
  Flags

public static final class Flags.Flag
extends Object

Contained within: Flags

This inner class represents an individual system flag. A set of standard system flag objects are predefined here.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static Flags.Flag ANSWERED</td>
</tr>
<tr>
<td>static Flags.Flag DELETED</td>
</tr>
<tr>
<td>static Flags.Flag DRAFT</td>
</tr>
<tr>
<td>static Flags.Flag FLAGGED</td>
</tr>
<tr>
<td>static Flags.Flag RECENT</td>
</tr>
<tr>
<td>static Flags.Flag SEEN</td>
</tr>
<tr>
<td>static Flags.Flag USER</td>
</tr>
</tbody>
</table>
A special flag that indicates that this folder supports user defined flags.

| Method Summary |

| Methods inherited from class `java.lang.Object` |

| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

| Field Detail |

**ANSWERED**

```java
public static final Flags.Flag ANSWERED
```

This message has been answered. This flag is set by clients to indicate that this message has been answered to.

**DELETED**

```java
public static final Flags.Flag DELETED
```

This message is marked deleted. Clients set this flag to mark a message as deleted. The expunge operation on a folder removes all messages in that folder that are marked for deletion.

**DRAFT**
public static final Flags.Flag DRAFT

This message is a draft. This flag is set by clients to indicate that the message is a draft message.

---

FLAGGED

public static final Flags.Flag FLAGGED

This message is flagged. No semantic is defined for this flag. Clients alter this flag.

---

RECENT

public static final Flags.Flag RECENT

This message is recent. Folder implementations set this flag to indicate that this message is new to this folder, that is, it has arrived since the last time this folder was opened.

Clients cannot alter this flag.

---

SEEN

public static final Flags.Flag SEEN

This message is seen. This flag is implicitly set by the implementation when the this Message's content is returned to the client in some form. The getInputStream and getContent methods on Message cause this flag to be set.
Clients can alter this flag.

USER

public static final Flags.Flag USER

A special flag that indicates that this folder supports user defined flags.

The implementation sets this flag. Clients cannot alter this flag but can use it to determine if a folder supports user defined flags by using folder.getPermanentFlags().contains(Flags.Flag.USER).
javax.mail.search Class FlagTerm

java.lang.Object
   ↓ javax.mail.search.SearchTerm
      ↓ javax.mail.search.FlagTerm

All Implemented Interfaces:
   Serializable

public final class FlagTerm extends SearchTerm

Extends: SearchTerm

Message Flag

This class implements comparisons for Message Flags.

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected Flags flags</th>
<th>Flags object containing the flags to test.</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected boolean set</td>
<td>Indicates whether to test for the presence or absence of the specified Flag.</td>
</tr>
</tbody>
</table>

Constructor Summary

FlagTerm(Flags flags, boolean set)
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>Flags getFlags()</td>
<td>Return the Flags to test.</td>
</tr>
<tr>
<td>boolean getTestSet()</td>
<td>Return true if testing whether the flags are set.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean match(Message msg)</td>
<td>The comparison method.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
clone, finalize, getClass, notify, notifyAll, toString, wait, wait

**Field Detail**

protected boolean set

Indicates whether to test for the presence or absence of the specified Flag. If true, then test whether all the specified flags are present, else test whether all the specified flags are absent.
protected Flags flags

Flags object containing the flags to test.

**Constructor Detail**

public FlagTerm(Flags flags, boolean set)

Set FlagTerm

**FlagTerm**

public FlagTerm(Flags flags, boolean set)

Constructor.

**Parameters:**
flags - Flags object containing the flags to check for
set - the flag setting to check for

**Method Detail**

public Flags getFlags()

Flag getFlags

**getFlags**

public Flags getFlags()

Return the Flags to test.
public boolean getTestSet()
true

getTestSet

public boolean getTestSet()

    Return true if testing whether the flags are set.

----------------------------------------

public boolean match(Message msg)

    msg Message
    return true false

match

public boolean match(Message msg)

    The comparison method.

    Specified by:
    match in class SearchTerm

    Parameters:
    msg - The flag comparison is applied to this Message

    Returns:
    true if the comparison succeeds, otherwise false.

----------------------------------------

public boolean equals(Object obj)

equals
public boolean equals(Object obj)

Equality comparison.

Overrides:
equals in class Object

public int hashCode()
hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

Overrides:
hashCode in class Object

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Class **FloatConverter**

**All Implemented Interfaces:**

- Converter

```java
public class FloatConverter
    extends Object
    implements Converter

Implements: Converter
```

**Converter** implementation for java.lang.Float (and float primitive) values.

### Field Summary

<table>
<thead>
<tr>
<th>Static String</th>
<th>CONVERTER_ID</th>
<th>The standard converter id for this converter.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static String</td>
<td>FLOAT_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion to Float fails.</td>
</tr>
<tr>
<td>Static String</td>
<td>STRING_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion of the Float Value to String fails.</td>
</tr>
</tbody>
</table>

### Constructor Summary

- `FloatConverter()`
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified <code>UIComponent</code>, into a model data object that is appropriate for being stored during the <code>Apply Request Values</code> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified <code>UIComponent</code>, into a String that is suitable for being included in the response generated during the <code>Render Response</code> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

**Methods inherited from class `java.lang.Object`**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`  

**Field Detail**

**CONVERTER_ID**

```
public static final String CONVERTER_ID
```

The standard converter id for this converter.

**See Also:**

`Constant Field Values`
FLOAT_ID

public static final String FLOAT_ID

The message identifier of the FacesMessage to be created if the conversion to Float fails. The message format string for this message may optionally include the following placeholders:

- {0} replaced by the unconverted value.
- {1} replaced by an example value.
- {2} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Float value to String fails. The message format string for this message may optionally include the following placeholders:

- {0} replaced by the unconverted value.
- {1} replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail

cpy{FloatConverter()}
FloatConverter

public FloatConverter()

Method Detail

public Object getAsObject(FacesContext context, UIComponent component, String value)

Throws ConverterException: NullPointerException

Throws NullPointerException: NullPointerException

getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

Description copied from interface: Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

Specified by:
getAsObject in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - String value to be converted (may be null)

Returns:
null if the value to convert is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws ConverterException: NullPointerException

Throws NullPointerException: NullPointerException

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
javax.xml.rpc.holders  **Class FloatHolder**

**java.lang.Object**  
---  
 javax.xml.rpc.holders.FloatHolder

**All Implemented Interfaces:**  
- **Holder**

```java
public final class FloatHolder
    extends Object
    implements Holder

Implements: Holder
```

---

### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>float</td>
<td>value</td>
</tr>
</tbody>
</table>

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>FloatHolder()</td>
</tr>
<tr>
<td>FloatHolder(float myfloat)</td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object</td>
</tr>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>
public float value

Constructor Detail

public FloatHolder()

FloatHolder

public FloatHolder()

public FloatHolder(float myfloat)

FloatHolder

public FloatHolder(float myfloat)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.rpc.holders  Class FloatWrapperHolder

java.lang.Object
  ↓ javax.xml.rpc.holders.FloatWrapperHolder

All Implemented Interfaces:
   Holder

public final class FloatWrapperHolder
  extends Object
  implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Float</td>
<td></td>
</tr>
</tbody>
</table>

Constructor Summary

| FloatWrapperHolder()          |
| FloatWrapperHolder(Float myfloat) |

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait,
value

public Float value

Constructor Detail

public FloatWrapperHolder()

FloatWrapperHolder

public FloatWrapperHolder()

public FloatWrapperHolder(Float myfloat)

FloatWrapperHolder

public FloatWrapperHolder(Float myfloat)
PS:
Flush mode setting.

When queries are executed within a transaction, if `FlushModeType.AUTO` is set on the `Query` object, or if the flush mode setting for the persistence context is `AUTO` (the default) and a flush mode setting has not been specified for the `Query` object, the persistence provider is responsible for ensuring that all updates to the state of all entities in the persistence context which could potentially affect the result of the query are visible to the processing of the query. The persistence provider implementation may achieve this by flushing those entities to the database or by some other means. If `FlushModeType.COMMIT` is set, the effect of updates made to entities in the persistence context upon queries is unspecified.
If there is no transaction active, the persistence provider must not flush to
the database.

Since:
Java Persistence 1.0

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>(Default) Flushing to occur at query execution</td>
</tr>
<tr>
<td>COMMIT</td>
<td>Flushing must occur only at transaction commit</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Enum**

`clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

Methods inherited from class java.lang.**Object**

`finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Enum Constant Detail

COMMIT
public static final FlushModeType COMMIT

Flushing must occur only at transaction commit

AUTO

public static final FlushModeType AUTO

(Default) Flushing to occur at query execution

Method Detail

final public static FlushModeType[] values()

values

public static final FlushModeType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

for(FlushModeType c : FlushModeType.values())
    System.out.println(c);

Returns:
    an array containing the constants of this enum type, in the order they're declared

public static FlushModeType valueOf(String name)

valueOf
public static FlushModeType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
   name - the name of the enum constant to be returned.

Returns:
   the enum constant with the specified name

Throws:
   IllegalArgumentException - if this enum type has no constant with the specified name

Overview Package Tree Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.mail Class Folder

java.lang.Object
   ▼ javax.mail.Folder

public abstract class Folder
extends Object

Folder Folder
Folder Message Folder Store Folder
Message Folder Folder

Store
   INBOX Store

INBOX "" $

Folder 'open' Folder
Folder

Folder StoreFolder Session
   getFolder Folder
list listSubscribed Store Folder
getFolder
Folder list listSubscribed

Folder Folder Message
Message Folder
Message Message

Folder is an abstract class that represents a folder for mail messages. Subclasses implement protocol specific Folders.
Folders can contain Messages, other Folders or both, thus providing a tree-like hierarchy rooted at the Store's default folder. (Note that some Folder implementations may not allow both Messages and other Folders in the same Folder).

The interpretation of folder names is implementation dependent. The different levels of hierarchy in a folder's full name are separated from each other by the hierarchy delimiter character.

The case-insensitive full folder name (that is, the full name relative to the default folder for a Store) **INBOX** is reserved to mean the "primary folder for this user on this server". Not all Stores will provide an INBOX folder, and not all users will have an INBOX folder at all times. The name **INBOX** is reserved to refer to this folder, when it exists, in Stores that provide it.

A Folder object obtained from a Store need not actually exist in the backend store. The `exists` method tests whether the folder exists or not. The `create` method creates a Folder.

A Folder is initially in the closed state. Certain methods are valid in this state; the documentation for those methods note this. A Folder is opened by calling its `open` method. All Folder methods, except `open`, `delete` and `renameTo`, are valid in this state.

The only way to get a Folder is by invoking the `getFolder` method on Store, Folder, or Session, or by invoking the `list` or `listSubscribed` methods on Folder. Folder objects returned by the above methods are not cached by the Store. Thus, invoking the `getFolder` method with the same folder name multiple times will return distinct Folder objects. Likewise for the `list` and `listSubscribed` methods.

The Message objects within the Folder are cached by the Folder. Thus, invoking `getMessage(msgno)` on the same message number multiple times will return the same Message object, until an expunge is done on this Folder.

Note that a Message's message number can change within a session if the containing Folder is expunged using the `expunge` method. Clients that use message numbers as references to messages should be aware
of this and should be prepared to deal with situation (probably by flushing out existing message number references and reloading them). Because of this complexity, it is better for clients to use Message objects as references to messages, rather than message numbers. Expunged Message objects still have to be pruned, but other Message objects in that folder are not affected by the expunge.

Author:
John Mani, Bill Shannon

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
</table>
| **static int** HOLDS_FOLDERS
  This folder can contain other folders |
| **static int** HOLDS_MESSAGES
  This folder can contain messages |
| **protected int** mode
  The open mode of this folder. |
| **static int** READ_ONLY
  The Folder is read only. |
| **static int** READ_WRITE
  The state and contents of this folder can be modified. |
| **protected Store** store
  The parent store. |

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
</table>
| **protected Folder(Store store)**
  Constructor that takes a Store object. |

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| **void addConnectionListener(ConnectionListener l)**
  Add a listener for Connection events on this Folder. |
| **void addFolderListener(FolderListener l)** |
Add a listener for Folder events on this Folder.

```java
void addMessageChangedListener(MessageChangedListener l)
Add a listener for MessageChanged events on this Folder.
```

Add a listener for MessageCount events on this Folder.

```java
void addMessageCountListener(MessageCountListener l)
```

Append given Messages to this folder.

```java
abstract void appendMessages(Message[] msgs)
```

Append given Messages to this folder.

```java
abstract void close(boolean expunge)
Close this Folder.
```

Copy the specified Messages from this Folder into another Folder.

```java
void copyMessages(Message[] msgs, Folder folder)
```

Close this Folder.

```java
abstract boolean close(boolean recurse)
Delete this Folder.
```

Tests if this folder physically exists on the Store.

```java
abstract boolean exists()
```

Expunge (permanently remove) messages marked DELETED.

```java
abstract boolean expunge()
```

Prefetch the items specified in the FetchProfile for the given Messages.

```java
void fetch(Message[] msgs, FetchProfile fp)
```

Prefetch the items specified in the FetchProfile for the given Messages.

```java
protected void finalize()
```

Get the number of deleted messages in this Folder.

```java
int getDeletedMessageCount()
```

Return the Folder object corresponding to the given name.

```java
abstract Folder getFolder(String name)
```

Returns the full name of this Folder.

```java
abstract String getFullName()
```

Get the number of deleted messages in this Folder.

```java
abstract getMessage(int msgnum)
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getMessage(String)</code></td>
<td>Get the Message object corresponding to the given message number.</td>
</tr>
<tr>
<td><code>getMessageCount()</code></td>
<td>Get total number of messages in this Folder.</td>
</tr>
<tr>
<td><code>getMessages()</code></td>
<td>Get all Message objects from this Folder.</td>
</tr>
<tr>
<td><code>getMessages(int[] msgnums)</code></td>
<td>Get the Message objects for message numbers specified in the array.</td>
</tr>
<tr>
<td><code>getMessages(int start, int end)</code></td>
<td>Get the Message objects for message numbers ranging from start through end, both start and end inclusive.</td>
</tr>
<tr>
<td><code>getMode()</code></td>
<td>Return the open mode of this folder.</td>
</tr>
<tr>
<td><code>getName()</code></td>
<td>Returns the name of this Folder.</td>
</tr>
<tr>
<td><code>getNewMessageCount()</code></td>
<td>Get the number of new messages in this Folder.</td>
</tr>
<tr>
<td><code>getParent()</code></td>
<td>Returns the parent folder of this folder.</td>
</tr>
<tr>
<td><code>getPermanentFlags()</code></td>
<td>Get the permanent flags supported by this Folder.</td>
</tr>
<tr>
<td><code>getSeparator()</code></td>
<td>Return the delimiter character that separates this Folder's pathname from the names of immediate subfolders.</td>
</tr>
<tr>
<td><code>getStore()</code></td>
<td>Returns the Store that owns this Folder object.</td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Returns the type of this Folder, that is, whether this folder can hold messages or subfolders or both.</td>
</tr>
<tr>
<td><code>getUnreadMessageCount()</code></td>
<td>Get the total number of unread messages in this Folder.</td>
</tr>
<tr>
<td><code>getURLName()</code></td>
<td>Return a URLName representing this folder.</td>
</tr>
<tr>
<td><code>hasNewMessages()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>abstract boolean</td>
<td>Returns true if this Folder has new messages since the last time this indication was reset.</td>
</tr>
<tr>
<td>abstract boolean isOpen()</td>
<td>Indicates whether this Folder is in the 'open' state.</td>
</tr>
<tr>
<td>boolean isSubscribed()</td>
<td>Returns true if this Folder is subscribed.</td>
</tr>
<tr>
<td>Folder[] list()</td>
<td>Convenience method that returns the list of folders under this Folder.</td>
</tr>
<tr>
<td>abstract Folder[] list(String pattern)</td>
<td>Returns a list of Folders belonging to this Folder's namespace that match the specified pattern.</td>
</tr>
<tr>
<td>Folder[] listSubscribed()</td>
<td>Convenience method that returns the list of subscribed folders under this Folder.</td>
</tr>
<tr>
<td>Folder[] listSubscribed(String pattern)</td>
<td>Returns a list of subscribed Folders belonging to this Folder's namespace that match the specified pattern.</td>
</tr>
<tr>
<td>protected void notifyConnectionListeners(int type)</td>
<td>Notify all ConnectionListeners.</td>
</tr>
<tr>
<td>protected void notifyFolderListeners(int type)</td>
<td>Notify all FolderListeners registered on this Folder and this folder's Store.</td>
</tr>
<tr>
<td>protected void notifyFolderRenamedListeners(Folder folder)</td>
<td>Notify all FolderListeners registered on this Folder and this folder's Store about the renaming of this folder.</td>
</tr>
<tr>
<td>protected void notifyMessageAddedListeners(Message[] msgs)</td>
<td>Notify all MessageCountListeners about the addition of messages into this folder.</td>
</tr>
<tr>
<td>protected void notifyMessageChangedListeners(int type, Message msg)</td>
<td>Notify all MessageChangedListeners.</td>
</tr>
<tr>
<td>protected void notifyMessageRemovedListeners(boolean removed, Message[] msgs)</td>
<td>Notify all MessageCountListeners about the removal of messages from this Folder.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>abstract void open(int mode)</code></td>
<td>Open this Folder.</td>
</tr>
<tr>
<td><code>void removeConnectionListener(ConnectionListener l)</code></td>
<td>Remove a Connection event listener.</td>
</tr>
<tr>
<td><code>void removeFolderListener(FolderListener l)</code></td>
<td>Remove a Folder event listener.</td>
</tr>
<tr>
<td><code>void removeMessageChangedListener(MessageChangedListener l)</code></td>
<td>Remove a MessageChanged listener.</td>
</tr>
<tr>
<td><code>void removeMessageCountListener(MessageCountListener l)</code></td>
<td>Remove a MessageCount listener.</td>
</tr>
<tr>
<td><code>abstract boolean renameTo(Folder f)</code></td>
<td>Rename this Folder.</td>
</tr>
<tr>
<td><code>Message[] search(SearchTerm term)</code></td>
<td>Search this Folder for messages matching the specified search criterion.</td>
</tr>
<tr>
<td><code>Message[] search(SearchTerm term, Message[] msgs)</code></td>
<td>Search the given array of messages for those that match the specified search criterion.</td>
</tr>
<tr>
<td><code>void setFlags(int[] msgnums, Flags flag, boolean value)</code></td>
<td>Set the specified flags on the messages whose message numbers are in the array.</td>
</tr>
<tr>
<td><code>void setFlags(int start, int end, Flags flag, boolean value)</code></td>
<td>Set the specified flags on the messages numbered from start through end, both start and end inclusive.</td>
</tr>
<tr>
<td><code>void setFlags(Message[] msgs, Flags flag, boolean value)</code></td>
<td>Set the specified flags on the messages specified in the array.</td>
</tr>
<tr>
<td><code>void setSubscribed(boolean subscribe)</code></td>
<td>Subscribe or unsubscribe this Folder.</td>
</tr>
<tr>
<td><code>String toString()</code></td>
<td>override the default toString(), it will return the String from Folder.getFullName() or if that is null, it will use the default toString() behavior.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**
Field Detail

store

protected `Store` store

The parent store.

mode

protected `int` mode

The open mode of this folder. The open mode is `Folder.READ_ONLY`, `Folder.READ_WRITE`, or -1 if not known.

Since:

JavaMail 1.1

HOLDS_MESSAGES

public static final `int` HOLDS_MESSAGES

This folder can contain messages

See Also:

Constant Field Values
HOLDS_FOLDERS

public static final int HOLDS_FOLDERS

This folder can contain other folders

See Also:
Constant Field Values

READ_ONLY

public static final int READ_ONLY

The Folder is read only. The state and contents of this folder cannot be modified.

See Also:
Constant Field Values

READ_WRITE

public static final int READ_WRITE

The state and contents of this folder can be modified.

See Also:
Constant Field Values

Constructor Detail

protected Folder(Store store)
Folder

protected Folder(Store store)

Constructor that takes a Store object.

Parameters:
store - the Store that holds this folder

Method Detail

abstract public String getName()

Folder

Folder

return

getName

public abstract String getName()

Returns the name of this Folder.
This method can be invoked on a closed Folder.

Returns:
name of the Folder

abstract public String getFullName()

Folder Store
Folder

getFullName

```java
public abstract String getFullName()
```

Returns the full name of this Folder. If the folder resides under the root hierarchy of this Store, the returned name is relative to the root. Otherwise an absolute name, starting with the hierarchy delimiter, is returned.

This method can be invoked on a closed Folder.

**Returns:**
full name of the Folder

public **URLName** getURLName() throws **MessagingException**

```java
URLName getURLName() throws MessagingException
```

Return a URLName representing this folder. The returned URLName does *not* include the password used to access the store.

**Returns:**
the URLName representing this folder
Throws: MessagingException
Since: JavaMail 1.1
See Also: URLName

public Store getStore()
Folder Store Folder
    return Store

getStore

public Store getStore()

Returns the Store that owns this Folder object. This method can be invoked on a closed Folder.

Returns: the Store

abstract public Folder getParent() throws MessagingException
Folder null

getParent

public abstract Folder getParent()
    throws MessagingException
Returns the parent folder of this folder. This method can be invoked on a closed Folder. If this folder is the top of a folder hierarchy, this method returns null.

Note that since Folder objects are not cached, invoking this method returns a new distinct Folder object.

**Returns:**
- Parent folder

**Throws:**
- `MessagingException`

---

```java
abstract public boolean exists() throws MessagingException

Store Folder
return true false
Throws MessagingException:
See also create
```

**exists**

```java
public abstract boolean exists()
throws MessagingException
```

Tests if this folder physically exists on the Store. This method can be invoked on a closed Folder.

**Returns:**
- true if the folder exists, otherwise false

**Throws:**
- `MessagingException` - typically if the connection to the server is lost.

**See Also:**
- `create(int)`

---

```java
abstract public Folder[] list(String pattern) throws
```
MessagingException
Folder Folder Folder

Personal/
Finance/
Stocks
Bonus
StockOptions
Jokes

"Personal" list("*")
"Personal" list("%") "Finance" "Jokes"
"Personal" list("Jokes") "Jokes"
"Finance" list("Stock*") "Stocks"
"StockOptions"

Store Folder Folder

Folder

<table>
<thead>
<tr>
<th>pattern</th>
<th>return</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder</td>
<td>Folder</td>
</tr>
</tbody>
</table>

Throws FolderNotFoundException:
Throws MessagingException:
See also listSubscribed

list

public abstract Folder[] list(String pattern) throws MessagingException

Returns a list of Folders belonging to this Folder's namespace that match the specified pattern. Patterns may contain the wildcard characters "%", which matches any character except hierarchy
delimiters, and "*", which matches any character.

As an example, given the folder hierarchy:

```
Personal/
   Finance/
      Stocks
      Bonus
      StockOptions
   Jokes
```

list("*") on "Personal" will return the whole hierarchy.
list("%") on "Personal" will return "Finance" and "Jokes".
list("Jokes") on "Personal" will return "Jokes".
list("Stock*") on "Finance" will return "Stocks" and "StockOptions".

Folder objects are not cached by the Store, so invoking this method on the same pattern multiple times will return that many distinct Folder objects.

This method can be invoked on a closed Folder.

**Parameters:**
- pattern - the match pattern

**Returns:**
- array of matching Folder objects. An empty array is returned if no matching Folders exist.

**Throws:**
- `FolderNotFoundException` - if this folder does not exist.
- `MessagingException`

**See Also:**
- `listSubscribed(java.lang.String)`

---

```java
public Folder[] listSubscribed(String pattern) throws MessagingException
```

Folder Folder Folder
```
   list
```
Folder

```
public Folder[] listSubscribed(String pattern)
    throws MessagingException
```

Returns a list of subscribed Folders belonging to this Folder's namespace that match the specified pattern. If the folder does not support subscription, this method should resolve to `list`. (The default implementation provided here, does just this). The pattern can contain wildcards as for `list`.

Note that, at a given level of the folder hierarchy, a particular folder may not be subscribed, but folders underneath that folder in the folder hierarchy may be subscribed. In order to allow walking the folder hierarchy, such unsubscribed folders may be returned, indicating that a folder lower in the hierarchy is subscribed. The `isSubscribed` method on a folder will tell whether any particular folder is actually subscribed.

Folder objects are not cached by the Store, so invoking this method on the same pattern multiple times will return that many distinct Folder objects.

This method can be invoked on a closed Folder.

**Parameters:**

- `pattern` - the match pattern

**Returns:**
array of matching subscribed Folder objects. An empty array is returned if no matching subscribed folders exist.

**Throws:**
- [FolderNotFoundException](#) - if this folder does not exist.
- [MessagingException](#)

**See Also:**
- [list(java.lang.String)](#)

---

**public Folder[] list() throws MessagingException**

Folder

```
public Folder[] list() throws MessagingException
```

- **Return**
  - Folder Folder
  - Throws [FolderNotFoundException](#): 
    - Throws [MessagingException](#):
  - See also [list](#)

### list

**public Folder[] list()**

```
public Folder[] list() throws MessagingException
```

Convenience method that returns the list of folders under this Folder. This method just calls the list(String pattern) method with "%" as the match pattern. This method can be invoked on a closed Folder.

**Returns:**
- array of Folder objects under this Folder. An empty array is returned if no subfolders exist.

**Throws:**
- [FolderNotFoundException](#) - if this folder does not exist.
- [MessagingException](#)

**See Also:**
- [list(java.lang.String)](#)

---

**public Folder[] listSubscribed() throws MessagingException**

```java
public Folder[] listSubscribed() throws MessagingException
```
listSubscribed

public Folder[] listSubscribed() throws MessagingException

Convenience method that returns the list of subscribed folders under this Folder. This method just calls the listSubscribed(String pattern) method with ")%" as the match pattern. This method can be invoked on a closed Folder.

Returns:
array of subscribed Folder objects under this Folder. An empty array is returned if no subscribed subfolders exist.

Throws:
FolderNotFoundException - if this folder does not exist.
MessagingException

See Also:
listSubscribed(java.lang.String)

abstract public char getSeparator() throws MessagingException

getSeparator

public abstract char getSeparator()
throws MessagingException

Return the delimiter character that separates this Folder's pathname from the names of immediate subfolders. This method can be invoked on a closed Folder.

**Returns:**
Hierarchy separator character

**Throws:**
 FolderNotFoundException - if the implementation requires the folder to exist, but it does not
 MessagingException

---

abstract public int getType() throws MessagingException

Folder Folder

return

Throws
 FolderNotFoundException: 

See also
 HOLDS_FOLDERS, HOLDS_MESSAGES

---

getType

public abstract int getType()

throws MessagingException

Returns the type of this Folder, that is, whether this folder can hold messages or subfolders or both. The returned value is an integer bitfield with the appropriate bits set. This method can be invoked on a closed folder.

**Returns:**
integer with appropriate bits set

**Throws:**
 FolderNotFoundException - if this folder does not exist.
 MessagingException

**See Also:**
 HOLDS_FOLDERS, HOLDS_MESSAGES
abstract public boolean create(int type) throws MessagingException

Store

Folder Store FolderListener CREATED
FolderEvent

type
return true false

Throws MessagingException:

See HOLDs_FOLDERS, HOLDs_MESSAGES,
also javax.mail.event.FolderEvent

create

custom public abstract boolean create(int type)
throws MessagingException

Create this folder on the Store. When this folder is created, any folders in its path that do not exist are also created.

If the creation is successful, a CREATED FolderEvent is delivered to any FolderListeners registered on this Folder and this Store.

Parameters:

type - The type of this folder.

Returns:

ture if the creation succeeds, else false.

Throws:

MessagingException

See Also:

HOLDs_FOLDERS, HOLDs_MESSAGES, FolderEvent

public boolean isSubscribed()

Folder true
Folder

true

    return  Folder  true

isSubscribed

public boolean isSubscribed()

    Returns true if this Folder is subscribed.
    This method can be invoked on a closed Folder.
    The default implementation provided here just returns true.

    Returns:
        true if this Folder is subscribed

public void setSubscribed(boolean subscribe) throws MessagingException
Folder Store

MethodNotSupportedException

    subscribe  true  false
    Throws  FolderNotFoundException:
    Throws  MethodNotSupportedException: Store
    Throws  MessagingException:
public void setSubscribed(boolean subscribe) throws MessagingException

Subscribe or unsubscribe this Folder. Not all Stores support subscription.

This method can be invoked on a closed Folder.

The implementation provided here just throws the MethodNotSupportedException.

Parameters:
subscribe - true to subscribe, false to unsubscribe

Throws:
FolderNotFoundException - if this folder does not exist.
MethodNotSupported Exception - if this store does not support subscription
MessagingException

abstract public boolean hasNewMessages() throws MessagingException

FolderPath true Folder IMAP Folder "" Folder Folder RECENT

Folder

Message Folder

return Store Message true

Throws FolderNotFoundException:
Throws MessagingException:
Returns true if this Folder has new messages since the last time this indication was reset. When this indication is set or reset depends on the Folder implementation (and in the case of IMAP, depends on the server). This method can be used to implement a lightweight "check for new mail" operation on a Folder without opening it. (For example, a thread that monitors a mailbox and flags when it has new mail.) This method should indicate whether any messages in the Folder have the RECENT flag set.

Note that this is not an incremental check for new mail, i.e., it cannot be used to determine whether any new messages have arrived since the last time this method was invoked. To implement incremental checks, the Folder needs to be opened.

This method can be invoked on a closed Folder that can contain Messages.

**Returns:**
true if the Store has new Messages

**Throws:**
FolderNotFoundException - if this folder does not exist.
MessagingException

---

abstract public Folder getFolder(String name) throws MessagingException
Folder Store Folder

Store Folder

Store Folder Folder

Folder

name Folder

return Folder
Throws  MessagingException:

getFolder

public abstract Folder getFolder(String name) throws MessagingException

Return the Folder object corresponding to the given name. Note that this folder does not physically have to exist in the Store. The exists() method on a Folder indicates whether it really exists on the Store.

In some Stores, name can be an absolute path if it starts with the hierarchy delimiter. Otherwise, it is interpreted relative to this Folder.

Folder objects are not cached by the Store, so invoking this method on the same name multiple times will return that many distinct Folder objects.

This method can be invoked on a closed Folder.

Parameters:
name - name of the Folder

Returns:
Folder object

Throws:
MessagingException

abstract public boolean delete(boolean recurse) throws MessagingException
Folder Folder

recurse  true  false

• (type == HOLDS_MESSAGES)
Store  FolderEvent
(type == HOLDS_FOLDERS)
Store FolderEvent
false

Store FolderEvent
3

1. delete() false
2. delete() true exists() true
   Store FolderEvent
3. HOLDS_FOLDERS | HOLDS_MESSAGES
   HOLDS_FOLDERS delete() true
   exists() true
   Store FolderEvent

return Folder true
Throws FolderNotFoundException:
Throws IllegalStateException:
Throws MessagingException:
See also javax.mail.event.FolderEvent

delete

public abstract boolean delete(boolean recurse)
throws MessagingException

Delete this Folder. This method will succeed only on a closed Folder.

The recurse flag controls whether the deletion affects subfolders or not. If true, all subfolders are deleted, then this folder itself is deleted. If false, the behaviour is dependent on the folder type and is elaborated below:
The folder can contain only messages: (type == HOLDS_MESSAGES).
All messages within the folder are removed. The folder itself is then removed. An appropriate FolderEvent is generated by the Store and this folder.

The folder can contain only subfolders: (type == HOLDS_FOLDERS).
If this folder is empty (does not contain any subfolders at all), it is removed. An appropriate FolderEvent is generated by the Store and this folder.
If this folder contains any subfolders, the delete fails and returns false.

The folder can contain subfolders as well as messages:
If the folder is empty (no messages or subfolders), it is removed. If the folder contains no subfolders, but only messages, then all messages are removed. The folder itself is then removed. In both the above cases, an appropriate FolderEvent is generated by the Store and this folder.

If the folder contains subfolders there are 3 possible choices an implementation is free to do:

1. The operation fails, irrespective of whether this folder contains messages or not. Some implementations might elect to go with this simple approach. The delete() method returns false.
2. Any messages within the folder are removed. Subfolders are not removed. The folder itself is not removed or affected in any manner. The delete() method returns true. And the exists() method on this folder will return true indicating that this folder still exists.
   An appropriate FolderEvent is generated by the Store and this folder.
3. Any messages within the folder are removed. Subfolders are not removed. The folder itself changes its type from HOLDS_FOLDERS | HOLDS_MESSAGES to HOLDS_FOLDERS. Thus new messages cannot be added
to this folder, but new subfolders can be created underneath. The delete() method returns true indicating success. The exists() method on this folder will return true indicating that this folder still exists. An appropriate FolderEvent is generated by the Store and this folder.

**Returns:**
true if the Folder is deleted successfully

**Throws:**
- `FolderNotFoundException` - if this folder does not exist
- `IllegalStateException` - if this folder is not in the closed state.
- `MessagingException`

**See Also:**
`FolderEvent`

---

**abstract public boolean renameTo(Folder f) throws MessagingException**

```java
public abstract boolean renameTo(Folder f) throws MessagingException
```

#### Store FolderListener RENAMED FolderEvent

```java
f Folder
return Folder true
```

**Throws**
- `FolderNotFoundException`
- `IllegalStateException`
- `MessagingException`

**See also**
`javax.mail.event.FolderEvent`

---

**renameTo**

```java
public abstract boolean renameTo(Folder f) throws MessagingException
```

Rename this Folder. This method will succeed only on a closed Folder.
If the rename is successful, a RENAMED FolderEvent is delivered to FolderListeners registered on this folder and its containing Store.

**Parameters:**
- `f` - a folder representing the new name for this Folder

**Returns:**
- true if the Folder is renamed successfully

**Throws:**
- `FolderNotFoundException` - if this folder does not exist
- `IllegalStateException` - if this folder is not in the closed state.
- `MessagingException` -

**See Also:**
- `FolderEvent`

---

abstract public void open(int mode) throws `MessagingException`

Folder Message Folder

Folder ConnectionListener OPENED
ConnectionEvent

Store

- `mode`:
  - Folder READ_ONLY
  - READ_WRITE

Throws
- `FolderNotFoundException`:

Throws
- `IllegalStateException`:

Throws
- `MessagingException`:

See
- `READ_ONLY`, `READ_WRITE`, `getType()`, `javax.mail.event.ConnectionEvent`

also

---

open

public abstract void open(int mode)

throws `MessagingException`

Open this Folder. This method is valid only on Folders that can
contain Messages and that are closed.

If this folder is opened successfully, an OPENED ConnectionEvent is delivered to any ConnectionListeners registered on this Folder.

The effect of opening multiple connections to the same folder on a specific Store is implementation dependent. Some implementations allow multiple readers, but only one writer. Others allow multiple writers as well as readers.

**Parameters:**
- mode - open the Folder READ_ONLY or READ_WRITE

**Throws:**
- `FolderNotFoundException` - if this folder does not exist.
- `IllegalStateException` - if this folder is not in the closed state.
- `MessagingException`

**See Also:**
- `READ_ONLY`, `READ_WRITE`, `getType()`, `ConnectionEvent`

---

abstract public void close(boolean expunge) throws `MessagingException`

Folder Folder

Folder ConnectionListener CLOSED
ConnectionEvent MessagingException

expunge   ture
Throws    IllegalStateException:
Throws    `MessagingException`:
See also   `javax.mail.event.ConnectionEvent`

---

close

public abstract void close(boolean expunge) throws `MessagingException`
Close this Folder. This method is valid only on open Folders.

A CLOSED ConnectionEvent is delivered to any ConnectionListeners registered on this Folder. Note that the folder is closed even if this method terminates abnormally by throwing a MessagingException.

**Parameters:**
- expunge - expunges all deleted messages if this flag is true

**Throws:**
- `IllegalStateException` - if this folder is not opened
- `MessagingException`

**See Also:**
- `ConnectionEvent`

---

```java
abstract public boolean isOpen()
Folder 'open'
return Folder 'open' true
```

**isOpen**

```java
public abstract boolean isOpen()
```

Indicates whether this Folder is in the 'open' state.

**Returns:**
- true if this Folder is in the 'open' state.

---

```java
public int getMode()
```

 Returns: 
- `Folder.READ_ONLY`
- `Folder.READ_WRITE`
- `-1`

**Throws**
- `IllegalStateException:`

return since JavaMail 1.1
- en
**getMode**

```java
class Folder {
    public int getMode() {
        return null;
    }
}
```

**Returns:**
the open mode of this folder

**Throws:**
`IllegalStateException` - if this folder is not opened

**Since:**
JavaMail 1.1

---

**abstract public Flags getPermanentFlags()**

```java
public abstract Flags getPermanentFlags() {
    return null;
}
```

Get the permanent flags supported by this Folder. Returns a Flags object that contains all the flags supported.

The special flag `Flags.USER` indicates that this Folder supports arbitrary user-defined flags.

The supported permanent flags for a folder may not be available until
the folder is opened.

**Returns:**
permanent flags, or null if not known

```java
abstract public int getMessageCount() throws MessagingException
Folder

Folder -1

-1

    return -1

Throws FolderNotFoundException:
Throws MessagingException:
```

**getMessageCount**

```java
public abstract int getMessageCount() throws MessagingException
```

Get total number of messages in this Folder.

This method can be invoked on a closed folder. However, note that for some folder implementations, getting the total message count can be an expensive operation involving actually opening the folder. In such cases, a provider can choose not to support this functionality in the closed state, in which case this method must return -1.

Clients invoking this method on a closed folder must be aware that this is a potentially expensive operation. Clients must also be prepared to handle a return value of -1 in this case.

**Returns:**
total number of messages. -1 may be returned by certain
implementations if this method is invoked on a closed folder.

Throws:
- `FolderNotFoundException` - if this folder does not exist.
- `MessagingException`

---

```java
public int getNewMessageCount() throws MessagingException
```

Get the number of new messages in this Folder.

This method can be invoked on a closed folder. However, note that for some folder implementations, getting the new message count can be an expensive operation involving actually opening the folder. In such cases, a provider can choose not to support this functionality in the closed state, in which case this method must return -1.

Clients invoking this method on a closed folder must be aware that this is a potentially expensive operation. Clients must also be prepared to handle a return value of -1 in this case.
This implementation returns -1 if this folder is closed. Else this implementation gets each Message in the folder using getMessage(int) and checks whether its RECENT flag is set. The total number of messages that have this flag set is returned.

**Returns:**
number of new messages. -1 may be returned by certain implementations if this method is invoked on a closed folder.

**Throws:**
- `FolderNotFoundException` - if this folder does not exist.
- `MessagingException` -

```java
public int getUnreadMessageCount() throws MessagingException
{
    Folder
    Folder

    int

    return -1

    getUnreadMessageCount() throws MessagingException
{
    Get the total number of unread messages in this Folder.

    This method can be invoked on a closed folder. However, note that for some folder implementations, getting the unread message count can be an expensive operation involving actually opening the folder.
```
In such cases, a provider can choose not to support this functionality in the closed state, in which case this method must return -1.

Clients invoking this method on a closed folder must be aware that this is a potentially expensive operation. Clients must also be prepared to handle a return value of -1 in this case.

This implementation returns -1 if this folder is closed. Else this implementation gets each Message in the folder using getMessage(int) and checks whether its SEEN flag is set. The total number of messages that do not have this flag set is returned.

Returns:
  total number of unread messages. -1 may be returned by certain implementations if this method is invoked on a closed folder.

Throws:
  FolderNotFoundException - if this folder does not exist.
  MessagingException

```java
public int getDeletedMessageCount() throws MessagingException
{
    return -1;
}
```

Folder -1

-1

-1

getMessage(int

return -1

Throws FolderNotFoundException:

Throws MessagingException:

since JavaMail 1.3

en
getDeletedMessageCount

public int getDeletedMessageCount() throws MessagingException

Get the number of deleted messages in this Folder.

This method can be invoked on a closed folder. However, note that for some folder implementations, getting the deleted message count can be an expensive operation involving actually opening the folder. In such cases, a provider can choose not to support this functionality in the closed state, in which case this method must return -1.

Clients invoking this method on a closed folder must be aware that this is a potentially expensive operation. Clients must also be prepared to handle a return value of -1 in this case.

This implementation returns -1 if this folder is closed. Else this implementation gets each Message in the folder using getMessage(int) and checks whether its DELETED flag is set. The total number of messages that have this flag set is returned.

Returns:
   number of deleted messages. -1 may be returned by certain implementations if this method is invoked on a closed folder.

Throws:
   FolderNotFoundException - if this folder does not exist.
   MessagingException

Since:
   JavaMail 1.3
Folder getMessage Message

Message

\[
\text{msgnum} \\
\text{return Message} \\
\text{Throws FolderNotFoundException:} \\
\text{Throws IllegalStateException:} \\
\text{Throws IndexOutOfBoundsException:} \\
\text{Throws MessagingException:} \\
\text{See also getMessageCount, fetch}
\]

getMessage

public abstract Message getMessage(int msgnum) throws MessagingException

Get the Message object corresponding to the given message number. A Message object's message number is the relative position of this Message in its Folder. Messages are numbered starting at 1 through the total number of message in the folder. Note that the message number for a particular Message can change during a session if other messages in the Folder are deleted and the Folder is expunged.

Message objects are light-weight references to the actual message that get filled up on demand. Hence Folder implementations are expected to provide light-weight Message objects.

Unlike Folder objects, repeated calls to getMessage with the same message number will return the same Message object, as long as no messages in this folder have been expunged.

Since message numbers can change within a session if the folder is expunged, clients are advised not to use message numbers as references to messages. Use Message objects instead.
Parameters:
   msgnum - the message number

Returns:
   the Message object

Throws:
   FolderNotFoundException - if this folder does not exist.
   IllegalStateException - if this folder is not opened
   IndexOutOfBoundsException - if the message number is out of range.
   MessagingException

See Also:
   getMessageCount(), fetch(javax.mail.Message[], javax.mail.FetchProfile)

public Message[] getMessages(int start, int end) throws MessagingException

getMessage(index) Message

Message Folder Message

getMessages
public Message[] getMessages(int start, int end) throws MessagingException

Get the Message objects for message numbers ranging from start through end, both start and end inclusive. Note that message numbers start at 1, not 0.

Message objects are light-weight references to the actual message that get filled up on demand. Hence Folder implementations are expected to provide light-weight Message objects.

This implementation uses getMessage(index) to obtain the required Message objects. Note that the returned array must contain (end-start+1) Message objects.

Parameters:
  start - the number of the first message
  end - the number of the last message

Returns:
  the Message objects

Throws:
  FolderNotFoundException - if this folder does not exist.
  IllegalStateException - if this folder is not opened.
  IndexOutOfBoundsException - if the start or end message numbers are out of range.
  MessagingException

See Also:
  fetch(javax.mail.Message[], javax.mail.FetchProfile)

public Message[] getMessages(int[] msgnums) throws MessagingException

Message

Message Folder Message

getMessage(index) Message
Message
getMessages

public Message[] getMessages(int[] msgnums)
throws MessagingException

Get the Message objects for message numbers specified in the array.

Message objects are light-weight references to the actual message that get filled up on demand. Hence Folder implementations are expected to provide light-weight Message objects.

This implementation uses getMessage(index) to obtain the required Message objects. Note that the returned array must contain msgnums.length Message objects.

Parameters:
msgnums - the array of message numbers

Returns:
the array of Message objects.

Throws:
FolderNotFoundException - if this folder does not exist.
IllegalStateException - if this folder is not opened.
IndexOutOfBoundsException - if any message number in the given array is out of range.
MessagingException

See Also:
fetch(javax.mail.Message[], javax.mail.FetchProfile)
public Message[] getMessages() throws MessagingException

Folder Message
Message Message

getMessageCount() getMessage() 1 Message

return Message
Throws FolderNotFoundException:
Throws IllegalStateException:
Throws MessagingException: fetch
See also

getMessages

public Message[] getMessages()
throws MessagingException

Get all Message objects from this Folder. Returns an empty array if the folder is empty. Clients can use Message objects (instead of sequence numbers) as references to the messages within a folder; this method supplies the Message objects to the client. Folder implementations are expected to provide light-weight Message objects, which get filled on demand.

This implementation invokes getMessageCount() to get the current message count and then uses getMessage() to get Message objects from 1 till the message count.

Returns: array of Message objects, empty array if folder is empty.
Throws:
FolderNotFoundException - if this folder does not exist.
IllegalStateException - if this folder is not opened.
MessagingException
See Also:
abstract public void appendMessages(Message[] msgs) throws MessagingException
Message Folder MessageCountListener MessageCountEvent

Message Message
msgs Message
Throws FolderNotFoundException:
Throws MessagingException:

appendMessages

public abstract void appendMessages(Message[] msgs) throws MessagingException

Append given Messages to this folder. This method can be invoked on a closed Folder. An appropriate MessageCountEvent is delivered to any MessageCountListener registered on this folder when the messages arrive in the folder.

Folder implementations must not abort this operation if a Message in the given message array turns out to be an expunged Message.

Parameters:
msgs - array of Messages to be appended

Throws:
FolderNotFoundException - if this folder does not exist.
MessagingException - if the append failed.

public void fetch(Message[] msgs, FetchProfile fp) throws MessagingException
Message FetchProfile
SubjectFrom X-mailer

Message[] msgs = folder.getMessages();

FetchProfile fp = new FetchProfile();
fp.add(FetchProfile.Item.ENVELOPE);
fp.add("X-mailer");
folder.fetch(msgs, fp);

for (int i = 0; i < folder.getMessageCount(); i++) {
display(msg[i].getFrom());
display(msg[i].getSubject());
display(msg[i].getHeader("X-mailer"));
}

fetch

public void fetch(Message[] msgs,
                FetchProfile fp)
    throws MessagingException

Prefetch the items specified in the FetchProfile for the given Messages.

Clients use this method to indicate that the specified items are
needed en-masse for the given message range. Implementations
are expected to retrieve these items for the given message range in
an efficient manner. Note that this method is just a hint to the
implementation to prefetch the desired items.

An example is a client filling its header-view window with the Subject, From and X-mailer headers for all messages in the folder.

```java
Message[] msgs = folder.getMessage();
FetchProfile fp = new FetchProfile();
fp.add(FetchProfile.Item.ENVELOPE);
fp.add("X-mailer");
folder.fetch(msgs, fp);
for (int i = 0; i < folder.getMessageCount(); i++) {
    display(msg[i].getFrom());
    display(msg[i].getSubject());
    display(msg[i].getHeader("X-mailer"));
}
```

The implementation provided here just returns without doing anything useful. Providers wanting to provide a real implementation for this method should override this method.

**Parameters:**
- `msgs` - fetch items for these messages
- `fp` - the FetchProfile

**Throws:**
- `IllegalStateException` - if this folder is not opened
- `MessagingException` - if this folder is not opened

```java
public void setFlags(Message[] msgs, Flags flag, boolean value) throws MessagingException
Message MessageChangedListener
MessageMessageChangedEvent
```

```java
Message Folder Flag Message
Message.setFlags
```
setFlags

```java
public void setFlags(Message[] msgs,
                      Flags flag,
                      boolean value)
throws MessagingException
```

Set the specified flags on the messages specified in the array. This will result in appropriate MessageChangedEvents being delivered to any MessageChangedListener registered on this Message's containing folder.

Note that the specified Message objects **must** belong to this folder. Certain Folder implementations can optimize the operation of setting Flags for a group of messages, so clients might want to use this method, rather than invoking `Message.setFlags` for each Message.

This implementation degenerates to invoking `setFlags()` on each Message object. Specific Folder implementations that can optimize this case should do so. Also, an implementation must not abort the operation if a Message in the array turns out to be an expunged Message.

**Parameters:**
- `msgs` - the array of message objects
- `flag` - Flags object containing the flags to be set
- `value` - set the flags to this boolean value

**Throws:**

- `IllegalStateException`: READ_ONLY
- `MessagingException`
IllegalStateException - if this folder is not opened or if it has been opened READ_ONLY.
MessagingException

See Also:
Message.setFlags(javax.mail.Flags, boolean),
MessageChangedEvent

public void setFlags(int start, int end, Flags flag, boolean value) throws MessagingException
start end start end 1 0
Message MessageChangedListener
MessageChangedEvent

Folder Flag Message

getMessage(int) Message setFlags Folder

start
end
flag Flag
value boolean

Throws IllegalStateException: READ_ONLY

Throws IndexOutOfBoundsException:

Throws MessagingException:

See also setFlags, javax.mail.event.MessageChangedEvent

setFlags

public void setFlags(int start,
int end,
Flags flag,
boolean value)
throws MessagingException

Set the specified flags on the messages numbered from start
through end, both start and end inclusive. Note that message numbers start at 1, not 0. This will result in appropriate MessageChangedEvents being delivered to any MessageChangedListener registered on this Message's containing folder.

Certain Folder implementations can optimize the operation of setting Flags for a group of messages, so clients might want to use this method, rather than invoking Message.setFlags for each Message.

The default implementation uses getMessage(int) to get each Message object and then invokes setFlags on that object to set the flags. Specific Folder implementations that can optimize this case should do so. Also, an implementation must not abort the operation if a message number refers to an expunged message.

Parameters:
- start - the number of the first message
- end - the number of the last message
- flag - Flags object containing the flags to be set
- value - set the flags to this boolean value

Throws:
- IllegalStateException - if this folder is not opened or if it has been opened READ_ONLY.
- IndexOutOfBoundsException - if the start or end message numbers are out of range.
- MessagingException

See Also:
- Message.setFlags(javax.mail.Flags, boolean)
- MessageChangedEvent

```java
public void setFlags(int[] msgnums, Flags flag, boolean value) throws MessagingException
```

Folder  Flag  Message
**setFlags**

```java
public void setFlags(int[] msgnums, Flags flag, boolean value)
    throws MessagingException
```

Set the specified flags on the messages whose message numbers are in the array. This will result in appropriate MessageChangedEvents being delivered to any MessageChangedListener registered on this Message's containing folder.

Certain Folder implementations can optimize the operation of setting Flags for a group of messages, so clients might want to use this method, rather than invoking `Message.setFlags` for each Message.

The default implementation uses `getMessage(int)` to get each `Message` object and then invokes `setFlags` on that object to set the flags. Specific Folder implementations that can optimize this case should do so. Also, an implementation must not abort the operation if a message number refers to an expunged message.

**Parameters:**
- `msgnums` - the array of message numbers
- `flag` - Flags object containing the flags to be set
- `value` - set the flags to this boolean value
Throws:

- `IllegalStateException` - if this folder is not opened or if it has been opened READ_ONLY.
- `IndexOutOfBoundsException` - if any message number in the given array is out of range.
- `MessagingException`

See Also:

- `Message.setFlags(javax.mail.Flags, boolean)`
- `MessageChangedEvent`

```java
public void copyMessages(Message[] msgs, Folder folder)
    throws MessagingException
```

Copy the specified Messages from this Folder into another Folder. This operation appends these Messages to the destination Folder.
The destination Folder does not have to be opened. An appropriate MessageCountEvent is delivered to any MessageCountListener registered on the destination folder when the messages arrive in the folder.

Note that the specified Message objects must belong to this folder. Folder implementations might be able to optimize this method by doing server-side copies.

This implementation just invokes appendMessages() on the destination folder to append the given Messages. Specific folder implementations that support server-side copies should do so, if the destination folder's Store is the same as this folder's Store. Also, an implementation must not abort the operation if a Message in the array turns out to be an expunged Message.

Parameters:
- msgs - the array of message objects
- folder - the folder to copy the messages to

Throws:
- FolderNotFoundException - if the destination folder does not exist.
- IllegalStateException - if this folder is not opened.
- MessagingException

See Also:
appendMessages(javax.mail.Message[])
expunge

public abstract Message[] expunge()
throws MessagingException

Expunge (permanently remove) messages marked DELETED. Returns an array containing the expunged message objects. The getMessageNumber method on each of these message objects returns that Message's original (that is, prior to the expunge) sequence number. A MessageCountEvent containing the expunged messages is delivered to any MessageCountListeners registered on the folder.

Expunge causes the renumbering of Message objects subsequent to the expunged messages. Clients that use message numbers as references to messages should be aware of this and should be prepared to deal with the situation (probably by flushing out existing message number caches and reloading them). Because of this complexity, it is better for clients to use Message objects as references to messages, rather than message numbers. Any expunged Messages objects still have to be pruned, but other Messages in that folder are not affected by the expunge.

After a message is expunged, only the isExpunged and getMessageNumber methods are still valid on the corresponding Message object; other methods may throw MessageRemovedException

Returns: array of expunged Message objects
Throws: FolderNotFoundException - if this folder does not exist
public Message[] search(SearchTerm term) throws MessagingException

Folder

search(term, getMessages())

term

return

Throws SearchException:

Throws FolderNotFoundException:

Throws IllegalStateException:

Throws MessagingException:

See also javax.mail.search.SearchTerm

search

public Message[] search(SearchTerm term) throws MessagingException

Search this Folder for messages matching the specified search criterion. Returns an array containing the matching messages. Returns an empty array if no matches were found.

This implementation invokes search(term, getMessages()), to apply the search over all the messages in this folder. Providers that can implement server-side searching might want to override this method to provide a more efficient implementation.

Parameters:

term - the search criterion

Returns:
array of matching messages

Throws:

SearchException - if the search term is too complex for the implementation to handle.

NotFoundException - if this folder does not exist.

IllegalStateException - if this folder is not opened.

MessagingException

See Also:

SearchTerm

public Message[] search(SearchTerm term, Message[] msgs) throws MessagingException

Message

match() SearchE

super.search()

term

msgs

return

Throws SearchException:

IllegalStateException:

MessagingException:

See also javax.mail.search.SearchTerm

search

public Message[] search(SearchTerm term, Message[] msgs) throws MessagingException

Search the given array of messages for those that match the specified search criterion. Returns an array containing the matching messages. Returns an empty array if no matches were found.
Note that the specified Message objects **must** belong to this folder.

This implementation iterates through the given array of messages, and applies the search criterion on each message by calling its `match()` method with the given term. The messages that succeed in the match are returned. Providers that can implement server-side searching might want to override this method to provide a more efficient implementation. If the search term is too complex or contains user-defined terms that cannot be executed on the server, providers may elect to either throw a `SearchException` or degenerate to client-side searching by calling `super.search()` to invoke this implementation.

**Parameters:**
- **term** - the search criterion
- **msgs** - the messages to be searched

**Returns:**
array of matching messages

**Throws:**
- `SearchException` - if the search term is too complex for the implementation to handle.
- `IllegalStateException` - if this folder is not opened
- `MessagingException`

**See Also:**
- `SearchTerm`

---

class public void addConnectionListener([ConnectionListener l])
Folder Connection

**ConnectionListener**
/ Connection Listener

**See also** `javax.mail.event.ConnectionEvent`

addConnectionListener
public void addConnectionListener(ConnectionListener l)

Add a listener for Connection events on this Folder.

The implementation provided here adds this listener to an internal list of ConnectionListeners.

**Parameters:**
1 - the Listener for Connection events

**See Also:**
ConnectionEvent

---

public void removeConnectionListener(ConnectionListener l)

Remove a Connection event listener.

The implementation provided here removes this listener from the internal list of ConnectionListeners.

**Parameters:**
1 - the listener

**See Also:**
addConnectionListener(javax.mail.event.ConnectionListener)

---

protected void notifyConnectionListeners(int type)
**ConnectionListener Connection**

**ConnectionListener**

<table>
<thead>
<tr>
<th>type</th>
<th>ConnectionEvent</th>
</tr>
</thead>
<tbody>
<tr>
<td>See also</td>
<td>javax.mail.event.ConnectionEvent</td>
</tr>
</tbody>
</table>

**notifyConnectionListeners**

```java
protected void notifyConnectionListeners(int type)
```

Notify all ConnectionListeners. Folder implementations are expected to use this method to broadcast connection events.

The provided implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered ConnectionListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

**Parameters:**
- type - the ConnectionEvent type

**See Also:**
- ConnectionEvent

---

**public void addFolderListener(FolderListener l)**

**Folder Listener**

<table>
<thead>
<tr>
<th>/</th>
<th>Folder Listener</th>
</tr>
</thead>
<tbody>
<tr>
<td>See also</td>
<td>javax.mail.event.FolderEvent</td>
</tr>
</tbody>
</table>

**addFolderListener**
public void addFolderListener(FolderListener l)

Add a listener for Folder events on this Folder.

The implementation provided here adds this listener to an internal list of FolderListeners.

Parameters:
- l - the Listener for Folder events

See Also:
FolderEvent

public void removeFolderListener(FolderListener l)

Remove a Folder event listener.

The implementation provided here removes this listener from the internal list of FolderListeners.

Parameters:
- l - the listener

See Also:
addFolderListener(javax.mail.event.FolderListener)

protected void notifyFolderListeners(int type)

Folder Store FolderListener Folder
notifyFolderListeners

protected void notifyFolderListeners(int type)

Notify all FolderListeners registered on this Folder and this folder's Store. Folder implementations are expected to use this method to broadcast Folder events.

The implementation provided here queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the FolderListeners registered on this folder. The implementation also invokes notifyFolderListeners on this folder's Store to notify any FolderListeners registered on the store.

Parameters:

type - type of FolderEvent

See Also:

notifyFolderRenamedListeners(javax.mail.Folder)

protected void notifyFolderRenamedListeners(Folder folder)

Folder Store FolderListener Folder

FolderListener Store
notifyFolderRenamedListeners Store FolderListener
since

JavaMail 1.1
See also notifyFolderListeners

**notifyFolderRenamedListeners**

```java
protected void notifyFolderRenamedListeners(Folder folder)
```

Notify all FolderListeners registered on this Folder and this folder's Store about the renaming of this folder. Folder implementations are expected to use this method to broadcast Folder events indicating the renaming of folders.

The implementation provided here queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the FolderListeners registered on this folder. The implementation also invokes `notifyFolderRenamedListeners` on this folder's Store to notify any FolderListeners registered on the store.

**Parameters:**

- `folder` - Folder representing the new name.

**Since:**

JavaMail 1.1

**See Also:**

- [notifyFolderListeners(int)](javax.mail.event.Folder)

---

**public void addMessageCountListener(MessageCountListener l)***

```java
Folder MessageCount

MessageCountListener
```

/ MessageCount Listener

See also [javax.mail.event.MessageCountEvent](javax.mail.event.MessageCountEvent)
addMessageCountListener

public void addMessageCountListener(MessageCountListener l)

Add a listener for MessageCount events on this Folder.

The implementation provided here adds this listener to an internal list of MessageCountListeners.

Parameters:
   l - the Listener for MessageCount events

See Also: MessageCountEvent

public void removeMessageCountListener(MessageCountListener l)

Remove a MessageCount listener.

The implementation provided here removes this listener from the internal list of MessageCountListeners.

Parameters:
   l - the listener

See Also: addMessageCountListener(javax.mail.event.MessageCountListener)
protected void notifyMessageAddedListeners(Message[] msgs)
MessageCountListener MessageCountListener

notifyMessageAddedListeners

protected void notifyMessageAddedListeners(Message[] msgs)

Notify all MessageCountListeners about the addition of messages into this folder. Folder implementations are expected to use this method to broadcast MessageCount events for indicating arrival of new messages.

The provided implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered MessageCountListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

protected void notifyMessageRemovedListeners(boolean removed, Message[] msgs)
MessageCountListener Folder MessageCount

MessageCountListener

notifyMessageRemovedListeners

protected void notifyMessageRemovedListeners(boolean removed, Message[] msgs)
Notify all MessageCountListeners about the removal of messages from this Folder. Folder implementations are expected to use this method to broadcast MessageCount events indicating removal of messages.

The provided implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered MessageCountListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

```
public void
addMessageChangedListener(MessageChangedListener l)
Folder MessageChanged
MessageChangedListener
    / MessageChanged Listener
See also javax.mail.event.MessageChangedEvent
```

```
addMessageChangedListener
public void addMessageChangedListener(MessageChangedListener l)
    Add a listener for MessageChanged events on this Folder.
    The implementation provided here adds this listener to an internal list of MessageChangedListeners.

Parameters:
    l - the Listener for MessageChanged events
See Also:
    MessageChangedEvent
```
removeMessageChangedListener(MessageChangedListener l)
MessageChanged
MessageChangedListener l
See also addMessageChangedListener

removeMessageChangedListener

public void removeMessageChangedListener(MessageChangedListener l)

Remove a MessageChanged listener.

The implementation provided here removes this listener from the internal list of MessageChangedListeners.

Parameters:
  l - the listener
See Also: addMessageChangedListener(javax.mail.event.MessageChangedListener)

protected void notifyMessageChangedListeners(int type, Message msg)
MessageChangedListener MessageChanged
MessageChangedListener

notifyMessageChangedListeners

protected void notifyMessageChangedListeners(int type, Message msg)
Notify all MessageChangedListeners. Folder implementations are expected to use this method to broadcast MessageChanged events.

The provided implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to registered MessageChangedListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

protected void finalize() throws Throwable

finalize

protected void finalize()
    throws Throwable

Overrides:
    finalize in class Object

Throws:
    Throwable

public String toString()

toString() Folder.getFullName() String null toString()

toString

public String toString()

    override the default toString(), it will return the String from Folder.getFullName() or if that is null, it will use the default toString() behavior.

Overrides:
    toString in class Object
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail.event Class FolderAdapter

java.lang.Object
   \ javax.mail.event.FolderAdapter

All Implemented Interfaces:
   EventListener, FolderListener

public abstract class FolderAdapter
extends Object
implements FolderListener

Implements: FolderListener

Folder

The adapter which receives Folder events. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.

Author:
   John Mani

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>FolderAdapter()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void folderCreated(FolderEvent e)</td>
</tr>
<tr>
<td>Invoked when a Folder is created.</td>
</tr>
<tr>
<td>void folderDeleted(FolderEvent e)</td>
</tr>
</tbody>
</table>
void Invoked when a folder is deleted.

void folderRenamed(FolderEvent e) Invoked when a folder is renamed.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public FolderAdapter()

FolderAdapter

public FolderAdapter()

Method Detail

public void folderCreated(FolderEvent e)

folderCreated

public void folderCreated(FolderEvent e)

Description copied from interface: FolderListener
Invoked when a Folder is created.

Specified by:
folderCreated in interface FolderListener

public void folderRenamed(FolderEvent e)
public void folderRenamed(FolderEvent e)

Description copied from interface: FolderListener
Invoked when a folder is renamed.

Specified by:
   folderRenamed in interface FolderListener

public void folderDeleted(FolderEvent e)

folderDeleted

public void folderDeleted(FolderEvent e)

Description copied from interface: FolderListener
Invoked when a folder is deleted.

Specified by:
   folderDeleted in interface FolderListener
javax.mail  **Class FolderClosedException**

java.lang.Object  
   └ java.lang.Throwable  
      └ java.lang.Exception  
         └ javax.mail.MessagingException  
            └ javax.mail.FolderClosedException

**All Implemented Interfaces:**  
 Serializable

---

```java
public class FolderClosedException extends MessagingException

**Extends:** Throwable > Exception > MessagingException

Messaging Folder

Folder "closed" Folder Messaging "open" Folder

getAddress()
```

This exception is thrown when a method is invoked on a Messaging object and the Folder that owns that object has died due to some reason.

Following the exception, the Folder is reset to the "closed" state. All messaging objects owned by the Folder should be considered invalid. The Folder can be reopened using the "open" method to reestablish the lost connection.

The `getAddress()` method returns more detailed information about the error that caused this exception.

**Author:**
Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>FolderClosedException(Folder folder) Constructor</td>
</tr>
<tr>
<td>FolderClosedException(Folder folder, String message) Constructor</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Folder getFolder()</td>
</tr>
<tr>
<td>Returns the dead Folder object</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.MessagingException

<table>
<thead>
<tr>
<th>Methods inherited from class javax.mail.MessagingException</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCause, getNextException, setNextException, toString</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang Throwable

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang Object

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

Constructor Detail

```java
public FolderClosedException(Folder folder) Constructor
```

folder Folder
FolderClosedException

public FolderClosedException(Folder folder)

Constructor

Parameters:
folder - the Folder

public FolderClosedException(Folder folder, String message)

Constructor

folder - the Folder
message - the detailed error message

FolderClosedException

public FolderClosedException(Folder folder, String message)

Constructor

Parameters:
folder - the Folder
message - the detailed error message

Method Detail

public Folder getFolder()

Folder

getchFolder
public Folder getFolder()

Returns the dead Folder object
javax.mail.event Class FolderEvent

java.lang.Object
   \ javax.mail.event.MailEvent
   \ javax.mail.event_FOLDEREvent

All Implemented Interfaces:
   Serializable

public class FolderEvent
   extends MailEvent

Extends: java.util.EventObject > MailEvent

Folder FolderEvent FolderListenerFolderListener
Folder Store

Store Folder IMAP

This class models Folder existence events. FolderEvents are delivered to FolderListeners registered on the affected Folder as well as the containing Store.

Service providers vary widely in their ability to notify clients of these events. At a minimum, service providers must notify listeners registered on the same Store or Folder object on which the operation occurs. Service providers may also notify listeners when changes are made through operations on other objects in the same virtual machine, or by other clients in the same or other hosts. Such notifications are not required and are typically not supported by mail protocols (including IMAP).

Author:
   John Mani, Bill Shannon
See Also:

Serialized Form

---

Field Summary

<table>
<thead>
<tr>
<th>static int CREATED</th>
<th>The folder was created.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int DELETED</td>
<td>The folder was deleted.</td>
</tr>
<tr>
<td>protected Folder folder</td>
<td>The folder the event occurred on.</td>
</tr>
<tr>
<td>protected Folder newFolder</td>
<td>The folder that represents the new name, in case of a RENAMED event.</td>
</tr>
<tr>
<td>static int RENAMED</td>
<td>The folder was renamed.</td>
</tr>
<tr>
<td>protected int type</td>
<td>The event type.</td>
</tr>
</tbody>
</table>

Fields inherited from class java.util.EventObject

source

---

Constructor Summary

FolderEvent(Object source, Folder oldFolder, Folder newFolder, int type)

Constructor.

FolderEvent(Object source, Folder folder, int type)

Constructor.

---

Method Summary

void dispatch(Object listener)

Invokes the appropriate FolderListener method

Folder getFolder()
Return the affected folder.

**Folder.getNewFolder()**

If this event indicates that a folder is renamed, (i.e, the event type is RENAMED), then this method returns the Folder object representing the new name.

**Folder.getType()**

Return the type of this event.

**Methods inherited from class java.util.EventObject**

getSource, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

**Field Detail**

**CREATED**

public static final int CREATED

The folder was created.

See Also: Constant Field Values

**DELETED**

public static final int DELETED

The folder was deleted.
See Also:
Constant Field Values

RENAMED

public static final int RENAMED

The folder was renamed.

See Also:
Constant Field Values

---

type

protected int type

The event type.

---

folder

protected transient Folder folder

The folder the event occurred on.

---

newFolder

protected transient Folder newFolder
The folder that represents the new name, in case of a RENAMED event.

Since:
JavaMail 1.1

Constructor Detail

public FolderEvent(Object source, Folder folder, int type)

Parameters:
source - The source of the event
folder - The affected folder
type - The event type

public FolderEvent(Object source, Folder oldFolder, Folder newFolder, int type)

RENAME
FolderEvent

public FolderEvent(Object source, Folder oldFolder, Folder newFolder, int type)

Constructor. Use for RENAMED events.

Parameters:
- source - The source of the event
- oldFolder - The folder that is renamed
- newFolder - The folder that represents the new name
- type - The event type

Since: JavaMail 1.1

Method Detail

public int getType()

    return

g getType

public int getType()

    Return the type of this event.

    Returns:
    type
public Folder getFolder()

    return
See also getNewFolder

getFolder

public Folder getFolder()

    Return the affected folder.

    Returns:
    the affected folder
See Also:
    getNewFolder()

public Folder getNewFolder()
RENAMED Folder

getFolder()

    return
since JavaMail 1.1
See also getFolder

getNewFolder

public Folder getNewFolder()

    If this event indicates that a folder is renamed, (i.e, the event type is
RENAMED), then this method returns the Folder object representing
the new name.

    The getFolder() method returns the folder that is renamed.
public void dispatch(Object listener)
FolderListener

dispatch

public void dispatch(Object listener)

Invokes the appropriate FolderListener method

Specified by: dispatch in class MailEvent
javax.mail.event Interface FolderListener

All Superinterfaces:
   EventListener

All Known Implementing Classes:
   FolderAdapter

public interface FolderListener

extends EventListener

Implements: java.util.EventListener
Implemented by: FolderAdapter

Folder Listener

This is the Listener interface for Folder events.

Author:
   John Mani

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>folderCreated</td>
<td>(FolderEvent e)</td>
<td>Invoked when a Folder is created.</td>
</tr>
<tr>
<td>folderDeleted</td>
<td>(FolderEvent e)</td>
<td>Invoked when a folder is deleted.</td>
</tr>
<tr>
<td>folderRenamed</td>
<td>(FolderEvent e)</td>
<td>Invoked when a folder is renamed.</td>
</tr>
</tbody>
</table>

Method Detail
public void folderCreated(FolderEvent e)

folderCreated

void folderCreated(FolderEvent e)

Invoked when a Folder is created.

public void folderDeleted(FolderEvent e)

folderDeleted

void folderDeleted(FolderEvent e)

Invoked when a folder is deleted.

public void folderRenamed(FolderEvent e)

folderRenamed

void folderRenamed(FolderEvent e)

Invoked when a folder is renamed.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
class FolderNotFoundException

extends MessagingException

Extends: Throwable > Exception > MessagingException

Folder

This exception is thrown by Folder methods, when those methods are invoked on a non existent folder.

Author:  
John Mani
See Also:  
Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FolderNotFoundException()</td>
<td>Constructs a MessagingException with no detail message.</td>
</tr>
<tr>
<td>FolderNotFoundException(Folder folder)</td>
<td>Constructs a MessagingException with the specified folder.</td>
</tr>
<tr>
<td>FolderNotFoundException(Folder folder, String s)</td>
<td>Constructs a MessagingException with the specified folder and the specified detail message.</td>
</tr>
</tbody>
</table>
**FolderNotFoundException**

*(String s, Folder folder)*

Constructs a MessagingException with the specified detail message and the specified folder.

---

## Method Summary

**getFolder**

Folder

Returns the offending Folder object.

---

### Methods inherited from class *javax.mail.MessagingException*

- getCause
- getNextException
- setNextException
- toString

### Methods inherited from class *java.lang.Throwable*

- fillInStackTrace
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace

### Methods inherited from class *java.lang.Object*

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

---

## Constructor Detail

**public FolderNotFoundException()**

MessagingException

---

**FolderNotFoundException**

**public FolderNotFoundException()**

Constructs a MessagingException with no detail message.

---

**public FolderNotFoundException(Folder folder)**

---
MessagingException

folder - Folder
since JavaMail 1.2

FolderNotFoundException

public FolderNotFoundException(Folder folder)

Constructs a MessagingException with the specified folder.

Parameters:
folder - the Folder

Since:
JavaMail 1.2

public FolderNotFoundException(Folder folder, String s)
MessagingException

folder - Folder
s - String
since JavaMail 1.2

FolderNotFoundException

public FolderNotFoundException(Folder folder, String s)

Constructs a MessagingException with the specified folder and the specified detail message.

Parameters:
folder - the Folder
s - the detail message

Since:
JavaMail 1.2
public FolderNotFoundException(String s, Folder folder)  
    MessagingException

    public FolderNotFoundException(String s, Folder folder)

    Constructs a MessagingException with the specified detail message and the specified folder.

    Parameters:
    s - the detail message
    folder - the Folder

    getFolder

    public Folder getFolder()

    Returns the offending Folder object.

    Returns:
    the Folder object. Note that the returned value can be null.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail.search  **Class FromStringTerm**

```java
java.lang.Object
  └javax.mail.search.SearchTerm
      └javax.mail.search.StringTerm
           └javax.mail.search.AddressStringTerm
                └javax.mail.search.FromStringTerm
```

All Implemented Interfaces:
- **Serializable**

```
public final class FromStringTerm
extends AddressStringTerm

Extends: SearchTerm > StringTerm > AddressStringTerm
```

From Address
```
FromString Address
```

*since*  JavaMail 1.1  **en**

This class implements string comparisons for the From Address header.

Note that this class differs from the FromTerm class in that this class does comparisons on address strings rather than Address objects. The string comparisons are case-insensitive.

*Since:*
- JavaMail 1.1

*See Also:*
- Serialized Form

---

**Field Summary**
### Fields inherited from class `javax.mail.search.StringTerm`

- `ignoreCase`, `pattern`

### Constructor Summary

**FromFromStringTerm(String pattern)**
Constructor.

### Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>equals(Object obj)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Equality comparison.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th><code>match(Message msg)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Check whether the address string specified in the constructor is a substring of the From address of this Message.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.mail.search.AddressStringTerm`

- `match`

### Methods inherited from class `javax.mail.search.StringTerm`

- `getIgnoreCase`, `getPattern`, `hashCode`, `match`

### Methods inherited from class `java.lang.Object`

- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

### Constructor Detail

**public FromFromStringTerm(String pattern)**

- `pattern`
FromStringTerm

public FromStringTerm(String pattern)

Constructor.

Parameters:
  pattern - the address pattern to be compared.

Method Detail

public boolean match(Message msg)

Check whether the address string specified in the constructor is a substring of the From address of this Message.

Specified by:
  match in class SearchTerm

Parameters:
  msg - The comparison is applied to this Message's From address.

Returns:
  true if the match succeeds, otherwise false.

public boolean equals(Object obj)
public boolean equals(Object obj)

Equality comparison.

Overrides:

equals in class AddressStringTerm
javax.mail.search  Class FromTerm

java.lang.Object  
   └ java.mail.search.SearchTerm  
      └ java.mail.search.AddressTerm  
         └ java.mail.search.FromTerm

All Implemented Interfaces:
   Serializable

public final class FromTerm
  extends AddressTerm

Extends: SearchTerm > AddressTerm

From Address

This class implements comparisons for the From Address header.

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

Fields inherited from class java.mail.search.AddressTerm
address

Constructor Summary

FromTerm(Address address)
   Constructor
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>boolean</td>
<td>match(Message msg)</td>
<td>The address comparator.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.search.AddressTerm
- getAddress, hashCode, match

Methods inherited from class java.lang.Object
- clone, finalize, getClass, notify, notifyAll, toString, wait, wait

Constructor Detail

public FromTerm(Address address)

Constructor

Parameters:
- address - The Address to be compared

Method Detail

public boolean match(Message msg)
The address comparator.

**Specified by:**
match in class `SearchTerm`

**Parameters:**
- `msg` - The address comparison is applied to this `Message`

**Returns:**
- true if the comparison succeeds, otherwise false

```java
public boolean match(Message msg)
```

**equals**

Equality comparison.

**Overrides:**
- `equals` in class `AddressTerm`

```java
public boolean equals(Object obj)
```
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
public class FunctionInfo
extends Object

(TLD)

since 2.0

Information for a function in a Tag Library. This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

Since:
   JSP 2.0

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>FunctionInfo(String name, String klass, String signature)</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getFunctionClass()</td>
<td>The class of the function.</td>
</tr>
<tr>
<td>String getFunctionSignature()</td>
<td>The signature of the function.</td>
</tr>
<tr>
<td>String getName()</td>
<td>The name of the function.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public FunctionInfo(String name, String klass, String signature)

Constructor for FunctionInfo.

Parameters:
name - The name of the function
klass - The class of the function
signature - The signature of the function

Method Detail

public String getName()

return

getName
public String getName()

    The name of the function.

    Returns:
    The name of the function

public String getFunctionClass()

    return

getFunctionClass

public String getFunctionClass()

    The class of the function.

    Returns:
    The class of the function

public String getFunctionSignature()

    return

getFunctionSignature

public String getFunctionSignature()

    The signature of the function.

    Returns:
    The signature of the function
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.el Class FunctionMapper

javax.lang.Object
  ↓ javax.el.FunctionMapper

public abstract class FunctionMapper
extends Object

EL

FunctionMapper  ${prefix:name()}

  since JSP 2.1

The interface to a map between EL function names and methods.

A FunctionMapper maps ${prefix:name()} style functions to a static method that can execute that function.

Since:
    JSP 2.1

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FunctionMapper()</td>
</tr>
</tbody>
</table>

---

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolveFunction</td>
<td>String prefix, String localName</td>
<td>Resolves the specified prefix and local name into a java.lang.Method.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public FunctionMapper()

FunctionMapper

public FunctionMapper()

Method Detail

abstract public Method resolveFunction(String prefix, String localName)

java.lang.Method

null

prefix ""

localName

return

resolveFunction

public abstract Method resolveFunction(String prefix, String localName)

Resolves the specified prefix and local name into a java.lang.Method.

Returns null if no function could be found that matches the given
prefix and local name.

**Parameters:**
- `prefix` - the prefix of the function, or "" if no prefix. For example, "fn" in `${fn:method()}`, or "" in `${method()}`.
- `localName` - the short name of the function. For example, "method" in `${fn:method()}`.

**Returns:**
- the static method to invoke, or `null` if no match was found.
| SUMMARY: NESTED | FIELD | CONSTR | METHOD |
| FRAMES: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
**Interface FunctionMapper**

**Deprecated. As of JSP 2.1, replaced by FunctionMapper**

The interface to a map between EL function names and methods.

Classes implementing this interface may, for instance, consult tag library information to resolve the map.

**Since:**

JSP 2.0

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>resolveFunction</strong></td>
<td>(String prefix, String localName)</td>
</tr>
</tbody>
</table>

**Deprecated.** Resolves the specified local name and prefix into a `Java.lang.Method`.

### Method Detail

```java
public Method resolveFunction(String prefix, String localName) {
    // Implementation
    return null;
}
```
resolveFunction

Method resolveFunction(String prefix, String localName)

Deprecated.
Resolves the specified local name and prefix into a Java.lang.Method. Returns null if the prefix and local name are not found.

Parameters:
prefix - the prefix of the function, or "" if no prefix.
localName - the short name of the function

Returns:
the result of the method mapping. Null means no entry found.
javax.annotation  Annotation Type Generated

@Documented
@Retention(value=SOURCE)
@Target(value={PACKAGE, TYPE, ANNOTATION_TYPE, METHOD, CONSTRUCTOR, FIELD, LOCAL_VARIABLE, PARAMETER})

public @interface Generated

<table>
<thead>
<tr>
<th>Implements: Annotation</th>
</tr>
</thead>
</table>
@Documented
@Retention(value=SOURCE)
@Target(value={PACKAGE, TYPE, ANNOTATION_TYPE, METHOD, CONSTRUCTOR, FIELD, LOCAL_VARIABLE, PARAMETER})

Generated com.company.package.classname


since 1.0

The Generated annotation is used to mark source code that has been generated. It can also be used to differentiate user written code from generated code in a single file. When used, the value element must have the name of the code generator. The recommended convention is to use the fully qualified name of the code generator in the value field. For example: com.company.package.classname. The date element is used to indicate the date the source was generated. The date element must follow the ISO 8601 standard. For example the date element would have the following value 2001-07-04T12:08:56.235-0700 which represents 2001-07-04 12:08:56 local time in the U.S. Pacific Time time zone. The comment element is a place holder for any comments that the code generator may want to include in the generated code.

Since: 1.0

Required Element Summary
This is used by the code generator to mark the generated classes and methods.

### Optional Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>comments</td>
<td>String</td>
<td>A place holder for any comments that the code generator may want to include in the generated code.</td>
</tr>
<tr>
<td>date</td>
<td>String</td>
<td>Date when the source was generated.</td>
</tr>
</tbody>
</table>

### Element Detail

**abstract public String[] value()**

**value**

```java
public abstract String[] value
```

This is used by the code generator to mark the generated classes and methods.

**abstract public String date()**

**date**

```java
public abstract String date
```

Date when the source was generated.
abstract public String comments()

comments

public abstract String comments

A place holder for any comments that the code generator may want to include in the generated code.

Default:

""

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms,

PS: 
### javax.persistence Annotation Type GeneratedValue

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface GeneratedValue

**Implements:** Annotation

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

```java
GeneratedValue

1

@Id
@GeneratedValue(strategy=SEQUENCE, generator="CUST_SEQ")
@Column(name="CUST_ID")
public Long getId() { return id; }
```

2

```java
@Id
@GeneratedValue(strategy=TABLE, generator="CUST_GEN")
@Column(name="CUST_ID")
Long id;
```

**since** Java Persistence 1.0

Provides for the specification of generation strategies for the values of primary keys. The `GeneratedValue` annotation may be applied to a primary key property or field of an entity or mapped superclass in conjunction with the `Id` annotation.

**Example 1:**

```java
@Id
@GeneratedValue(strategy=SEQUENCE, generator="CUST_SEQ")
@Column(name="CUST_ID")
```
public Long getId() { return id; }

Example 2:

@Id
@GeneratedValue(strategy=TABLE, generator="CUST_GEN")
@Column(name="CUST_ID")
Long id;

Since:
Java Persistence 1.0

---

Optional Element Summary

<table>
<thead>
<tr>
<th>generator</th>
<th>(Optional) The name of the primary key generator to use as specified in the SequenceGenerator or TableGenerator annotation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>strategy</td>
<td>(Optional) The primary key generation strategy that the persistence provider must use to generate the annotated entity primary key.</td>
</tr>
</tbody>
</table>

abstract public GenerationType strategy()

strategy

public abstract GenerationType strategy

(Optional) The primary key generation strategy that the persistence provider must use to generate the annotated entity primary key.

Default:
AUTO
abstract public String generator()

  SequenceGenerator  TableGenerator

ID

generator

public abstract String generator

  (Optional) The name of the primary key generator to use as specified in the SequenceGenerator or TableGenerator annotation.

  Defaults to the id generator supplied by persistence provider.

  Default: ""

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.persistence  **Enum GenerationType**

java.lang.Object
  └ java.lang.Enum<GenerationType>
    └ javax.persistence.GenerationType

**All Implemented Interfaces:**
  Serializable, Comparable<GenerationType>

```java
public enum GenerationType
extends Enum<GenerationType>
```

**Extends:** Enum<E>

**since**  
Java Persistence 1.0

Defines the types of primary key generation.

**Since:**  
Java Persistence 1.0

---

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Auto</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO</td>
<td>Indicates that the persistence provider should pick an appropriate strategy for the particular database.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identity</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IDENTITY</td>
<td>Indicates that the persistence provider must assign primary keys for the entity using database identity column.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sequence</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SEQUENCE</td>
<td>Indicates that the persistence provider must assign primary keys for the entity using database sequence column.</td>
</tr>
</tbody>
</table>
Indicates that the persistence provider must assign primary keys for the entity using an underlying database table to ensure uniqueness.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Enum`:
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

Methods inherited from class `java.lang.Object`:
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Enum Constant Detail

TABLE

```java
public static final GenerationType TABLE
```

Indicates that the persistence provider must assign primary keys for the entity using an underlying database table to ensure uniqueness.

SEQUENCE
public static final GenerationType SEQUENCE

Indicates that the persistence provider must assign primary keys for the entity using database sequence column.

IDENTITY

public static final GenerationType IDENTITY

Indicates that the persistence provider must assign primary keys for the entity using database identity column.

AUTO

public static final GenerationType AUTO

Indicates that the persistence provider should pick an appropriate strategy for the particular database. The AUTO generation strategy may expect a database resource to exist, or it may attempt to create one. A vendor may provide documentation on how to create such resources in the event that it does not support schema generation or cannot create the schema resource at runtime.

Method Detail

final public static GenerationType[] values()
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(GenerationType c : GenerationType.values())
    System.out.println(c);
```

**Returns:**
- an array containing the constants of this enum type, in the order they're declared

---

**public static GenerationTypevalueOf(String name)**

`valueOf` method is used to get an enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
- `name` - the name of the enum constant to be returned.

**Returns:**
- the enum constant with the specified name

**Throws:**
- `IllegalArgumentException` - if this enum type has no constant with the specified name
PS:
javax.resource.spi.security Interface GenericCredential

**Deprecated.** The preferred way to represent generic credential information is via the `org.ietf.jgss.GSSCredential` interface in J2SE Version 1.4, which provides similar functionality.

```java
class GenericCredential
```

- EIS GenericCredential EIS
- EIS GSS-API GSS-API

**version** 0.7
**since** 0.7
**deprecated**

The preferred way to represent generic credential information is via the `org.ietf.jgss.GSSCredential` interface in J2SE Version 1.4, which provides similar functionality.

**See also** `javax.security.auth.Subject`, `java.security.Principal`

The interface `javax.resource.spi.security.GenericCredential` defines a security mechanism independent interface for accessing security credential of a resource principal.

The `GenericCredential` interface provides a Java wrapper over an underlying mechanism specific representation of a security credential. For example, the `GenericCredential` interface can be used to wrap...
Kerberos credentials.

The connector architecture does not define any standard format and requirements for security mechanism specific credentials. For example, a security credential wrapped by a GenericCredential interface can have a native representation specific to an operating system.

The GenericCredential interface enables a resource adapter to extract information about a security credential. The resource adapter can then manage EIS sign-on for a resource principal by either:

- using the credentials in an EIS specific manner if the underlying EIS supports the security mechanism type represented by the GenericCredential instance, or,
- using GSS-API if the resource adapter and underlying EIS instance support GSS-API.

**Since:** 0.7

**Version:** 0.7

**Author:** Rahul Sharma

**See Also:** Subject, Principal

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(Object another)</code></td>
<td><strong>Deprecated.</strong> Tests if this GenericCredential instance refers to the same entity as the supplied object.</td>
</tr>
<tr>
<td><code>getCredentialData()</code></td>
<td><strong>Deprecated.</strong> Gets security data for a specific security mechanism represented by the GenericCredential.</td>
</tr>
<tr>
<td><code>getMechType()</code></td>
<td><strong>Deprecated.</strong> Returns the mechanism type for the GenericCredential instance.</td>
</tr>
</tbody>
</table>
### Method Detail

#### public String getName()

```java
GenericCredential
return
```

**getName**

```java
String getName()
```

**Deprecated.** Returns the name of the resource principal associated with a `GenericCredential` instance.

**Returns:**

Name of the principal

---

#### public String getMechType()

```java
GenericCredential GenericCredential GSS
(OID) GenericCredential OID
return
```

**getMechType**

```java
String getMechType()
```
Deprecated.
Returns the mechanism type for the GenericCredential instance. The mechanism type definition for GenericCredential should be consistent with the Object Identifier (OID) based representation specified in the GSS specification. In the GenericCredential interface, the mechanism type is returned as a stringified representation of the OID specification.

**Returns:**
mechanism type

---

```java
public byte[] getCredentialData() throws SecurityException

GenericCredential EIS
```

**Throws:**
`SecurityException`

---

getCredentialData

```java
byte[] getCredentialData() throws SecurityException
```

**Deprecated.**
Gets security data for a specific security mechanism represented by the GenericCredential. An example is authentication data required for establishing a secure association with an EIS instance on behalf of the associated resource principal.

The getCredentialData method returns the credential representation as an array of bytes. Note that the connector architecture does not define any standard format for the returned credential data.

**Returns:**
credential representation as an array of bytes.

**Throws:**
SecurityException - Failed operation due to security related error condition

public boolean equals(Object another)
GenericCredential
true false

equals

boolean equals(Object another)

Deprecated.
Tests if this GenericCredential instance refers to the same entity as the supplied object. The two credentials must be acquired over the same mechanisms and must refer to the same principal. Returns true if the two GenericCredentials refer to the same entity; false otherwise.

Overrides:
equals in class Object

public int hashCode()
GenericCredential
return

hashCode

int hashCode()

Deprecated.
Returns the hash code for this GenericCredential

Overrides:
hashCode in class Object
Returns:
hash code for this GenericCredential

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.handler Class GenericHandler

java.lang.Object

javax.xml.rpc.handler.GenericHandler

All Implemented Interfaces:
   Handler

public abstract class GenericHandler
    extends Object
    implements Handler

Implements: Handler

javax.xml.rpc.handler.GenericHandler Handler
   SOAP Message Handler
   GenericHandler Handler

GenericHandler Handler init destroy Handler

version 1.0

The javax.xml.rpc.handler.GenericHandler class implements the Handler interface. SOAP Message Handler developers should typically subclass GenericHandler class unless the Handler class needs another class as a superclass.

The GenericHandler class is a convenience abstract class that makes writing Handlers easy. This class provides default implementations of the lifecycle methods init and destroy and also different handle methods. A Handler developer should only override methods that it needs to specialize as part of the derived Handler implementation class.

Version:
   1.0

Author:
### Constructor Summary

<table>
<thead>
<tr>
<th>protect</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected</td>
<td><code>GenericHandler()</code></td>
</tr>
<tr>
<td></td>
<td>Default constructor</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void destroy()</code></td>
<td>The destroy method indicates the end of lifecycle for a Handler instance.</td>
</tr>
<tr>
<td><code>abstract QName[] getHeaders()</code></td>
<td>Gets the header blocks processed by this Handler instance.</td>
</tr>
<tr>
<td><code>boolean handleFault(MessageContext context)</code></td>
<td>The handleFault method processes the SOAP faults based on the SOAP message processing model.</td>
</tr>
<tr>
<td><code>boolean handleRequest(MessageContext context)</code></td>
<td>The handleRequest method processes the request SOAP message.</td>
</tr>
<tr>
<td><code>boolean handleResponse(MessageContext context)</code></td>
<td>The handleResponse method processes the response message.</td>
</tr>
<tr>
<td><code>void init(HandlerInfo config)</code></td>
<td>The init method to enable the Handler instance to initialize itself.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`: `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`
protected GenericHandler()

GenericHandler

protected GenericHandler()

Default constructor

### Method Detail

**public boolean handleRequest**(*MessageContext* context)

handleRequest  SOAP  true  SOAP  Handler

See also  handleRequest

**handleRequest**

**public boolean handleRequest**(*MessageContext* context)

- The `handleRequest` method processes the request SOAP message. The default implementation of this method returns `true`. This indicates that the handler chain should continue processing of the request SOAP message. This method should be overridden if the derived Handler class needs to specialize implementation of this method.

Specified by:  handleRequest in interface Handler

Parameters:
- `context` - `MessageContext` parameter provides access to the request message.

Returns:
- boolean Indicates the processing mode
  - `Return true` to indicate continued processing of the request
The HandlerChain takes the responsibility of invoking the next entity. The next entity may be the next handler in the HandlerChain or if this handler is the last handler in the chain, the next entity is the service endpoint object.

- Return `false` to indicate blocking of the request handler chain. In this case, further processing of the request handler chain is blocked and the target service endpoint is not dispatched. The JAX-RPC runtime system takes the responsibility of invoking the response handler chain next with the SOAPMessageContext. The Handler implementation class has the responsibility of setting the appropriate response SOAP message in either handleRequest and/or handleResponse method. In the default processing model, the response handler chain starts processing from the same Handler instance (that returned `false`) and goes backward in the execution sequence.

**See Also:**

`Handler.handleRequest(javax.xml.rpc.handler.MessageContext)`

---

```java
public boolean handleResponse(MessageContext context)
```

**See also**

- `handleResponse`

---

```java
public boolean handleResponse(MessageContext context)
```

The `handleResponse` method processes the response message. The default implementation of this method returns `true`. This indicates that the handler chain should continue processing of the response SOAP message. This method should be overridden if the derived Handler class needs to specialize implementation of this method.

**Specified by:**

- `handleResponse` in interface `Handler`
**Parameters:**

context - MessageContext parameter provides access to the response SOAP message

**Returns:**

boolean Indicates the processing mode

- Return `true` to indicate continued processing of the response handler chain. The HandlerChain invokes the `handleResponse` method on the next `Handler` in the handler chain.
- Return `false` to indicate blocking of the response handler chain. In this case, no other response handlers in the handler chain are invoked.

**See Also:**

`Handler.handleResponse(javax.xml.rpc.handler.MessageContext)`

---

```java
public boolean handleFault(MessageContext context)
```

**See also** `handleFault`

---

**handleFault**

```java
public boolean handleFault(MessageContext context)
```

The `handleFault` method processes the SOAP faults based on the SOAP message processing model. The default implementation of this method returns `true`. This indicates that the handler chain should continue processing of the SOAP fault. This method should be overridden if the derived Handler class needs to specialize implementation of this method.

**Specified by:**

`handleFault` in interface `Handler`

**Parameters:**

context - MessageContext parameter provides access to the SOAP message
Returns:
boolean Indicates the processing mode
  • Return true to indicate continued processing of SOAP Fault. The HandlerChain invokes the handleFault method on the next Handler in the handler chain.
  • Return false to indicate end of the SOAP fault processing. In this case, no other handlers in the handler chain are invoked.

See Also:
Handler.handleFault(javax.xml.rpc.handler.MessageContext)

public void init(HandlerInfo config)
init  Handler  Handler
  See also

init

public void init(HandlerInfo config)

The init method to enable the Handler instance to initialize itself. This method should be overridden if the derived Handler class needs to specialize implementation of this method.

Specified by:
init in interface Handler

See Also:
Handler.init(javax.xml.rpc.handler.HandlerInfo)

public void destroy()
destroy  Handler  Handler
  See also

destroy
public void destroy()

   The destroy method indicates the end of lifecycle for a Handler instance. This method should be overridden if the derived Handler class needs to specialize implementation of this method.

   Specified by:
       destroy in interface Handler

See Also:
    Handler.destroy()

abstract public javax.xml.namespace.QName[] getHeaders()

   Handler (header block)
   return QName

getHeaders

public abstract QName[] getHeaders()

   Gets the header blocks processed by this Handler instance.

   Specified by:
       getHeaders in interface Handler

Returns:
   Array of QNames of header blocks processed by this handler instance. QName is the qualified name of the outermost element of the Header block.
PS:
javax.servlet  **Class GenericServlet**

java.lang.Object  
    javax.servlet.GenericServlet

**All Implemented Interfaces:**  
    Serializable, Servlet, ServletConfig

**Direct Known Subclasses:**  
    HttpServlet

```java
public abstract class GenericServlet

extends Object
implements Servlet, ServletConfig, Serializable

Implements: Servlet, ServletConfig, java.io.Serializable
Extended by: HttpServlet

servlet Web  HTTP servlet
javax.servlet.http.HttpServlet

GenericServlet  Servlet  ServletConfig  servlet
GenericServlet  HttpServlet

GenericServlet  servlet  init  destroy  ServletConfig
                GenericServlet  log  ServletContext

servlet  service
```

Defines a generic, protocol-independent servlet. To write an HTTP servlet for use on the Web, extend HttpServlet instead.

GenericServlet implements the Servlet and ServletConfig interfaces. GenericServlet may be directly extended by a servlet, although it's more common to extend a protocol-specific subclass such as HttpServlet.
GenericServlet makes writing servlets easier. It provides simple versions of the lifecycle methods init and destroy and of the methods in the ServletConfig interface. GenericServlet also implements the log method, declared in the ServletContext interface.

To write a generic servlet, you need only override the abstract service method.

**Author:**
Various

**See Also:**
Serialized Form

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericServlet()</td>
<td>Does nothing.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>destroy()</td>
<td>Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.</td>
</tr>
<tr>
<td>String</td>
<td>getInitParameter(String name)</td>
<td>Returns a String containing the value of the named initialization parameter, or null if the parameter does not exist.</td>
</tr>
<tr>
<td>Enumeration</td>
<td>getInitParameterNames()</td>
<td>Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters.</td>
</tr>
<tr>
<td>ServletConfig</td>
<td>getServletConfig()</td>
<td>Returns this servlet's ServletConfig object.</td>
</tr>
<tr>
<td>ServletContext</td>
<td>getServletContext()</td>
<td>Returns a reference to the ServletContext in which</td>
</tr>
</tbody>
</table>
A convenience method which can be overridden so that there's no need to call super.init(config).

Called by the servlet container to indicate to a servlet that the servlet is being placed into service.

`service(ServletRequest req, ServletResponse res)`

Called by the servlet container to allow the servlet to respond to a request.

Methods inherited from class java.lang/Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public GenericServlet()

Does nothing. All of the servlet initialization is done by one of the init methods.

Method Detail

public void destroy()

Called by the servlet container to indicate to a servlet that the servlet is being taken out of service. See Servlet.destroy().

Specified by:
destroy in interface Servlet

public String getInitParameter(String name)

Returns a string containing the value of the named initialization parameter, or null if the parameter does not exist. See
**ServletConfig.getInitParameter(java.lang.String)**

This method is supplied for convenience. It gets the value of the named parameter from the servlet's ServletConfig object.

**Specified by:**
getInitParameter in interface ServletConfig

**Parameters:**
- name - a String specifying the name of the initialization parameter

**Returns:**
- String a string containing the value of the initialization parameter

---

```java
public java.util.Enumeration<E> getInitParameterNames()
```

**getInitParameterNames**

```java
public Enumeration getInitParameterNames()
```

Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters. See ServletConfig.getInitParameterNames().

This method is supplied for convenience. It gets the parameter names from the servlet's ServletConfig object.

**Specified by:**
getInitParameterNames in interface ServletConfig
Returns:
Enumeration an enumeration of String objects containing the names of the servlet's initialization parameters

```java
public ServletConfig getServletConfig()

getServletConfig
```

public ServletConfig getServletConfig()

Returns this servlet's ServletConfig object.

Specified by:
getServletConfig in interface Servlet

Returns:
ServletConfig the ServletConfig object that initialized this servlet

See Also:
Servlet.init(javax.servlet.ServletConfig)

```java
public ServletContext getServletContext()

getServletContext
```

public ServletContext getServletContext()

Returns a reference to the ServletContext in which this servlet is
running. See `ServletConfig.getServletContext()`.

This method is supplied for convenience. It gets the context from the servlet's `ServletConfig` object.

**Specified by:**

`getServletContext` in interface `ServletConfig`

**Returns:**

`ServletContext` the `ServletContext` object passed to this servlet by the `init` method

**See Also:**

`ServletContext`

```java
public String getServletInfo()

servlet
    return String servlet

getServletInfo
```

Returns information about the servlet, such as author, version, and copyright. By default, this method returns an empty string. Override this method to have it return a meaningful value. See `Servlet.getServletInfo()`.

**Specified by:**

`getServletInfo` in interface `Servlet`

**Returns:**

String information about this servlet, by default an empty string

```java
public void init(ServletConfig config) throws ServletException

servlet  servlet
```

`Servlet#init`
public void init(ServletConfig config) throws ServletException

Called by the servlet container to indicate to a servlet that the servlet is being placed into service. See Servlet.init(javax.servlet.ServletConfig).

This implementation stores the ServletConfig object it receives from the servlet container for later use. When overriding this form of the method, call super.init(config).

Specified by:
init in interface Servlet

Parameters:
config - the ServletConfig object that contains configuration information for this servlet

Throws:
ServletException - if an exception occurs that interrupts the servlet's normal operation

See Also:
UnavailableException

---

public void init() throws ServletException

super.init(config)
Throws: ServletException: servlet

init

public void init() throws ServletException

A convenience method which can be overridden so that there's no need to call super.init(config).

Instead of overriding init(ServletConfig), simply override this method and it will be called by GenericServlet.init(ServletConfig config). The ServletConfig object can still be retrieved via getServletConfig().

Throws:
ServletException - if an exception occurs that interrupts the servlet's normal operation

log

public void log(String msg)

Writes the specified message to a servlet log file, prepended by the servlet's name. See ServletContext.log(String).

Parameters:
msg - a String specifying the message to be written to the log file

public void log(String message, Throwable t)
servlet Throwable servlet Throwable
log(String, Throwable)
message String
t java.lang.Throwable

log

public void log(String message, Throwable t)

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file, prepended by the servlet's name. See ServletContext.log(String, Throwable).

Parameters:
message - a String that describes the error or exception
t - the java.lang.Throwable error or exception

abstract public void service(HttpServletRequest req, HttpServletResponse res) throws ServletException, java.io.IOException
servlet servlet

HttpServletRequest
req ServletRequest
res servlet

Throws ServletException: servlet
Throws java.io.IOException:

service

public abstract void service(HttpServletRequest req, HttpServletResponse res)
throws ServletException, IOException

Called by the servlet container to allow the servlet to respond to a request. See Servlet.service(javax.servlet.ServletRequest, javax.servlet.ServletResponse).

This method is declared abstract so subclasses, such as HttpServlet, must override it.

Specified by:
    service in interface Servlet

Parameters:
    req - the ServletRequest object that contains the client's request
    res - the ServletResponse object that will contain the servlet's response

Throws:
    ServletException - if an exception occurs that interferes with the servlet's normal operation occurred
    IOException - if an input or output exception occurs

public String getServletName()

    servlet ServletConfig#getServletName
    return servlet

getServletName

public String getServletName()

    Returns the name of this servlet instance. See ServletConfig.getServletName().

Specified by:
    getServletName in interface ServletConfig

Returns:
    the name of this servlet instance
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
```java
javax.ejb  Interface Handle

All Superinterfaces:
  Serializable

public interface Handle
  extends Serializable

Implements: java.io.Serializable

Handle  EJB  EJB  EJB ""

The Handle interface is implemented by all EJB object handles. A handle is an abstraction of a network reference to an EJB object. A handle is intended to be used as a "robust" persistent reference to an EJB object.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>EJBObject getEJBObject()</td>
</tr>
<tr>
<td>Obtain the EJB object reference represented by this handle.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public EJBObject getEJBObject() throws java.rmi.RemoteException</td>
</tr>
<tr>
<td>EJB</td>
</tr>
<tr>
<td>return EJB</td>
</tr>
<tr>
<td>Throws java.rmi.RemoteException: EJB</td>
</tr>
</tbody>
</table>
```
getEJBObject

EJBObject getEJBObject()

    throws RemoteException

Obtain the EJB object reference represented by this handle.

Returns:
    the EJB object reference represented by this handle.

Throws:
    RemoteException - The EJB object could not be obtained because of a system-level failure.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.ejb.spi Interface HandleDelegate**

```java
public interface HandleDelegate
```

HandleDelegate EJB javax.ejb.Handle javax.ejb.HomeHandle
EJB EJBObj ect EJBHome

JNDI "java:comp/HandleDelegate" HandleDelegate

The HandleDelegate interface is implemented by the EJB container. It is used by portable implementations of javax.ejb.Handle and javax.ejb.HomeHandle. It is not used by EJB components or by client components. It provides methods to serialize and deserialize EJBObj ect and EJBHome references to streams.

The HandleDelegate object is obtained by JNDI lookup at the reserved name "java:comp/HandleDelegate".

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>readEJBHome(ObjectInputStream istream)</code></td>
<td>Deserialize the EJBHome reference corresponding to a HomeHandle.</td>
</tr>
<tr>
<td><code>readEJBObject(ObjectInputStream istream)</code></td>
<td>Deserialize the EJBObj ect reference corresponding to a Handle.</td>
</tr>
<tr>
<td><code>writeEJBHome(EJBHome ejbHome, ObjectOutputStream ostream)</code></td>
<td>Serialize the EJBHome reference corresponding to a HomeHandle.</td>
</tr>
<tr>
<td><code>writeEJBObject(EJBObj ect ejbObj ect, ObjectOutputStream ostream)</code></td>
<td>Serialize the EJBObj ect reference corresponding to a</td>
</tr>
</tbody>
</table>
public void writeEJBObject(EJBOBJECT ejbObject, java.io.ObjectOutputStream ostream) throws java.io.IOException:
Handle EJBOBJECT

Handle writeObject ostream Handle writeObject

serialize the EJBObject reference corresponding to a Handle.

This method is called from the writeObject method of portable Handle implementation classes. The ostream object is the same object that was passed in to the Handle class's writeObject.

Parameters:
  ejbObject - The EJBObject reference to be serialized.
  ostream - The output stream.

Throws:
  IOException - The EJBObject could not be serialized because of a system-level failure.
public EJBObject readEJBObject(java.io.ObjectInputStream istream) throws java.io.IOException, ClassNotFoundException
Handle EJBObject

readEJBObject Handle readObject istream
Handle readObject readEJBObject istream
EJBObject EJBObject

    istream
return EJBObject

Throws java.io.IOException: EJBObject
Throws ClassNotFoundException: EJBObject

readEJBObject

EJBObject readEJBObject(ObjectInputStream istream)
throws IOException,
          ClassNotFoundException

Deserializes the EJBObj reference corresponding to a Handle.

readEJBObject is called from the readObject method of portable Handle implementation classes. The istream object is the same object that was passed in to the Handle class's readObject. When readEJBObject is called, istream must point to the location in the stream at which the EJBObj reference can be read. The container must ensure that the EJBObj reference is capable of performing invocations immediately after deserialization.

Parameters:
    istream - The input stream.

Returns:
    The deserialized EJBObj reference.

Throws:
    IOException - The EJBObj could not be deserialized because of a system-level failure.
ClassNotFoundException - The EJBObject could not be deserialized because some class could not be found.

```java
public void writeEJBHome(EJBHome ejbHome, java.io.ObjectOutputStream ostream) throws java.io.IOException
    HomeHandle HomeHandle writeObject ostream Handle
    writeObject ejbHome EJBHome
    ostream
    Throws java.io.IOException: EJBObject
```

**writeEJBHome**

```java
void writeEJBHome(EJBHome ejbHome, ObjectOutputStream ostream)
    throws IOException
```

Serialize the EJBHome reference corresponding to a HomeHandle.

This method is called from the writeObject method of portable HomeHandle implementation classes. The ostream object is the same object that was passed in to the Handle class's writeObject.

**Parameters:**
- `ejbHome` - The EJBHome reference to be serialized.
- `ostream` - The output stream.

**Throws:**
- `IOException` - The EJBObject could not be serialized because of a system-level failure.

```java
public EJBHome readEJBHome(java.io.ObjectInputStream
```
readEJBHome

Deserializes the EJBHome reference corresponding to a HomeHandle.

Parameters:
- istream - The input stream.

Returns:
- The deserialized EJBHome reference.

Throws:
- IOException - The EJBHome could not be deserialized because
of a system-level failure.

ClassNotFoundException - The EJBHome could not be deserialized because some class could not be found.
javax.xml.rpc.handler Interface Handler

All Known Implementing Classes:
   GenericHandler

public interface Handler

Implemented by: GenericHandler

javax.xml.rpc.handler.Handler SOAP SOAP
handleRequest handleResponse handleFault SOAPMessageContext
SOAPMessage SOAPMessage

version 1.0

The javax.xml.rpc.handler.Handler interface is required to be implemented by a SOAP message handler. The handleRequest, handleResponse and handleFault methods for a SOAP message handler get access to the SOAPMessage from the SOAPMessageContext. The implementation of these methods can modify the SOAPMessage including the headers and body elements.

Version:
   1.0

Author:
   Rahul Sharma

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong> destroy()</td>
</tr>
<tr>
<td><strong>QName[]</strong> getHeaders()</td>
</tr>
</tbody>
</table>
```java
boolean handleFault(MessageContext context)
    The handleFault method processes the SOAP faults based on the SOAP message processing model.

boolean handleRequest(MessageContext context)
    The handleRequest method processes the request message.

boolean handleResponse(MessageContext context)
    The handleResponse method processes the response SOAP message.

void init(HandlerInfo config)
    The init method enables the Handler instance to initialize itself.
```

### Method Detail

**public boolean handleRequest(MessageContext context)**

- **context**
  - MessageContext
  - boolean

- **return**
  - JAXRPCException: handleRequest
  - Throws
    - JAXRPCException
    - HandlerChain
    - HandlerChain
    - SOAP

- **Throws**
  - SOAPFaultException: SOAP Handler handleRequest / handleFault SOAP SOAP handleRequest SOAPFaultException HandlerChain SOAP HandlerChain handleFault HandlerChain handleFault Handler
  - handleRequest SOAPFaultException
handleRequest

boolean handleRequest(MessageContext context)

The handleRequest method processes the request message.

Parameters:
context - MessageContext parameter provides access to the request message.

Returns:
boolean Indicates the processing mode
- Return true to indicate continued processing of the request handler chain. The HandlerChain takes the responsibility of invoking the next entity. The next entity may be the next handler in the HandlerChain or if this handler is the last handler in the chain, the next entity is the service endpoint object.
- Return false to indicate blocking of the request handler chain. In this case, further processing of the request handler chain is blocked and the target service endpoint is not dispatched. The JAX-RPC runtime system takes the responsibility of invoking the response handler chain next with the SOAPMessageContext. The Handler implementation class has the responsibility of setting the appropriate response SOAP message in either handleRequest and/or handleResponse method. In the default processing model, the response handler chain starts processing from the same Handler instance (that returned false) and goes backward in the execution sequence.

Throws:
JAXRPCException - This exception indicates handler specific runtime error. If JAXRPCException is thrown by a handleRequest method, the HandlerChain terminates the further processing of this handler chain. On the server side, the HandlerChain generates a SOAP fault that indicates that the message could not be processed for reasons not directly attributable to the contents of the message itself but rather to a runtime error during the processing of the message. On the client side, the exception is propagated to the client code.
**SOAPFaultException** - This indicates a SOAP fault. The Handler implementation class has the responsibility of setting the SOAP fault in the SOAP message in either handleRequest and/or handleFault method. If SOAPFaultException is thrown by a server-side request handler's handleRequest method, the HandlerChain terminates the further processing of the request handlers in this handler chain and invokes the handleFault method on the HandlerChain with the SOAP message context. Next, the HandlerChain invokes the handleFault method on handlers registered in the handler chain, beginning with the Handler instance that threw the exception and going backward in execution. The client-side request handler's handleRequest method should not throw the SOAPFaultException.

```java
public boolean handleResponse(MessageContext context)

handleResponse SOAP
context MessageContext SOAP
    boolean

return
    true HandlerChain Handler handleResponse
    false

Throws JAXRPCException: handleResponse

JAXRPCException HandlerChain HandlerChain SOAP

handleResponse

boolean handleResponse(MessageContext context)

    The handleResponse method processes the response SOAP message.

Parameters:
    context - MessageContext parameter provides access to the response SOAP message
```
Returns:

boolean Indicates the processing mode

- Return true to indicate continued processing of the response handler chain. The HandlerChain invokes the handleResponse method on the next Handler in the handler chain.
- Return false to indicate blocking of the response handler chain. In this case, no other response handlers in the handler chain are invoked.

Throws:

JAXRPCException - Indicates handler specific runtime error. If JAXRPCException is thrown by a handleResponse method, the HandlerChain terminates the further processing of this handler chain. On the server side, the HandlerChain generates a SOAP fault that indicates that the message could not be processed for reasons not directly attributable to the contents of the message itself but rather to a runtime error during the processing of the message. On the client side, the runtime exception is propagated to the client code.

public boolean handleFault(MessageContext context) throws JAXRPCException

handleFault SOAP SOAP context MessageContext SOAP boolean

return

- true SOAP HandlerChain Handler handleFault
- false SOAP

JAXRPCException: handleFault JAXRPCException

Throws HandlerChain HandlerChain SOAP JAXRPCException

handleFault

boolean handleFault(MessageContext context)
The `handleFault` method processes the SOAP faults based on the SOAP message processing model.

**Parameters:**
- `context` - MessageContext parameter provides access to the SOAP message

**Returns:**
- boolean Indicates the processing mode
  - Return `true` to indicate continued processing of SOAP Fault. The HandlerChain invokes the `handleFault` method on the next `Handler` in the handler chain.
  - Return `false` to indicate end of the SOAP fault processing. In this case, no other handlers in the handler chain are invoked.

**Throws:**
- `JAXRPCException` - Indicates handler specific runtime error. If `JAXRPCException` is thrown by a `handleFault` method, the HandlerChain terminates the further processing of this handler chain. On the server side, the HandlerChain generates a SOAP fault that indicates that the message could not be processed for reasons not directly attributable to the contents of the message itself but rather to a runtime error during the processing of the message. On the client side, the `JAXRPCException` is propagated to the client code.

```java
public void init(HandlerInfo config)
```

```java
init Handler init HandlerInfo HandlerInfo
```

```java
init Handler SOAP HandlerInfo
```

```java
Throws JAXRPCException:
```

init
void init(HandlerInfo config)

The init method enables the Handler instance to initialize itself. The init method passes the handler configuration as a HandlerInfo instance. The HandlerInfo is used to configure the Handler (for example: setup access to an external resource or service) during the initialization.

In the init method, the Handler class may get access to any resources (for example; access to a logging service or database) and maintain these as part of its instance variables. Note that these instance variables must not have any state specific to the SOAP message processing performed in the various handle method.

**Parameters:**

HandlerInfo - Configuration for the initialization of this handler

**Throws:**

JAXRPCException - If initialization of the handler fails

public void destroy()

destroy Handler Handler
destroy cleanup

**Throws**

JAXRPCException:

void destroy()

The destroy method indicates the end of lifecycle for a Handler instance. The Handler implementation class should release its resources and perform cleanup in the implementation of the destroy method.

**Throws:**

JAXRPCException - If any error during destroy
public javax.xml.namespace.QName[] getHeaders()

return QName

getHeaders

QName[] getHeaders()

Gets the header blocks that can be processed by this Handler instance.

Returns:
Array of QNames of header blocks processed by this handler instance. QName is the qualified name of the outermost element of the Header block.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws.handler  **Interface Handler<C extends MessageContext>**

All Known Subinterfaces:
   LogicalHandler<C>, SOAPHandler<T>

```java
public interface Handler<C extends MessageContext> {

    // Method Summary
    void close(MessageContext context)
    boolean handleMessage(C context)
    boolean handleFault(C context)
}
```

**Implemented by:** GenericHandler

javax.xml.rpc.handler.Handler  SOAP SOAP
handleRequest  handleResponse  handleFault  SOAPMessageContext
SOAPMessage  SOAPMessage

**version**  1.0

The `Handler` interface is the base interface for JAX-WS handlers.

**Since:**
   JAX-WS 2.0
Method Detail

handleMessage

boolean handleMessage(C context)

The handleMessage method is invoked for normal processing of inbound and outbound messages. Refer to the description of the handler framework in the JAX-WS specification for full details.

Parameters:

context - the message context.

Returns:

An indication of whether handler processing should continue for the current message
- Return true to continue processing.
- Return false to block processing.

Throws:

RuntimeException - Causes the JAX-WS runtime to cease handler processing and generate a fault.
ProtocolException - Causes the JAX-WS runtime to switch to fault message processing.

handleFault

boolean handleFault(C context)

The handleFault method is invoked for fault message processing. Refer to the description of the handler framework in the JAX-WS specification for full details.

Parameters:

context - the message context
Returns:
An indication of whether handler fault processing should continue for the current message
- Return true to continue processing.
- Return false to block processing.

Throws:
RuntimeException - Causes the JAX-WS runtime to cease handler fault processing and dispatch the fault.
ProtocolException - Causes the JAX-WS runtime to cease handler fault processing and dispatch the fault.

close

void close(MessageContext context)

Called at the conclusion of a message exchange pattern just prior to the JAX-WS runtime dispatching a message, fault or exception. Refer to the description of the handler framework in the JAX-WS specification for full details.

Parameters:
context - the message context
javax.jws  Annotation Type HandlerChain

@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD, FIELD})
public @interface HandlerChain

**Implements**: Annotation
@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD, FIELD})

Web Service  Java  Web Service
@SOAPMessageHandlers

Associates the Web Service with an externally defined handler chain. This annotation is typically used in scenarios where embedding the handler configuration directly in the Java source is not appropriate; for example, where the handler configuration needs to be shared across multiple Web Services, or where the handler chain consists of handlers for multiple transports. It is an error to combine this annotation with the @SOAPMessageHandlers annotation.

**Author:**
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

### Required Element Summary

| String file | Location of the handler chain file. |

### Optional Element Summary

| String name | Deprecated. As of JSR-181 2.0 with no replacement. |
abstract public String file()

1. externalForm java.net.URL
   http://myhandlers.foo.com/handlerfile1.xml
2. bar/handlerfile1.xml

file

public abstract String file

Location of the handler chain file.

The location supports 2 formats:

2. A relative path from the source file or class file (ex: bar/handlerfile1.xml).

abstract public String
name()

deprecated JSR-181 2.0

name

@Deprecated
public abstract String name
Deprecation. As of JSR-181 2.0 with no replacement.

Name of the handler chain in the configuration file

Default:

""

PREV CLASS NEXT CLASS
SUMMARY: REQUIRED | OPTIONAL
FRAMES NO FRAMES
DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
The javax.xml.rpc.handler.HandlerChain represents a list of handlers. All elements in the HandlerChain are of the type javax.xml.rpc.handler.Handler.

An implementation class for the HandlerChain interface abstracts the policy and mechanism for the invocation of the registered handlers.

Version:
1.0
Author:
Rahul Sharma
See Also:
HandlerChain
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void destroy()</code></td>
<td>Indicates the end of lifecycle for a HandlerChain.</td>
</tr>
<tr>
<td><code>String[] getRoles()</code></td>
<td>Gets SOAP actor roles registered for this HandlerChain at this SOAP node.</td>
</tr>
<tr>
<td><code>boolean handleFault(MessageContext context)</code></td>
<td>The handleFault method initiates the SOAP fault processing for this handler chain.</td>
</tr>
<tr>
<td><code>boolean handleRequest(MessageContext context)</code></td>
<td>The handleRequest method initiates the request processing for this handler chain.</td>
</tr>
<tr>
<td><code>boolean handleResponse(MessageContext context)</code></td>
<td>The handleResponse method initiates the response processing for this handler chain.</td>
</tr>
<tr>
<td><code>void init(Map config)</code></td>
<td>Initializes the configuration for a HandlerChain.</td>
</tr>
<tr>
<td><code>void setRoles(String[] soapActorNames)</code></td>
<td>Sets SOAP Actor roles for this HandlerChain.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `java.util.List`:
- `add`, `add`, `addAll`, `addAll`, `clear`, `contains`, `containsAll`, `equals`, `get`, `hashCode`, `indexOf`, `isEmpty`, `iterator`, `lastIndexOf`, `listIterator`, `listIterator`, `remove`, `remove`, `removeAll`, `retainAll`, `set`, `size`, `subList`, `toArray`, `toArray`
**handleRequest**

boolean `handleRequest(MessageContext context)`

The `handleRequest` method initiates the request processing for this handler chain.

**Parameters:**
- `context` - `MessageContext` parameter provides access to the request SOAP message.

**Returns:**
- boolean Returns `true` if all handlers in chain have been processed. Returns `false` if a handler in the chain returned `false` from its `handleRequest` method.

**Throws:**
- `JAXRPCException` - if any processing error happens

**See Also:**
- `Handler.handleRequest(javax.xml.rpc.handler.MessageContext)`

---

**handleResponse**

`handleResponse(MessageContext context)`

`context` MessageContext SOAP

`return` boolean `true handleResponse`

**Throws**
- `JAXRPCException`: `handleResponse`

**See also**
- `handleResponse`

**handleResponse**

boolean `handleResponse(MessageContext context)`

The `handleResponse` method initiates the response processing for this handler chain.

**Parameters:**
- `context` - `MessageContext` parameter provides access to the response SOAP message.

**Returns:**
boolean Returns \texttt{true} if all handlers in chain have been processed. Returns \texttt{false} if a handler in the chain returned \texttt{false} from its \texttt{handleResponse} method.

\textbf{Throws:} \\
\texttt{JAXRPCException} - if any processing error happens

\textbf{See Also:} \\
\texttt{Handler.handleResponse\,(javax.xml.rpc.handler.MessageContext)}

\textbf{public boolean handleFault\,(messageContext context)}

\textbf{handleFault} \texttt{SOAP}

\texttt{context \quad MessageContext \quad SOAP}

\texttt{return \quad boolean \quad true handleFault}

\textbf{Throws:} \texttt{JAXRPCException}:

\textbf{See also} \texttt{handleFault}

**handleFault**

boolean \texttt{handleFault\,(messageContext context)}

The \texttt{handleFault} method initiates the SOAP fault processing for this handler chain.

\textbf{Parameters:}

\texttt{context} - \texttt{MessageContext} parameter provides access to the SOAP message.

\textbf{Returns:}

boolean Returns \texttt{true} if all handlers in chain have been processed. Returns \texttt{false} if a handler in the chain returned \texttt{false} from its \texttt{handleFault} method.

\textbf{Throws:}

\texttt{JAXRPCException} - if any processing error happens

\textbf{See Also:} \\
\texttt{Handler.handleFault\,(javax.xml.rpc.handler.MessageContext)}

**public void init\,(java.util.Map<\texttt{K, V}> config)**
**HandlerChain**

`config`

**Throws**

JAXRPCException:

### init

```java
void init(Map config)
```

Initializes the configuration for a HandlerChain.

**Parameters:**

- `config`: Configuration for the initialization of this handler chain

**Throws:**

JAXRPCException - If any error during initialization

### destroy

```java
public void destroy()
```

Indicates the end of lifecycle for a HandlerChain.

**Throws:**

JAXRPCException - If any error during destroy

### setRoles(String[] soapActorNames)

```java
public void setRoles(String[] soapActorNames)
```
setRoles

void setRoles(String[] soapActorNames)

Sets SOAP Actor roles for this HandlerChain. This specifies the set of roles in which this HandlerChain is to act for the SOAP message processing at this SOAP node. These roles assumed by a HandlerChain must be invariant during the processing of an individual SOAP message through the HandlerChain.

A HandlerChain always acts in the role of the special SOAP actor next. Refer to the SOAP specification for the URI name for this special SOAP actor. There is no need to set this special role using this method.

Parameters:

soapActorNames - URIs for SOAP actor name

See Also:

NamespaceConstants

getRoles

String[] getRoles()

public String[] getRoles()

HandlerChain SOAP SOAP SOAP

return URI SOAP (String[])

See also

javax.xml.rpc.NamespaceConstants
Gets SOAP actor roles registered for this HandlerChain at this SOAP node. The returned array includes the special SOAP actor next.

**Returns:**
String[] SOAP Actor roles as URIs

**See Also:**
NamespaceConstants
The `javax.xml.rpc.handler.HandlerInfo` represents information about a handler in the HandlerChain. A HandlerInfo instance is passed in the `Handler.init` method to initialize a `Handler` instance.

**Version:**
1.0

**Author:**
Rahul Sharma

**See Also:**
- `HandlerChain`
- `Serialized Form`

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HandlerInfo()</code></td>
<td>Default constructor</td>
</tr>
</tbody>
</table>
**HandlerInfo**

*Class* handlerClass, *Map* config, *QName[]* headers

**Constructor for HandlerInfo**

### Method Summary

<table>
<thead>
<tr>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class</strong></td>
<td><strong>getHandlerClass()</strong>&lt;br&gt;Gets the Handler class</td>
</tr>
<tr>
<td><strong>Map</strong></td>
<td><strong>getHandlerConfig()</strong>&lt;br&gt;Gets the Handler configuration</td>
</tr>
<tr>
<td><strong>QName[]</strong></td>
<td><strong>getHeaders()</strong>&lt;br&gt;Gets the header blocks processed by this Handler.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setHandlerClass(Class handlerClass)</strong>&lt;br&gt;Sets the Handler class</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setHandlerConfig(Map config)</strong>&lt;br&gt;Sets the Handler configuration as java.util.Map</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setHeaders(QName[] headers)</strong>&lt;br&gt;Sets the header blocks processed by this Handler.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`  

### Constructor Detail

**public HandlerInfo()**

**HandlerInfo**

**public HandlerInfo()**

Default constructor
public HandlerInfo(Class<T> handlerClass, java.util.Map<K, V> config, javax.xml.namespace.QName[] headers)

**HandlerInfo**

```
  handlerClass  Handler  Java
  config       java.util.Map
  headers      Handler  QName QName
```

**Constructor for HandlerInfo**

**Parameters:**
- `handlerClass` - Java Class for the Handler
- `config` - Handler Configuration as a java.util.Map
- `headers` - QNames for the header blocks processed by this Handler. QName is the qualified name of the outermost element of a header block

**Method Detail**

public void setHandlerClass(Class<T> handlerClass)

**setHandlerClass**

```
  handlerClass  Handler
```

**Sets the Handler class**

**Parameters:**
public Class<T> getHandlerClass()

```
Handler
```

return Handler

null handler

`getHandlerClass`

Public `Class` `getHandlerClass()`

Gets the Handler class

**Returns:**

Returns null if no Handler class has been set; otherwise the set handler class

`setHandlerConfig`

Public `void` `setHandlerConfig(java.util.Map<K, V> config)`

```
java.util.Map Handler
```

config

`setHandlerConfig`

Public `void` `setHandlerConfig(Map config)`

Sets the Handler configuration as `java.util.Map`

**Parameters:**

config - Configuration map

`getHandlerConfig`

Public `java.util.Map<K, V> getHandlerConfig()`

```
Handler
```

return Map
getHandlerConfig

public Map getHandlerConfig()

Gets the Handler configuration

Returns:
  Returns empty Map if no configuration map has been set; otherwise returns the set configuration map

public void setHeaders(javax.xml.namespace.QName[] headers)

Sets the header blocks processed by this Handler.

Parameters:
  headers - QNames of the header blocks. QName is the qualified name of the outermost element of the SOAP header block

public javax.xml.namespace.QName[] getHeaders()

getHeaders

public QName[] getHeaders()
Gets the header blocks processed by this Handler.

**Returns:**
Array of QNames for the header blocks. Returns `null` if no header blocks have been set using the `setHeaders` method.
**javax.xml.rpc.handler**  
**Interface HandlerRegistry**

**All Superinterfaces:**  
[Serializable](#)

---

```java
public interface HandlerRegistry
extends Serializable

Implements: java.io.Serializable
```

The `javax.xml.rpc.handler.HandlerRegistry` provides support for the programmatic configuration of handlers in a `HandlerRegistry`.

A handler chain is registered per service endpoint, as indicated by the qualified name of a port. The `getHandlerChain` returns the handler chain (as a `java.util.List`) for the specified service endpoint. The returned handler chain is configured using the `java.util.List` interface. Each element in this list is required to be of the Java type `javax.xml.rpc.handler.HandlerInfo`.

**Version:**

1.0

**Author:**

Rahul Sharma

**See Also:**

[Service](#)
Method Summary

<table>
<thead>
<tr>
<th>List</th>
<th>getHandlerChain(QName portName)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the handler chain for the specified service endpoint.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th>setHandlerChain(QName portName, List chain)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sets the handler chain for the specified service endpoint as a java.util.List.</td>
</tr>
</tbody>
</table>

Method Detail

public java.util.List<E>
getHandlerChain(javax.xml.namespace.QName portName)
javax.xml.rpc.handler.HandlerInfo

portName

return java.util.List

Throws IllegalArgumentException: portName

getHandlerChain

List getHandlerChain(QName portName)

Gets the handler chain for the specified service endpoint. The returned List is used to configure this specific handler chain in this HandlerRegistry. Each element in this list is required to be of the Java type javax.xml.rpc.handler.HandlerInfo.

Parameters:

portName - Qualified name of the target service endpoint

Returns:

java.util.List Handler chain

Throws:

IllegalArgumentException - If an invalid portName is specified
public void setHandlerChain(javax.xml.namespace.QName portName, java.util.List<E> chain)

java.util.List Java

javax.xml.rpc.handler.HandlerInfo

portName

chain

Throws JAXRPCException:

Throws UnsupportedOperationException:

Throws IllegalArgumentException: portName

setHandlerChain

void setHandlerChain(QName portName, List chain)

Sets the handler chain for the specified service endpoint as a java.util.List. Each element in this list is required to be of the Java type javax.xml.rpc.handler.HandlerInfo.

Parameters:

portName - Qualified name of the target service endpoint

chain - A List representing configuration for the handler chain

Throws:

JAXRPCException - If any error in the configuration of the handler chain

UnsupportedOperationException - If this set operation is not supported. This is done to avoid any overriding of a pre-configured handler chain.

IllegalArgumentException - If an invalid portName is specified
to license terms.

PS:
HandlerResolver is an interface implemented by an application to get control over the handler chain set on proxy/dispatch objects at the time of their creation.

A HandlerResolver may be set on a Service using the setHandlerResolver method.

When the runtime invokes a HandlerResolver, it will pass it a PortInfo object containing information about the port that the proxy/dispatch object will be accessing.

Since: JAX-WS 2.0

See Also: 

Service.setHandlerResolver(javax.xml.ws.handler.HandlerResolver)

Method Summary

<table>
<thead>
<tr>
<th>List&lt;Handler&gt;</th>
<th>getHandlerChain(PortInfo portInfo)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the handler chain for the specified port.</td>
</tr>
</tbody>
</table>
public java.util.List<E> getHandlerChain(PortInfo portInfo)

getHandlerChain

List<Handler> getHandlerChain(PortInfo portInfo)

Gets the handler chain for the specified port.

Parameters:
portInfo - Contains information about the port being accessed.

Returns:
java.util.List Handler chain
javax.mail  **Class Header**

`java.lang.Object`  
- `javax.mail.Header`

**Direct Known Subclasses:**  
- `InternetHeaders.InternetHeader`

```java
public class Header
extends Object

Extended by: InternetHeaders.InternetHeader
```

Header /

The Header class stores a name/value pair to represent headers.

**Author:**  
John Mani

### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String</td>
<td>name</td>
<td>The name of the header.</td>
</tr>
<tr>
<td>protected String</td>
<td>value</td>
<td>The value of the header.</td>
</tr>
</tbody>
</table>

### Constructor Summary

```java
Header(String name, String value)
```

Construct a Header object.

### Method Summary

...
String **getName()**

Returns the name of this header.

String **getValue()**

Returns the value of this header.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

name

protected **String name**

The name of the header.

**Since:**

JavaMail 1.4

value

protected **String value**

The value of the header.

**Since:**

JavaMail 1.4

Constructor Detail
public Header(String name, String value)

Constructor.

Parameters:
  name - name of the header
  value - value of the header

Method Detail

public String getName()

return

getName

public String getName()

Returns the name of this header.

Returns:
  name of the header

public String getValue()

return
**getValue**

```java
public String getValue()
```

Returns the value of this header.

**Returns:**
value of the header
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS	NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES	NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
public final class HeaderTerm

extends StringTerm

Extends: SearchTerm > StringTerm

Message

This class implements comparisons for Message headers. The comparison is case-insensitive.

Author:
  Bill Shannon, John Mani
See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected String headerName</th>
</tr>
</thead>
<tbody>
<tr>
<td>The name of the header.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.mail.search.StringTerm
ignoreCase, pattern
Constructor Summary

**HeaderTerm**(*String* headerName, *String* pattern)

Constructor.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>equals(<em>Object</em> obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>String</td>
<td>getHeaderName()</td>
<td>Return the name of the header to compare with.</td>
</tr>
<tr>
<td>int</td>
<td>hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean</td>
<td>match(<em>Message</em> msg)</td>
<td>The header match method.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.search.*StringTerm*

getIgnoreCase, getPattern, match

Methods inherited from class java.lang.*Object*

clone, finalize, getClass, notify, notifyAll, toString, wait, wait

Field Detail

**headerName**

protected *String* headerName

The name of the header.
public HeaderTerm(String headerName, String pattern)

    headerName
    pattern

HeaderTerm

public HeaderTerm(String headerName, String pattern)

    Constructor.

    Parameters:
    headerName - The name of the header
    pattern - The pattern to search for

Method Detail

public String getHeaderName()

getHeaderName

public String getHeaderName()

    Return the name of the header to compare with.

public boolean match(Message msg)

    msg
    Message
    return true false
**match**

public boolean match(Message msg)

The header match method.

**Specified by:**

match in class SearchTerm

**Parameters:**

msg - The match is applied to this Message's header

**Returns:**

true if the match succeeds, otherwise false

---

**public boolean equals(Object obj)**

**equals**

public boolean equals(Object obj)

Equality comparison.

**Overrides:**

equals in class StringTerm

---

**public int hashCode()**

**hashCode**

public int hashCode()

Compute a hashCode for this object.

**Overrides:**
hashCode in class StringTerm
public class HeaderTokenizer
extends Object

Inner classes: HeaderTokenizer.Token

RFC822 MIME RFC822 MIME
CRLF SPACE

version 1.11, 07/05/04

This class tokenizes RFC822 and MIME headers into the basic symbols specified by RFC822 and MIME.

This class handles folded headers (ie headers with embedded CRLF SPACE sequences). The folds are removed in the returned tokens.

Version:
  1.11, 07/05/04
Author:
  John Mani

<table>
<thead>
<tr>
<th>Nested Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static class</td>
</tr>
<tr>
<td>HeaderTokenizer.Token</td>
</tr>
<tr>
<td>The Token class represents tokens returned by the HeaderTokenizer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
</table>
### Constructor Summary

**HeaderTokenizer(String** header)
Constructor.

**HeaderTokenizer(String** header, **String** delimiters)
Constructor.

**HeaderTokenizer(String** header, **String** delimiters, **boolean** skipComments)
Constructor that takes a rfc822 style header.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getRemainder()</strong></td>
<td>Return the rest of the Header.</td>
</tr>
<tr>
<td><strong>next()</strong></td>
<td>Parses the next token from this String.</td>
</tr>
<tr>
<td><strong>peek()</strong></td>
<td>Peek at the next token, without actually removing the token from the parse stream.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang. **Object**

**clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait**

### Field Detail

RFC822
public static final String RFC822

RFC822 specials

See Also:
Constant Field Values

MIME

public static final String MIME

MIME specials

See Also:
Constant Field Values

Constructor Detail

public HeaderTokenizer(String header, String delimiters, boolean skipComments)

rfc822

header

delimiters

ATOM

true

skipComments

RFC822

MI

HeaderTokenizer

public HeaderTokenizer(String header, String delimiters, boolean skipComments)

Constructor that takes a rfc822 style header.

Parameters:
The rfc822 header to be tokenized

delimiters - Set of delimiter characters to be used to delimit ATOMS. These are usually RFC822 or MIME

skipComments - If true, comments are skipped and not returned as tokens

public HeaderTokenizer(String header, String delimiters)

header

delimiters

HeaderTokenizer

public HeaderTokenizer(String header, String delimiters)

Constructor. Comments are ignored and not returned as tokens

Parameters:

header - The header that is tokenized
delimiters - The delimiters to be used

public HeaderTokenizer(String header)

RFC822 RFC822 ATOM

HeaderTokenizer

public HeaderTokenizer(String header)

Constructor. The RFC822 defined delimiters - RFC822 - are used to delimit ATOMS. Also comments are skipped and not returned as tokens
public HeaderTokenizer.Token next() throws ParseException
String

next()  EOF Token
    return  Token
    Throws  ParseException:

next

public HeaderTokenizer.Token next() throws ParseException

    Parses the next token from this String.

    Clients sit in a loop calling next() to parse successive tokens until an
    EOF Token is returned.

    Returns:
        the next Token

    Throws:
        ParseException - if the parse fails

public HeaderTokenizer.Token peek() throws ParseException

    return  Token
    Throws  ParseException:
**peek**

```java
public HeaderTokenizer.Token peek()
    throws ParseException
```

Peek at the next token, without actually removing the token from the parse stream. Invoking this method multiple times will return successive tokens, until `next()` is called.

**Returns:**
the next Token

**Throws:**
`ParseException` - if the parse fails

---

**public String getRemainder()**

```java
return (String) null
```

**getRemainder**

```java
public String getRemainder()
```

Return the rest of the Header.

**Returns:**
String rest of header. null is returned if we are already at end of header

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.mail.internet	Class HeaderTokenizer.Token

java.lang.Object

Enclosing class:

    HeaderTokenizer

public static class HeaderTokenizer.Token

extends Object

Contained within: HeaderTokenizer

Token HeaderTokenizer

The Token class represents tokens returned by the HeaderTokenizer.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int ATOM</td>
</tr>
<tr>
<td>static int COMMENT</td>
</tr>
<tr>
<td>static int EOF</td>
</tr>
<tr>
<td>static int QUOTEDSTRING</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeaderTokenizer.Token(int type, String value)</td>
</tr>
<tr>
<td>Constructor.</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td><code>getType()</code></td>
<td>Return the type of the token.</td>
</tr>
<tr>
<td>String</td>
<td><code>getValue()</code></td>
<td>Returns the value of the token just read.</td>
</tr>
</tbody>
</table>

Fields

**ATOM**

```java
public static final int ATOM
```

Token type indicating an ATOM.

See Also:

Constant Field Values

**QUOTEDSTRING**

```java
public static final int QUOTEDSTRING
```

Token type indicating a quoted string. The value field contains the string without the quotes.

See Also:

Constant Field Values
public static final int COMMENT

Token type indicating a comment. The value field contains the comment string without the comment start and end symbols.

See Also:
Constant Field Values

---

public static final int EOF

Token type indicating end of input.

See Also:
Constant Field Values

### Constructor Detail

public HeaderTokenizer.Token(int type, String value)

<table>
<thead>
<tr>
<th>type</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Token</td>
</tr>
</tbody>
</table>

### HeaderTokenizer.Token

public HeaderTokenizer.Token(int type, String value)

Constructor.
Parameters:
  type - Token type
  value - Token value

Method Detail

public int getType()

  • ATOM ASCII CTL"("<>"
  • QUOTEDSTRING ASCII
  • COMMENT "(" ")" ASCII
  • EOF

getType

public int getType()

  Return the type of the token. If the token represents a delimiter or a control character, the type is that character itself, converted to an integer. Otherwise, it's value is one of the following:
  • ATOM A sequence of ASCII characters delimited by either SPACE, CTL, "(" <= " or the specified SPECIALS
  • QUOTEDSTRING A sequence of ASCII characters within quotes
  • COMMENT A sequence of ASCII characters within "(" and ")".
  • EOF End of header

public String getValue()

  return

getValue
public String getValue()

Returns the value of the token just read. When the current token is a quoted string, this field contains the body of the string, without the quotes. When the current token is a comment, this field contains the body of the comment.

Returns:

token value
javax.transaction  **Class HeuristicCommitException**

`java.lang.Object`  
`   ↓ java.lang.Throwable`  
`      ↓ java.lang.Exception`  
`         ↓ javax.transaction.HeuristicCommitException`  

**All Implemented Interfaces:**  
  * Serializable*

---

```java
public class HeuristicCommitException

extends Exception

Extends: Throwable > Exception
```

This exception is thrown by the rollback operation on a resource to report that a heuristic decision was made and that all relevant updates have been committed.

**See Also:**  
  * Serialized Form*

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HeuristicCommitException()</code></td>
<td></td>
</tr>
<tr>
<td><code>HeuristicCommitException(String msg)</code></td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary
Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public HeuristicCommitException()

HeuristicCommitException

public HeuristicCommitException()

public HeuristicCommitException(String msg)

HeuristicCommitException

public HeuristicCommitException(String msg)
PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS	| NEXT CLASS	| SUMMARY: NESTED | FIELD | CONSTR | METHOD |
FRAMES	| NO FRAMES	| DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD |
FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
Class HeuristicMixedException

java.lang.Object
   └ java.lang.Throwable
      └ java.lang.Exception
         └ javax.transaction.HeuristicMixedException

All Implemented Interfaces:
   Serializable

public class HeuristicMixedException
   extends Exception

Extends: Throwable > Exception

This exception is thrown to report that a heuristic decision was made and that some relevant updates have been committed and others have been rolled back.

See Also:
   Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeuristicMixedException()</td>
</tr>
<tr>
<td>HeuristicMixedException(String msg)</td>
</tr>
</tbody>
</table>

---

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
</table>
Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

class HeuristicMixedException

public HeuristicMixedException()

public HeuristicMixedException()

public HeuristicMixedException(String msg)

public HeuristicMixedException(String msg)
PS:
javax.transaction Class HeuristicRollbackException

java.lang.Object
   ↘ java.lang.Throwable
      ↘ java.lang.Exception
         ↘ javax.transaction.HeuristicRollbackException

All Implemented Interfaces:
   Serializable

public class HeuristicRollbackException
  extends Exception

Extends: Throwable > Exception

This exception is thrown by the commit operation to report that a heuristic
decision was made and that all relevant updates have been rolled back.

See Also:
   Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeuristicRollbackException()</td>
</tr>
<tr>
<td>HeuristicRollbackException(String</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Throwable</td>
</tr>
</tbody>
</table>
fillInStackTrace, get Cause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

**public HeuristicRollbackException()**

**HeuristicRollbackException**

**public HeuristicRollbackException()**

**public HeuristicRollbackException(String msg)**

**HeuristicRollbackException**

**public HeuristicRollbackException(String msg)**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public final class HexBinaryAdapter
extends XmlAdapter<String, byte[]>

Extends: ">XmlAdapter

xs:hexBinary XmlAdapter

XML hexBinary since JAXB 2.0

This XmlAdapter binds byte[] to the hexBinary representation in XML.

Since:
JAXB 2.0
Author:
Kohsuke Kawaguchi

### Constructor Summary

| HexBinaryAdapter() |

### Method Summary

| String marshal(byte[] bytes) |

Convert a bound type to a value type.
Constructor Detail

public HexBinaryAdapter()

HexBinaryAdapter

public HexBinaryAdapter()

Method Detail

public byte[] unmarshal(String s)

unmarshal

public byte[] unmarshal(String s)

Description copied from class: XmlAdapter
Convert a value type to a bound type.

Specified by:
unmarshal in class XmlAdapter<String, byte[]>

Parameters:
 s - The value to be converted. Can be null.

public String marshal(byte[] bytes)
public String marshal(byte[] bytes)

Description copied from class: XmlAdapter
Convert a bound type to a value type.

Specified by:
    marshal in class XmlAdapter<String,byte[]>

Parameters:
    bytes - The value to be converted. Can be null.
javax.xml.rpc.holders Interface Holder

All Known Implementing Classes:

BigDecimalHolder, BigIntegerHolder, BooleanHolder,
BooleanWrapperHolder, ByteArrayHolder, ByteHolder,
ByteWrapperHolder, CalendarHolder, DoubleHolder,
DoubleWrapperHolder, FloatHolder, FloatWrapperHolder,
IntegerWrapperHolder, IntHolder, LongHolder, LongWrapperHolder,
ObjectHolder, QNameHolder, ShortHolder, ShortWrapperHolder,
StringHolder

java.xml.rpc.holders.Holder Holder Holder Holder Holder

version 1.0

The java.xml.rpc.holders.Holder interface represents the base interface for both standard and generated Holder classes. A generated Holder class is required to implement this Holder interface.

Version: 1.0

Author: Rahul Sharma
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws Class Holder<T>

java.lang.Object
   javax.xml.ws.Holder<T>

public final class Holder<T>
extends Object

Implemented by: BigDecimalHolder, BigIntegerHolder, BooleanHolder, BooleanWrapperHolder, ByteArrayHolder, ByteHolder, ByteWrapperHolder, CalendarHolder, DoubleHolder, DoubleWrapperHolder, FloatHolder, FloatWrapperHolder, IntegerWrapperHolder, IntHolder, LongHolder, LongWrapperHolder, ObjectHolder, QNameHolder, ShortHolder, ShortWrapperHolder, StringHolder

java.xml.rpc.holders.Holder

version 1.0

Holds a value of type T.

Since:
   JAX-WS 2.0

Field Summary

| T value | The value contained in the holder. |

Constructor Summary

Holder() Creates a new holder with a null value.
**Holder**(T value)
Create a new holder with the specified value.

**Method Summary**

Methods inherited from class java.lang.[Object](#)
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Field Detail**

value

public T value

The value contained in the holder.

**Constructor Detail**

Holder

public Holder()

Creates a new holder with a null value.

---

Holder

public Holder(T value)
Create a new holder with the specified value.

**Parameters:**

value - The value to be stored in the holder.
javax.ejb

Interface HomeHandle

All Superinterfaces:
  Serializable

public interface HomeHandle
  extends Serializable

  Implements: java.io.Serializable

HomeHandle  home  home  home  ""

The HomeHandle interface is implemented by all home object handles. A handle is an abstraction of a network reference to a home object. A handle is intended to be used as a "robust" persistent reference to a home object.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getEJBHome()</td>
<td>Obtain the home object represented by this handle.</td>
</tr>
</tbody>
</table>

Method Detail

public EJBHome getEJBHome() throws java.rmi.RemoteException

  return  home

  Throws  java.rmi.RemoteException: home
getEJBHome

EJBHome  getEJBHome()  throws RemoteException

Obtain the home object represented by this handle.

Returns:
the home object represented by this handle.

Throws:
RemoteException - The home object could not be obtained because of a system-level failure.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html Class HtmlColumn

java.lang.Object
  ▼ javax.faces.component.UIComponent
      ▼ javax.faces.component.UIComponentBase
          ▼ javax.faces.component.UIColumn
              ▼ javax.faces.component.html.HtmlColumn

All Implemented Interfaces:
  StateHolder

public class HtmlColumn
  extends UIComponent

Extends: UIComponent > UIColumn

HTML table

Represents a column that will be rendered in an HTML table element.

Field Summary

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIComponent

COMPONENT_FAMILY

Fields inherited from class javax.faces.component.UIComponent

bindings

## Constructor Summary

<table>
<thead>
<tr>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>HtmlColumn()</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getFooterClass()</td>
<td>Return the value of the footerClass property.</td>
</tr>
<tr>
<td>String getHeaderClass()</td>
<td>Return the value of the headerClass property.</td>
</tr>
<tr>
<td>void restoreState(FacesContext _context, Object _state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object saveState(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void setFooterClass(String footerClass)</td>
<td>Set the value of the footerClass property.</td>
</tr>
<tr>
<td>void setHeaderClass(String headerClass)</td>
<td>Set the value of the headerClass property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UiColumn`:
- getFamily, getFooter, getHeader, setFooter, setHeader

Methods inherited from class `javax.faces.component.UIClientComponentBase`:
- addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, get FAC etCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding
Methods inherited from class javax.faces.component.UIComponent
- encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
 Constant Field Values

Constructor Detail

public HtmlColumnType

Method Detail

public String getFooterClass()
getFooterClass

public String getFooterClass()

Return the value of the footerClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any footer generated for this column.

public void setFooterClass(String footerClass)

setFooterClass

public void setFooterClass(String footerClass)

Set the value of the footerClass property.

public String getHeaderClass()

headerClass

CSS
**getHeaderClass**

public **String** getHeaderClass()

Return the value of the `headerClass` property.

Contents: Space-separated list of CSS style class(es) that will be applied to any header generated for this column.

---

**public void setHeaderClass(String headerClass)**

`headerClass`

---

**setHeaderClass**

public void **setHeaderClass(String headerClass)**

Set the value of the `headerClass` property.

---

**public Object saveState(FacesContext _context)**

**saveState**

public **Object** saveState( **FacesContext _context**)

Description copied from interface: **StateHolder**

Gets the state of the instance as a `Serializable` Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the**
state of children and facets. That is done via the `StateManager`

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**
`saveState` in interface `StateHolder`

**Overrides:**
`saveState` in class `UIComponentBase`

---

```java
public void restoreState(FacesContext _context, Object _state)
```

**restoreState**

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface: `StateHolder`**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement `StateHolder` (such as a `UIComponent` with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
`restoreState` in interface `StateHolder`
Overrides:

restoreState in class UIComponentBase
javax.faces.component.html Class HtmlCommandButton

java.lang.Object
  ▼ javax.faces.component.UIComponent
     ▼ javax.faces.component.UIComponentBase
        ▼ javax.faces.component.UICommand
           ▼ javax.faces.component.html.HtmlCommandButton

All Implemented Interfaces:
   ActionSource, ActionSource2, StateHolder

public class HtmlCommandButton
  extends UICommand

Extends: UIComponent > UIComponentBase > UICommand

submit  reset  HTML  input

    rendererType "javax.faces.Button"  setRenderType()

Represents an HTML input element for a button of type submit or reset. The label text is specified by the component value.

By default, the rendererType property must be set to "javax.faces.Button". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
</tr>
<tr>
<td></td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UICommand
### Fields inherited from class `javax.faces.component.UIComponent` bindings

### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HtmlCommandButton()</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAccesskey()</code></td>
<td>String ()</td>
<td>Return the value of the <code>accesskey</code> property.</td>
</tr>
<tr>
<td><code>getAlt()</code></td>
<td>String ()</td>
<td>Return the value of the <code>alt</code> property.</td>
</tr>
<tr>
<td><code>getDir()</code></td>
<td>String ()</td>
<td>Return the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>getImage()</code></td>
<td>String ()</td>
<td>Return the value of the <code>image</code> property.</td>
</tr>
<tr>
<td><code>getLabel()</code></td>
<td>String ()</td>
<td>Return the value of the <code>label</code> property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>String ()</td>
<td>Return the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>getOnblur()</code></td>
<td>String ()</td>
<td>Return the value of the <code>onblur</code> property.</td>
</tr>
<tr>
<td><code>getOnchange()</code></td>
<td>String ()</td>
<td>Return the value of the <code>onchange</code> property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>String ()</td>
<td>Return the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>String ()</td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>String ()</td>
<td>Return the value of the <code>onfocus</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>String ()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the <code>onselect</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Return the value of the <code>type</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>isDisabled()</code></td>
<td>Return the value of the <code>disabled</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>isReadonly()</code></td>
<td>Return the value of the <code>readonly</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>--------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Object</strong> <strong>saveState</strong>(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setAccesskey</strong>(String accesskey)</td>
<td>Set the value of the accesskey property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setAlt</strong>(String alt)</td>
<td>Set the value of the alt property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setDir</strong>(String dir)</td>
<td>Set the value of the dir property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setDisabled</strong>(boolean disabled)</td>
<td>Set the value of the disabled property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setImage</strong>(String image)</td>
<td>Set the value of the image property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setLabel</strong>(String label)</td>
<td>Set the value of the label property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setLang</strong>(String lang)</td>
<td>Set the value of the lang property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnblur</strong>(String onblur)</td>
<td>Set the value of the onblur property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnchange</strong>(String onchange)</td>
<td>Set the value of the onchange property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnclick</strong>(String onclick)</td>
<td>Set the value of the onclick property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOndblclick</strong>(String ondblclick)</td>
<td>Set the value of the ondblclick property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnfocus</strong>(String onfocus)</td>
<td>Set the value of the onfocus property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnkeydown</strong>(String onkeydown)</td>
<td>Set the value of the onkeydown property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnkeypress</strong>(String onkeypress)</td>
<td>Set the value of the onkeypress property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnkeyup</strong>(String onkeyup)</td>
<td>Set the value of the onkeyup property.</td>
<td></td>
</tr>
<tr>
<td>void <strong>setOnmousedown</strong>(String onmousedown)</td>
<td>Set the value of the onmousedown property.</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Signature</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setOnmousemove</strong>(String onmousemove)</td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setOnmouseout</strong>(String onmouseout)</td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setOnmouseover</strong>(String onmouseover)</td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setOnmouseup</strong>(String onmouseup)</td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setOnselect</strong>(String onselect)</td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setReadonly</strong>(boolean readonly)</td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setStyle</strong>(String style)</td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setStyleClass</strong>(String styleClass)</td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setTabindex</strong>(String tabindex)</td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setTitle</strong>(String title)</td>
<td>Set the value of the title property.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setType</strong>(String type)</td>
<td>Set the value of the type property.</td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.faces.component.UICommand**

- addActionListener, broadcast, getAction, getActionExpression, getActionListener, getActionListeners, getFamily, getValue, isImmediate, queueEvent, removeActionListener, setAction, setActionExpression, setActionListener, setImmediate, setValue

**Methods inherited from class javax.faces.component.UIComponentBase**

- addFacesListener, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes
### Field Detail

**COMPONENT_TYPE**

```java
enum COMPONENT_TYPE {
    ...
}
```

The standard component type for this component.

**See Also:**
- [Constant Field Values](#)

### Constructor Detail

**public HtmlCommandButton()**

**HtmlCommandButton**

**public HtmlCommandButton()**
getAccesskey

public String getAccesskey()

accesskey

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

setAccesskey

public void setAccesskey(String accesskey)

accesskey

Set the value of the accesskey property.
getAlt

public String getAlt()

    Return the value of the alt property.

    Contents: Alternate textual description of the element rendered by this component.

setAlt

public void setAlt(String alt)

    alt

getDir

public String getDir()

    dir

"LTR" left-to-right "RTL" right-to-left

gDir

public String getDir()
Return the value of the \texttt{dir} property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

\begin{verbatim}
public void setDir(String dir)
    dir

setDir
public void setDir(String dir)
    Set the value of the \texttt{dir} property.
\end{verbatim}

\begin{verbatim}
public boolean isDisabled()
    disabled

false true disabled="disabled"

isDisabled
public boolean isDisabled()
    Return the value of the \texttt{disabled} property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".
\end{verbatim}
public void setDisabled(boolean disabled)

disabled

setDisabled

public void setDisabled(boolean disabled)

Set the value of the disabled property.

g getImage

public String getImage()

image

URL URL "input" "image"
"type" "value"

g getImage

public String getImage()

Return the value of the image property.

Contents: Absolute or relative URL of the image to be displayed for this button. If specified, this "input" element will be of type "image". Otherwise, it will be of the type specified by the "type" property with a label specified by the "value" property.

public void setImage(String image)
public void setImage(String image)

Set the value of the image property.

public String getLabel()

label

public String getLabel()

Return the value of the label property.

Contents: A localized user presentable name for this component.

public void setLabel(String label)

label

public void setLabel(String label)
Set the value of the `label` property.

```java
public String getLang()
{
    lang
}
```

**getLang**

```java
public String getLang()
{
    return the value of the `lang` property.
    Contents: Code describing the language used in the generated markup for this component.
}
```

```java
public void setLang(String lang)
{
    lang
}
```

**setLang**

```java
public void setLang(String lang)
{
    Set the value of the `lang` property.
}
```

```java
public String getOnblur()
{
    onblur
}
```
getOnblur

public String getOnblur()

Return the value of the onblur property.

Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

onblur

setOnblur

public void setOnblur(String onblur)

Set the value of the onblur property.

public String getOnchange()

onchange

Javascript

getOnchange

public String getOnchange()
Return the value of the `onchange` property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

```java
public void setOnchange(String onchange)
```

`onchange`

---

```java
public void setOnchange(String onchange)
```

Set the value of the `onchange` property.

---

```java
public String getOnclick()
```

`onclick`

**Javascript**

```java
public String getOnclick()
```

Return the value of the `onclick` property.

Contents: Javascript code executed when a pointer button is clicked over this element.

---

```java
public void setOnclick(String onclick)
```
onclick

setOnonclick

public void setOnonclick(String onclick)

    Set the value of the onclick property.

public String getOndblclick()

    ondblclick

Javascript

getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.

    Contents: Javascript code executed when a pointer button is double clicked over this element.

public void setOndblclick(String ondblclick)

    ondblclick

setOndblclick

public void setOndblclick(String ondblclick)
Set the value of the ondblclick property.

public String getOnfocus()

 побсct

getOnfocus

public String getOnfocus()

Return the value of the onfocus property.

Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onfocus)

 побсct

setOnfocus

public void setOnfocus(String onfocus)

Set the value of the onfocus property.

public String getOnkeydown()

 побcдк drawbacks
Javascript

getonkeydown

class Javascript:

public String getOnkeydown()

    Return the value of the onkeydown property.

    Contents: Javascript code executed when a key is pressed down over this element.

-----------------------------------------------

public void setOnkeydown(String onkeydown)

    onkeydown

setOnkeydown

class Javascript:

public void setOnkeydown(String onkeydown)

    Set the value of the onkeydown property.

-----------------------------------------------

public String getOnkeypress()

    onkeypress

Javascript

getonkeypress

class Javascript:

public String getOnkeypress()

    getOnkeypress()
Return the value of the `onkeypress` property.

Contents: Javascript code executed when a key is pressed and released over this element.

```java
public void setOnkeypress(String onkeypress)

    onkeypress

setOnkeypress
```

```java
public void setOnkeypress(String onkeypress)

    Set the value of the `onkeypress` property.

public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup
```

```java
public String getOnkeyup()

    Return the value of the `onkeyup` property.

    Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)
onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.

public String getOnmousedown()

    onmousedown

Javascript

getOnmousedown

public String getOnmousedown()

    Return the value of the onmousedown property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.

public void setOnmousedown(String onmousedown)

    onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)
Set the value of the `onmousedown` property.

```java
public String getOnmousemove()

  onmousemove

Javascript

getOnmousemove
```

Public String getOnmousemove()

  Return the value of the `onmousemove` property.

  Contents: Javascript code executed when a pointer button is moved within this element.

```
public void setOnmousemove(String onmousemove)

  onmousemove

setOnmousemove
```

Public void setOnmousemove(String onmousemove)

  Set the value of the `onmousemove` property.

```
public String getOnmouseout()

  onmouseout
```
getOnmouseout

public String getOnmouseout()

   Return the value of the onmouseout property.

   Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

   onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

   Set the value of the onmouseout property.

public String getOnmouseover()

   onmouseover

getOnmouseover

public String getOnmouseover()
Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.

```
public void setOnmouseover(String onmouseover)
```

```
onmouseover
```

```
setOnmouseover
```

```
public void setOnmouseover(String onmouseover)
```

```
Set the value of the onmouseover property.
```

```
public String getOnmouseup()
```

```
onmouseup
```

```
Javascript
```

```
getOnmouseup
```

```
public String getOnmouseup()
```

```
Return the value of the onmouseup property.

Contents: Javascript code executed when a pointer button is released over this element.
```

```
public void setOnmouseup(String onmouseup)
```
setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

getOnselect

public String getOnselect()

    onSelect

Javascript

getOnselect

public String getOnselect()

    Return the value of the onSelect property.

    Contents: Javascript code executed when text within this element is selected by the user.

setOnselect

public void setOnselect(String onSelect)

    onSelect

setOnselect

public void setOnselect(String onSelect)
Set the value of the `onselect` property.

```java
public boolean isReadonly()
    {readonly
false true readonly="readonly"

isReadonly

public boolean isReadonly()
    Return the value of the `readonly` property.
    Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `readonly="readonly"`.

public void setReadonly(boolean readonly)
    {readonly

setReadonly

public void setReadonly(boolean readonly)
    Set the value of the `readonly` property.
public String getStyle()

    style

    CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style

setStyle

public void setStyle(String style)

    Set the value of the style property.

public String getStyleClass()

    styleClass

    CSS "class"
**getStyleClass**

public String getStyleClass()

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

**public void setStyleClass(String styleClass)**

`styleClass`

**setStyleClass**

public void setStyleClass(String `styleClass`)

Set the value of the `styleClass` property.

**public String getTabindex()**

`tabindex`

Tab 0 32767

**getTabindex**

public String getTabindex()
Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

```java
public void setTabindex(String tabindex)

setTabindex

public void setTabindex(String tabindex)

Set the value of the `tabindex` property.
```

```java
public String getTitle()

title

g getTitle

public String getTitle()

Return the value of the `title` property.

Contents: Advisory title information about markup elements generated for this component.

```java
public void setTitle(String title)
setTitle

public void setTitle(String title)

Set the value of the title property.

public String getType()

type

"submit" "reset" "submit"

getType

public String getType()

Return the value of the type property.

Contents: Type of button to create. Valid values are "submit" and "reset". If not specified, or not a valid value, the default value is "submit".

public void setType(String type)

type

setType
public void setType(String type)

Set the value of the type property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
saveState in interface StateHolder

Overrides:
saveState in class UICommand

public void restoreState(FacesContext _context, Object
`_state)``

**restoreState**

```java
public void restoreState(FacesContext _context,
                         Object _state)
```

Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**

`restoreState` in interface **StateHolder**

**Overrides:**

`restoreState` in class **UICommand**

--

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html Class HtmlCommandLink

does not inherit from a class

public class HtmlCommandLink
extends UICommand

Extends: UIComponent > UIComponentBase > UICommand

HTML a JavaScript

\[
\text{rendererType } " \text{javax.faces.Link}" \text{ setRendererType()}
\]

Represents an HTML a element for a hyperlink that acts like a submit button. This component must be placed inside a form, and requires JavaScript to be enabled in the client.

By default, the rendererType property must be set to "javax.faces.Link". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UICommand
### Fields inherited from class javax.faces.component.UIComponent bindings

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Htm1CommandLink()</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getAccesskey()</td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td>String getCharset()</td>
<td>Return the value of the charset property.</td>
</tr>
<tr>
<td>String getCoords()</td>
<td>Return the value of the coords property.</td>
</tr>
<tr>
<td>String getDir()</td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td>String getHreflang()</td>
<td>Return the value of the hreflang property.</td>
</tr>
<tr>
<td>String getLang()</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String getOnblur()</td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>String getOnclick()</td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td>String getOndblclick()</td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td>String getOnfocus()</td>
<td>Return the value of the onfocus property.</td>
</tr>
<tr>
<td>String getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String getOnkeypress()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>getOnkeypress()</td>
<td>Return the value of the onkeypress property.</td>
</tr>
<tr>
<td>getOnkeyup()</td>
<td>Return the value of the onkeyup property.</td>
</tr>
<tr>
<td>getOnmousedown()</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>getOnmousemove()</td>
<td>Return the value of the onmousemove property.</td>
</tr>
<tr>
<td>getOnmouseout()</td>
<td>Return the value of the onmouseout property.</td>
</tr>
<tr>
<td>getOnmouseover()</td>
<td>Return the value of the onmouseover property.</td>
</tr>
<tr>
<td>getOnmouseup()</td>
<td>Return the value of the onmouseup property.</td>
</tr>
<tr>
<td>getRel()</td>
<td>Return the value of the rel property.</td>
</tr>
<tr>
<td>getRev()</td>
<td>Return the value of the rev property.</td>
</tr>
<tr>
<td>getShape()</td>
<td>Return the value of the shape property.</td>
</tr>
<tr>
<td>getStyle()</td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>getStyleClass()</td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>getTabIndex()</td>
<td>Return the value of the tabindex property.</td>
</tr>
<tr>
<td>getTarget()</td>
<td>Return the value of the target property.</td>
</tr>
<tr>
<td>getTitle()</td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td>getType()</td>
<td>Return the value of the type property.</td>
</tr>
<tr>
<td>isDisabled()</td>
<td>Return the value of the disabled property.</td>
</tr>
<tr>
<td>restoreState(FacesContext _context, Object _state)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>setAccesskey(String accesskey)</code></td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td><code>setCharset(String charset)</code></td>
<td>Set the value of the charset property.</td>
</tr>
<tr>
<td><code>setCoords(String coords)</code></td>
<td>Set the value of the coords property.</td>
</tr>
<tr>
<td><code>setDir(String dir)</code></td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td><code>setDisabled(boolean disabled)</code></td>
<td>Set the value of the disabled property.</td>
</tr>
<tr>
<td><code>setHreflang(String hreflang)</code></td>
<td>Set the value of the hreflang property.</td>
</tr>
<tr>
<td><code>setLang(String lang)</code></td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td><code>setOnblur(String onblur)</code></td>
<td>Set the value of the onblur property.</td>
</tr>
<tr>
<td><code>setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td><code>setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>void setOnmousemove(String onmousemove)</td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td>void setOnmouseout(String onmouseout)</td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td>void setOnmouseover(String onmouseover)</td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td>void setOnmouseup(String onmouseup)</td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td>void setRel(String rel)</td>
<td>Set the value of the rel property.</td>
</tr>
<tr>
<td>void setRev(String rev)</td>
<td>Set the value of the rev property.</td>
</tr>
<tr>
<td>void setShape(String shape)</td>
<td>Set the value of the shape property.</td>
</tr>
<tr>
<td>void setStyle(String style)</td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>void setStyleClass(String styleClass)</td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td>void setTabIndex(String tabindex)</td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td>void setTitle(String title)</td>
<td>Set the value of the title property.</td>
</tr>
<tr>
<td>void setType(String type)</td>
<td>Set the value of the type property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UICommand`
- addActionListener, broadcast, getAction, getActionExpression, getActionListener, getActionListeners, getFamily, getValue, isImmediate, queueEvent, removeActionListener, setAction, setActionExpression, setActionListener, setImmediate, setValue

Methods inherited from class `javax.faces.component.UIComponentBase`
Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE
public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail

public HtmlCommandLink()
public HtmlCommandLink()

### Method Detail

**getAccesskey**

```java
public String getAccesskey()
```

**accesskey**

**getAccesskey**

```java
public String getAccesskey()
```

Return the value of the `accesskey` property.

Contents: Access key that, when pressed, transfers focus to this element.

**setAccesskey**

```java
public void setAccesskey(String accesskey)
```

**accesskey**

**setAccesskey**

```java
public void setAccesskey(String accesskey)
```

Set the value of the `accesskey` property.
public String getCharset()

    charset

getCharset

public String getCharset()

    Return the value of the charset property.

    Contents: The character encoding of the resource designated by this hyperlink.

public void setCharset(String charset)

    charset

setCharset

public void setCharset(String charset)

    Set the value of the charset property.

public String getCoords()

    coords
getCoords

public String getCoords()

    Return the value of the coords property.

    Contents: The position and shape of the hot spot on the screen (for use in client-side image maps).


public void setCoords(String coords)

coords

setCoords

public void setCoords(String coords)

    Set the value of the coords property.


public String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

    Return the value of the dir property.
Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

```java
public void setDir(String dir)
```

`dir`

setDir

```java
public void setDir(String dir)
```

Set the value of the `dir` property.

```java
public boolean isDisabled()
```

`disabled`

isDisabled

```java
public boolean isDisabled()
```

Return the value of the `disabled` property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit.

```java
public void setDisabled(boolean disabled)
```
public void setDisabled(boolean disabled)

Set the value of the disabled property.

public String getHreflang()

Return the value of the hreflang property.

Contents: The language code of the resource designated by this hyperlink.

public void setHreflang(String hreflang)

public void setHreflang(String hreflang)
Set the value of the `hreflang` property.

```java
public String getLang()
```

`lang`

**getLang**

```java
public String getLang()
```

Return the value of the `lang` property.

Contents: Code describing the language used in the generated markup for this component.

```java
public void setLang(String lang)
```

`lang`

**setLang**

```java
public void setLang(String lang)
```

Set the value of the `lang` property.

```java
public String getOnblur()
```

`onblur`
getOnblur

```java
public String getOnblur()
```

Return the value of the `onblur` property.

Contents: Javascript code executed when this element loses focus.

---

public void setOnblur(String onblur)

```java
onblur
```

---

setOnblur

```java
public void setOnblur(String onblur)
```

Set the value of the `onblur` property.

---

public String getOnclick()

```java
onclick
```

---

getOnclick

```java
public String getOnclick()
```
Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.

```java
public void setOnclick(String onclick)
```

`onclick`

---

`setOnclick`

```java
public void setOnclick(String onclick)
```

Set the value of the onclick property.

---

`getOndblclick`

```java
public String getOndblclick()
```

`ondblclick`

Javascript

---

`getOndblclick`

```java
public String getOndblclick()
```

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

```java
public void setOndblclick(String ondblclick)
```
**setOndblclick**

```
public void setOndblclick(String ondblclick)
```

Set the value of the ondblclick property.

**getOnfocus**

```
public String getOnfocus()
```

**getOnfocus**

```
public String getOnfocus()
```

Return the value of the onfocus property.

Contents: Javascript code executed when this element receives focus.

**setOnfocus(String onfocus)**

```
public void setOnfocus(String onfocus)
```

**setOnfocus**

```
public void setOnfocus(String onfocus)
```
Set the value of the `onfocus` property.

```java
public String getOnkeydown()
```

`onkeydown`

**Javascript**

**getOnkeydown**

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

```java
public void setOnkeydown(String onkeydown)
```

`onkeydown`

**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.

```java
public String getOnkeypress()
```

`onkeypress`
Javascript

getOnkeypress

public String getOnkeypress()

    Return the value of the onkeypress property.

    Contents: Javascript code executed when a key is pressed and released over this element.

    public void setOnkeypress(String onkeypress)

        onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

    Set the value of the onkeypress property.

public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup

public String getOnkeyup()
Return the value of the `onkeyup` property.

Contents: Javascript code executed when a key is released over this element.

---

```java
public void setOnkeyup(String onkeyup)

    onkeyup
```

---

`setOnkeyup`

```java
public void setOnkeyup(String onkeyup)

    Set the value of the `onkeyup` property.
```

---

```java
public String getOnmousedown()

    onmousedown
```

---

`getOnmousedown`

```java
public String getOnmousedown()

    Return the value of the `onmousedown` property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.
```

---

```java
public void setOnmousedown(String onmousedown)
```

onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

Set the value of the onmousedown property.

public String getOnmousemove()

onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

Return the value of the onmousemove property.

Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

onmousemove

setOnmousemove

public void setOnmousemove(String onmousemove)
Set the value of the onmousemove property.

```
public String getOnmouseout()
```

```
onmouseout
```

Javascript

```
public String getOnmouseout()
```

```
Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.

```
public void setOnmouseout(String onmouseout)
```

```
onmouseout
```

```
setOnmouseout
```

```
public void setOnmouseout(String onmouseout)
```

```
Set the value of the onmouseout property.

```
public String getOnmouseover()
```

```
onmouseover
```
Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

----------------------------------------

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

----------------------------------------

public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public String getOnmouseup()
Return the value of the `onmouseup` property.

Contents: Javascript code executed when a pointer button is released over this element.

---

```java
public void setOnmouseup(String onmouseup)
```

`onmouseup`

---

`setOnmouseup`

```java
public void setOnmouseup(String onmouseup)
```

Set the value of the `onmouseup` property.

---

```java
public String getRel()
```

`rel`

---

`getRel`

```java
public String getRel()
```

Return the value of the `rel` property.

Contents: The relationship from the current document to the anchor specified by this hyperlink. The value of this attribute is a space-separated list of link types.
public void setRel(String rel)

    rel

setRel

public void setRel(String rel)

    Set the value of the rel property.

public String getRev()

    rev

getRev

public String getRev()

    Return the value of the rev property.

    Contents: A reverse link from the anchor specified by this hyperlink to the current document. The value of this attribute is a space-separated list of link types.

public void setRev(String rev)

    rev
**setRev**

```java
public void setRev(String rev)
```

Set the value of the `rev` property.

**public String getShape()**

```java
shape
defaultrectcircle
poly
```

**getShape**

```java
public String getShape()
```

Return the value of the `shape` property.

Contents: The shape of the hot spot on the screen (for use in client-side image maps). Valid values are: default (entire region); rect (rectangular region); circle (circular region); and poly (polygonal region).

**public void setShape(String shape)**

```java
shape
```

**setShape**

```java
public void setShape(String shape)
```
public String getStyle()

style

CSS

getStyle

public String getStyle()

Return the value of the style property.

Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

style

setStyle

public void setStyle(String style)

Set the value of the style property.

public String getStyleClass()

styleClass
CSS "class"

getStyleClass

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

```java
public void setStyleClass(String styleClass)
```

```java
styleClass
```

setStyleClass

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.

```java
public String getTabindex()
```

```java
tabindex
```

Tab 0 32767

getTabindex
public String getTabindex()

    Return the value of the tabindex property.

    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

public void setTabindex(String tabindex)

    tabindex

setTabindex

public void setTabindex(String tabindex)

    Set the value of the tabindex property.

garget

public String getTarget()

    target

getTarget

public String getTarget()

    Return the value of the target property.

    Contents: Name of a frame where the resource retrieved via this hyperlink is to be displayed.
public void setTarget(String target)

target

setTarget

public void setTarget(String target)

Set the value of the target property.

public String getTitle()

title

ggetTitle

public String getTitle()

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

public void setTitle(String title)

setTitle
setTitle

public void setTitle(String title)

Set the value of the title property.

------------------

getType

public String getType()

type

------------------

public void setType(String type)

type

------------------

setType

public void setType(String type)

Set the value of the type property.
public Object saveState(FacesContext _context)

saveState

```java
public Object saveState(FacesContext _context)
```

**Description copied from interface:** StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**
saveState in interface StateHolder

**Overrides:**
saveState in class UICommand

public void restoreState(FacesContext _context, Object _state)

restoreState
public void restoreState(FacesContext _context,
Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext,
java.lang.Object) method on all those instances as well.

Specified by: 
restoreState in interface StateHolder

Overrides: 
restoreState in class UICommand

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.component.html  Class HtmlDataTable

java.lang.Object
   ↓ javax.faces.component.UIComponent
      ↓ javax.faces.component.UIComponentBase
         ↓ javax.faces.component.UIData
            ↓ javax.faces.component.html.HtmlDataTable

All Implemented Interfaces:
   NamingContainer, StateHolder

public class HtmlDataTable
   extends UIData

Extends: UIComponent > UIComponentBase > UIData

HTML table UIColumn

   rendererType "javax.faces.Table"  setRendererType()

Represents a set of repeating data (segregated into columns by child UIColumn components) that will be rendered in an HTML table element.

By default, the rendererType property must be set to "javax.faces.Table". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIData

COMPONENT_FAMILY
Fields inherited from class javax.faces.component.UIComponent
bindings

Fields inherited from interface javax.faces.component.NamingContainer
SEPARATOR_CHAR

Constructor Summary

HtmlDataTable()

Method Summary

String getBgcolor()
	Return the value of the bgcolor property.

int getBorder()
	Return the value of the border property.

String getCaptionClass()
	Return the value of the captionClass property.

String getCaptionStyle()
	Return the value of the captionStyle property.

String getCellpadding()
	Return the value of the cellpadding property.

String getCellspacing()
	Return the value of the cellspacing property.

String getColumnClasses()
	Return the value of the columnClasses property.

String getDir()
	Return the value of the dir property.

String getFooterClass()
	Return the value of the footerClass property.

String getFrame()
	Return the value of the frame property.
<table>
<thead>
<tr>
<th>String</th>
<th>Method Name</th>
<th>Function Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getHeaderClass()</td>
<td>Return the value of the headerClass property.</td>
</tr>
<tr>
<td>String</td>
<td>getLang()</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnclick()</td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td>String</td>
<td>getOndblclick()</td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeypress()</td>
<td>Return the value of the onkeypress property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeyup()</td>
<td>Return the value of the onkeyup property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousedown()</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousemove()</td>
<td>Return the value of the onmousemove property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseout()</td>
<td>Return the value of the onmouseout property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseover()</td>
<td>Return the value of the onmouseover property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseup()</td>
<td>Return the value of the onmouseup property.</td>
</tr>
<tr>
<td>String</td>
<td>getRowClasses()</td>
<td>Return the value of the rowClasses property.</td>
</tr>
<tr>
<td>String</td>
<td>getRules()</td>
<td>Return the value of the rules property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyle()</td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyleClass()</td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>String</td>
<td>getSummary()</td>
<td>Return the value of the summary property.</td>
</tr>
<tr>
<td>String getTitle()</td>
<td>Return the value of the title property.</td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------</td>
<td></td>
</tr>
<tr>
<td>String getWidth()</td>
<td>Return the value of the width property.</td>
<td></td>
</tr>
<tr>
<td>void restoreState(FacesContext _context, Object _state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
<td></td>
</tr>
<tr>
<td>Object saveState(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
<td></td>
</tr>
<tr>
<td>void setBgcolor(String bgcolor)</td>
<td>Set the value of the bgcolor property.</td>
<td></td>
</tr>
<tr>
<td>void setBorder(int border)</td>
<td>Set the value of the border property.</td>
<td></td>
</tr>
<tr>
<td>void setCaptionClass(String captionClass)</td>
<td>Set the value of the captionClass property.</td>
<td></td>
</tr>
<tr>
<td>void setCaptionStyle(String captionStyle)</td>
<td>Set the value of the captionStyle property.</td>
<td></td>
</tr>
<tr>
<td>void setCellpadding(String cellpadding)</td>
<td>Set the value of the cellpadding property.</td>
<td></td>
</tr>
<tr>
<td>void setCellspacing(String cellspacing)</td>
<td>Set the value of the cellspacing property.</td>
<td></td>
</tr>
<tr>
<td>void setColumnClasses(String columnClasses)</td>
<td>Set the value of the columnClasses property.</td>
<td></td>
</tr>
<tr>
<td>void setDir(String dir)</td>
<td>Set the value of the dir property.</td>
<td></td>
</tr>
<tr>
<td>void setFooterClass(String footerClass)</td>
<td>Set the value of the footerClass property.</td>
<td></td>
</tr>
<tr>
<td>void setFrame(String frame)</td>
<td>Set the value of the frame property.</td>
<td></td>
</tr>
<tr>
<td>void setHeaderClass(String headerClass)</td>
<td>Set the value of the headerClass property.</td>
<td></td>
</tr>
<tr>
<td>void setLang(String lang)</td>
<td>Set the value of the lang property.</td>
<td></td>
</tr>
<tr>
<td>void setOnclick(String onclick)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>setOndblclick(String ondblclick)</td>
<td>Set the value of the ondblclick property.</td>
<td></td>
</tr>
<tr>
<td>setOnkeydown(String onkeydown)</td>
<td>Set the value of the onkeydown property.</td>
<td></td>
</tr>
<tr>
<td>setOnkeypress(String onkeypress)</td>
<td>Set the value of the onkeypress property.</td>
<td></td>
</tr>
<tr>
<td>setOnkeyup(String onkeyup)</td>
<td>Set the value of the onkeyup property.</td>
<td></td>
</tr>
<tr>
<td>setOnmousedown(String onmousedown)</td>
<td>Set the value of the onmousedown property.</td>
<td></td>
</tr>
<tr>
<td>setOnmousemove(String onmousemove)</td>
<td>Set the value of the onmousemove property.</td>
<td></td>
</tr>
<tr>
<td>setOnmouseout(String onmouseout)</td>
<td>Set the value of the onmouseout property.</td>
<td></td>
</tr>
<tr>
<td>setOnmouseover(String onmouseover)</td>
<td>Set the value of the onmouseover property.</td>
<td></td>
</tr>
<tr>
<td>setOnmouseup(String onmouseup)</td>
<td>Set the value of the onmouseup property.</td>
<td></td>
</tr>
<tr>
<td>setRowClasses(String rowClasses)</td>
<td>Set the value of the rowClasses property.</td>
<td></td>
</tr>
<tr>
<td>setRules(String rules)</td>
<td>Set the value of the rules property.</td>
<td></td>
</tr>
<tr>
<td>setStyle(String style)</td>
<td>Set the value of the style property.</td>
<td></td>
</tr>
<tr>
<td>setStyleClass(String styleClass)</td>
<td>Set the value of the styleClass property.</td>
<td></td>
</tr>
<tr>
<td>setSummary(String summary)</td>
<td>Set the value of the summary property.</td>
<td></td>
</tr>
<tr>
<td>setTitle(String title)</td>
<td>Set the value of the title property.</td>
<td></td>
</tr>
<tr>
<td>setWidth(String width)</td>
<td>Set the value of the width property.</td>
<td></td>
</tr>
</tbody>
</table>
Methods inherited from class javax.faces.component.UIData
broadcast, encodeBegin, getClientId, getDataModel, getFamily, getFirst, getFooter, getHeader, getRowCount, getRowData, getRowIndex, getRows, getValue, getVar, invokeOnComponent, isRowAvailable, processDecodes, processUpdates, processValidators, queueEvent, setDataModel, setFirst, setFooter, setHeader, setRowIndex, setRows, setValue, setValueBinding, setValueExpression, setVar

Methods inherited from class javax.faces.component.UIComponentBase
addFacesListener, decode, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, isRendered, isTransient, processRestoreState, processSaveState, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
### Constructor Detail

```java
public HtmlDataTable()
```

### Method Detail

```java
public String getBgcolor()
```

**getBgcolor**

```java
public String getBgcolor()
```

Return the value of the `bgcolor` property.

Contents: Name or code of the background color for this table.

```java
public void setBgcolor(String bgcolor)
```

**bgcolor**
**setBgcolor**

```java
public void setBgcolor(String bgcolor)
```

Set the value of the bgcolor property.

---

**public int getBorder()**

```java
border
```

---

**getBorder**

```java
public int getBorder()
```

Return the value of the border property.

Contents: Width (in pixels) of the border to be drawn around this table.

---

**public void setBorder(int border)**

```java
border
```

---

**setBorder**

```java
public void setBorder(int border)
```

Set the value of the border property.
public String getCaptionClass()

    captionClass

    CSS

getCaptionClass

public String getCaptionClass()

    Return the value of the captionClass property.

    Contents: Space-separated list of CSS style class(es) that will be applied to any caption generated for this table.

-----------------------------------------------

public void setCaptionClass(String captionClass)

    captionClass

setCaptionClass

public void setCaptionClass(String captionClass)

    Set the value of the captionClass property.

-----------------------------------------------

public String getCaptionStyle()

    captionStyle

    CSS
getCaptionStyle

public String getCaptionStyle()

Return the value of the captionStyle property.

Contents: CSS style(s) to be applied when this caption is rendered.

------------------------------

public void setCaptionStyle(String captionStyle)

captionStyle

setCaptionStyle

public void setCaptionStyle(String captionStyle)

Set the value of the captionStyle property.

------------------------------

public String getCellpadding()

cellpadding

getCellpadding

public String getCellpadding()

Return the value of the cellpadding property.
Contents: Definition of how much space the user agent should leave between the border of each cell and its contents.

public void setCellpadding(String cellpadding)

cellpadding

setCellpadding

public void setCellpadding(String cellpadding)

Set the value of the cellpadding property.

public String getCellspacing()

cellspacing

gCellspacing

public String getCellspacing()

Return the value of the cellspacing property.

Contents: Definition of how much space the user agent should leave between the left side of the table and the leftmost column, the top of the table and the top of the top side of the topmost row, and so on for the right and bottom of the table. It also specifies the amount of space to leave between cells.
public void setCellspacing(String cellspacing)

    cellspacing

setCellspacing

public void setCellspacing(String cellspacing)
    Set the value of the cellspacing property.

public String getColumnClasses()

columnClasses

CSS "columns" "class"
"columns" "columns"

getColumnClasses

public String getColumnClasses()
    Return the value of the columnClasses property.

Contents: Comma-delimited list of CSS style classes that will be applied to the columns of this table. A space separated list of classes may also be specified for any individual column. If the number of elements in this list is less than the number of columns specified in the "columns" attribute, no "class" attribute is output for each column greater than the number of elements in the list. If the number of elements in the list is greater than the number of columns specified in the "columns" attribute, the elements at the position in the list after the value of the "columns" attribute are ignored.
public void setColumnClasses(String columnClasses)

columnClasses

setColumnClasses

public void setColumnClasses(String columnClasses)

Set the value of the columnClasses property.

-----------------------------

gpublic String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

gpublic String getDir()

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

-----------------------------

gpublic void setDir(String dir)

dir
**setDir**

public void setDir(String dir)

Set the value of the dir property.

**public String getFooterClass()**

footerClass

CSS

**getFooterClass**

public String getFooterClass()

Return the value of the footerClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any footer generated for this table.

**public void setFooterClass(String footerClass)**

footerClass

**setFooterClass**

public void setFooterClass(String footerClass)

Set the value of the footerClass property.
public String getFrame()

    frame

    none
    above
    below
    hsides
    vsides
    lhs
    rhs
    box
    border

getFrame

public String getFrame()

    Return the value of the frame property.

    Contents: Code specifying which sides of the frame surrounding this table will be visible. Valid values are: none (no sides, default value); above (top side only); below (bottom side only); hsides (top and bottom sides only); vsides (right and left sides only); lhs (left hand side only); rhs (right hand side only); box (all four sides); and border (all four sides).

public void setFrame(String frame)

    frame

setFrame

public void setFrame(String frame)

    Set the value of the frame property.

public String getHeaderClass()
Getter and Setter:

getHeaderClass

public String getHeaderClass()

Return the value of the headerClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any header generated for this table.

setHeaderClass

public void setHeaderClass(String headerClass)

Set the value of the headerClass property.

getLang

public String getLang()
**getLang**

public String getLang()

Return the value of the `lang` property.

Contents: Code describing the language used in the generated markup for this component.

---

**public void setLang(String lang)**

`lang`

---

**setLang**

public void setLang(String lang)

Set the value of the `lang` property.

---

**public String getOnclick()**

`onclick`

**Javascript**

---

**getOnclick**

public String getOnclick()

Return the value of the `onclick` property.

Contents: Javascript code executed when a pointer button is clicked.
setOnclick

public void setOnclick(String onclick)

onclick

getOndblclick

public String getOndblclick()

getOndblclick

public  String getOndblclick()

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double clicked over this element.
**setOndblclick**

```java
public void setOndblclick(String ondblclick)
```

Set the value of the `ondblclick` property.

---

**public String getOnkeydown()**

```java
public String getOnkeydown()
```

`onkeydown`

**Javascript**

---

**getOnkeydown**

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

---

**public void setOnkeydown(String onkeydown)**

`onkeydown`

---

**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.
public String getOnkeypress()

    onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

    Return the value of the onkeypress property.
    Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

    onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

    Set the value of the onkeypress property.

public String getOnkeyup()

    onkeyup

Javascript
getOnkeyup

```
public String getOnkeyup()

    Return the value of the onkeyup property.

    Contents: Javascript code executed when a key is released over this element.
```

---

public void setOnkeyup(String onkeyup)

    onkeyup

---

setOnkeyup

```
public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.
```

---

public String getOnmousedown()

    onmousedown

Javascript

getOnmousedown

```
public String getOnmousedown()

    Return the value of the onmousedown property.
```
Contents: Javascript code executed when a pointer button is pressed down over this element.

```java
public void setOnmousedown(String onmousedown)

    onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

```
**setOnmousemove**

```java
public void setOnmousemove(String onmousemove)
```

Set the value of the `onmousemove` property.

---

**public String getOnmouseout()**

```java
onmouseout
```

**Javascript**

**getOnmouseout**

```java
public String getOnmouseout()
```

Return the value of the `onmouseout` property.

Contents: Javascript code executed when a pointer button is moved away from this element.

---

**public void setOnmouseout(String onmouseout)**

```java
onmouseout
```

**setOnmouseout**

```java
public void setOnmouseout(String onmouseout)
```

Set the value of the `onmouseout` property.
public String getOnmouseover()

    onmouseover

Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

public String getOnmouseup()

    onmouseup

Javascript
getOnmouseup

public String getOnmouseup()

    Return the value of the onmouseup property.

    Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getRowClasses()

    rowClasses

CSS

getRowClasses

public String getRowClasses()

    Return the value of the rowclasses property.
Contents: Comma-delimited list of CSS style classes that will be applied to the rows of this table. A space separated list of classes may also be specified for any individual row. The styles are applied, in turn, to each row in the table. For example, if the list has two elements, the first style class in the list is applied to the first row, the second to the second row, the first to the third row, the second to the fourth row, etc. In other words, we keep iterating through the list until we reach the end, and then we start at the beginning again.

public void setRowClasses(String rowClasses)

    rowClasses

setRowClasses

public void setRowClasses(String rowClasses)

    Set the value of the rowClasses property.

public String getRules()

    rules

getRules

public String getRules()

    Return the value of the rules property.
Contents: Code specifying which rules will appear between cells within this table. Valid values are: none (no rules, default value); groups (between row groups); rows (between rows only); cols (between columns only); and all (between all rows and columns).

```java
public void setRules(String rules)
```

```
setRules
```

```java
public void setRules(String rules)
```

```
Set the value of the rules property.
```

```java
public String getStyle()
```

```
ggetStyle
```

```java
public String getStyle()
```

```
Return the value of the style property.
```

```
Contents: CSS style(s) to be applied when this component is rendered.
```

```java
public void setStyle(String style)
```
public void setStyle(String style)

Set the value of the style property.

public String getStyleClass()

styleClass

CSS "class"

public String getStyleClass()

Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

styleClass

setStyleClass
public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

public String getSummary()

    summary

    Braille

getSummary

public String getSummary()

    Return the value of the summary property.

    Contents: Summary of this table's purpose and structure, for user agents rendering to non-visual media such as speech and Braille.

public void setSummary(String summary)

    summary

setSummary

public void setSummary(String summary)

    Set the value of the summary property.

public String getTitle()
getTitle

public String getTitle()

    Return the value of the title property.

    Contents: Advisory title information about markup elements generated for this component.

_________________________________________________________________________

public void setTitle(String title)

    title

setTitle

public void setTitle(String title)

    Set the value of the title property.

_________________________________________________________________________

public String getWidth()

    width
**getWidth**

```java
public String getWidth()
```

Return the value of the width property.

Contents: Width of the entire table, for visual user agents.

---

**public void setWidth(String width)**

```java
setWidth
```

Set the value of the width property.

---

**public Object saveState(FacesContext _context)**

**saveState**

```java
saveState
```

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(FacesContext context) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager.
This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be Serializable.

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIData`

---

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface:** `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
- `restoreState` in interface `StateHolder`

**Overrides:**
- `restoreState` in class `UIData`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
javax.faces.component.html Class HtmlForm

java.lang.Object  
   ↓ javax.faces.component.UIComponent  
      ↓ javax.faces.component.UIComponentBase  
         ↓ javax.faces.component.UIForm  
            ↓ javax.faces.component.html.HtmlForm

All Implemented Interfaces:
   NamingContainer, StateHolder

public class HtmlForm
   extends UIForm

Extends: UIComponent > UIComponentBase > UIForm

HTML form input

   rendererType "javax.faces.Form"  setRendererType()

Represents an HTML form element. Child input components will be submitted unless they have been disabled.

By default, the rendererType property must be set to "javax.faces.Form". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIForm

COMPONENT_FAMILY
### Fields inherited from class `javax.faces.component.UIComponent` bindings

### Fields inherited from interface `javax.faces.component.NamingContainer` SEPARATOR_CHAR

### Constructor Summary

**HtmlForm()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAccept()</code></td>
<td>Return the value of the <code>accept</code> property.</td>
</tr>
<tr>
<td><code>getAcceptcharset()</code></td>
<td>Return the value of the <code>acceptcharset</code> property.</td>
</tr>
<tr>
<td><code>getDir()</code></td>
<td>Return the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>getEnctype()</code></td>
<td>Return the value of the <code>enctype</code> property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>Return the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>Return the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td>String (getOnmousedown())</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td>String (getOnmousemove())</td>
<td>Return the value of the onmousemove property.</td>
</tr>
<tr>
<td>String (getOnmouseout())</td>
<td>Return the value of the onmouseout property.</td>
</tr>
<tr>
<td>String (getOnmouseover())</td>
<td>Return the value of the onmouseover property.</td>
</tr>
<tr>
<td>String (getOnmouseup())</td>
<td>Return the value of the onmouseup property.</td>
</tr>
<tr>
<td>String (getOnreset())</td>
<td>Return the value of the onreset property.</td>
</tr>
<tr>
<td>String (getOnsubmit())</td>
<td>Return the value of the onsubmit property.</td>
</tr>
<tr>
<td>String (getStyle())</td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>String (getStyleClass())</td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>String (getTarget())</td>
<td>Return the value of the target property.</td>
</tr>
<tr>
<td>String (getTitle())</td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td>void (restoreState(FacesContext _context, Object _state))</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object (saveState(FacesContext _context))</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void (setAccept(String accept))</td>
<td>Set the value of the accept property.</td>
</tr>
<tr>
<td>void (setAcceptcharset(String acceptcharset))</td>
<td>Set the value of the acceptcharset property.</td>
</tr>
<tr>
<td>void (setDir(String dir))</td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td>void (setEnctype(String enctype))</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>void setLang(String lang)</code></td>
<td>Set the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>void setOnclick(String onclick)</code></td>
<td>Set the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>void setOndblclick(String ondblclick)</code></td>
<td>Set the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>void setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>void setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>void setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>void setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>void setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>void setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>void setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>void setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>void setOnreset(String onreset)</code></td>
<td>Set the value of the <code>onreset</code> property.</td>
</tr>
<tr>
<td><code>void setOnsubmit(String onsubmit)</code></td>
<td>Set the value of the <code>onsubmit</code> property.</td>
</tr>
<tr>
<td><code>void setTitle(String title)</code></td>
<td>Set the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>void setStyle(String style)</code></td>
<td>Set the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>void setStyleClass(String styleClass)</code></td>
<td>Set the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>void setTarget(String target)</code></td>
<td>Set the value of the <code>target</code> property.</td>
</tr>
<tr>
<td><code>void setTitle(String title)</code></td>
<td>Set the value of the <code>title</code> property.</td>
</tr>
</tbody>
</table>
Set the value of the title property.

Methods inherited from class javax.faces.component.UIForm
getContainerClientId, getFamily, isPrependId, isSubmitted, processDecodes, processUpdates, processValidators, setPrependId, setSubmitted

Methods inherited from class javax.faces.component.UIComponentBase
addFacesListener, broadcast, decode, encodeBegin, encodeChildren, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constructor Detail

public HtmlForm()

Method Detail

public String getAccept()

   accept

getAccept

public String getAccept()

   Return the value of the accept property.

   Contents: List of content types that a server processing this form will handle correctly

public void setAccept(String accept)

   accept
**setAccept**

```java
public void setAccept(String accept)
```

Set the value of the accept property.

**public String getAcceptcharset()**

```java
acceptcharset
```

**getAcceptcharset**

```java
public String getAcceptcharset()
```

Return the value of the acceptcharset property.

Contents: List of character encodings for input data that are accepted by the server processing this form.

**public void setAcceptcharset(String acceptcharset)**

```java
acceptcharset
```

**setAcceptcharset**

```java
public void setAcceptcharset(String acceptcharset)
```

Set the value of the acceptcharset property.
public String getDir()

dir
"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-
left).

    public void setDir(String dir)

dir

setDir

public void setDir(String dir)

    Set the value of the dir property.

    public String getEnctype()

tenctype

"application/x-www-form-urlencoded"
**getEnctype**

```java
public String getEnctype()
```

Return the value of the `enctype` property.

Contents: Content type used to submit the form to the server. If not specified, the default value is "application/x-www-form-urlencoded".

---

**public void setEnctype(String enctype)**

`enctype`

---

**setEnctype**

```java
public void setEnctype(String enctype)
```

Set the value of the `enctype` property.

---

**public String getLang()**

`lang`

---

**getLang**

```java
public String getLang()
```

Return the value of the `lang` property.
Contents: Code describing the language used in the generated markup for this component.

```
public void setLang(String lang)

lang

setLang

public void setLang(String lang)

Set the value of the lang property.
```

```
public String getOnclick()

onclick

Javascript

getOnclick

public String getOnclick()

Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.
```

```
public void setOnclick(String onclick)

onclick
```
**setOnclick**

public void setOnclick(String onclick)

Set the value of the onclick property.

---

**public String getOndblclick()**

ondblclick

**Javascript**

**getOndblclick**

public String getOndblclick()

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

---

**public void setOndblclick(String ondblclick)**

ondblclick

**setOndblclick**

public void setOndblclick(String ondblclick)

Set the value of the ondblclick property.
public String getOnkeydown()

    onkeydown

Javascript

getOnkeydown

class javscript

public String getOnkeydown()

    Return the value of the onkeydown property.

    Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

    onkeydown

setOnkeydown

class javscript

public void setOnkeydown(String onkeydown)

    Set the value of the onkeydown property.

public String getOnkeypress()

    onkeypress

Javascript
**getOnkeypress**

```java
public String getOnkeypress()
```

Return the value of the `onkeypress` property.

Contents: Javascript code executed when a key is pressed and released over this element.

**public void setOnkeypress(String onkeypress)**

```java
onkeypress
```

**setOnkeypress**

```java
public void setOnkeypress(String onkeypress)
```

Set the value of the `onkeypress` property.

**public String getOnkeyup()**

```java
onkeyup
```

**Javascript**

**getOnkeyup**

```java
public String getOnkeyup()
```

Return the value of the `onkeyup` property.
Contents: Javascript code executed when a key is released over this element.

---

public void setOnkeyup(String onkeyup)

    onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.

---

public String getOnmousedown()

    onmousedown

Javascript

getOnmousedown

public  String  getOnmousedown()

    Return the value of the onmousedown property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.

---

public void setOnmousedown(String onmousedown)

    onmousedown
setOnmousedown

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

-----------------------------------------------

getOnmousemove

public String getOnmousemove()

    onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

    Return the value of the onmousemove property.

    Contents: Javascript code executed when a pointer button is moved within this element.

-----------------------------------------------

setOnmousemove

public void setOnmousemove(String onmousemove)

    onmousemove

setOnmousemove

public void setOnmousemove(String onmousemove)

    Set the value of the onmousemove property.
public String getOnmouseout()

    onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

    Return the value of the onmouseout property.
    Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

    onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

    Set the value of the onmouseout property.

public String getOnmouseover()

    onmouseover

Javascript
getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public String getOnmouseup()

    Return the value of the onmouseup property.
Contents: Javascript code executed when a pointer button is released over this element.

**public void setOnmouseup(String onmouseup)**

```
onmouseup
```

**setOnmouseup**

```
public void setOnmouseup(String onmouseup)
```

Set the value of the `onmouseup` property.

**public String getOnreset()**

```
onreset
```

**getOnreset**

```
public String getOnreset()
```

Return the value of the `onreset` property.

Contents: Javascript code executed when this form is reset.

**public void setOnreset(String onreset)**

```
onreset
```
**setOnreset**

public void setOnreset(String onreset)

Set the value of the onreset property.

---

**public String getOnsubmit()**

onsubmit

Javascript

**getOnsubmit**

public String getOnsubmit()

Return the value of the onsubmit property.

Contents: Javascript code executed when this form is submitted.

---

**public void setOnsubmit(String onsubmit)**

onsubmit

**setOnsubmit**

public void setOnsubmit(String onsubmit)

Set the value of the onsubmit property.
public String getStyle()

   style

CSS

gTextStyle

public String getStyle()

   Return the value of the style property.

   Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

   style

setStyle

public void setStyle(String style)

   Set the value of the style property.

public String getStyleClass()

   styleClass

CSS "class"
**getStyleClass**

public String getStyleClass()

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

**public void setStyleClass(String styleClass)**

`styleClass`

**setStyleClass**

public void `setStyleClass(String styleClass)`

Set the value of the `styleClass` property.

**public String getTarget()**

`target`

**getTarget**

public String `getTarget()`
Return the value of the target property.

Contents: Name of a frame where the response retrieved after this form submit is to be displayed.

public void setTarget(String target)

target

setDefault

public void setTarget(String target)

Set the value of the target property.

public String getTitle()

title

getTitle

public String getTitle()

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

public void setTitle(String title)
setTitle

public void setTitle(String title)

Set the value of the title property.

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
    restoreState in interface StateHolder

Overrides:
    restoreState in class UIComponentBase
PS:
javax.faces.component.html  Class HtmlGraphicImage

java.lang.Object  
  ↓  javax.faces.component.UIComponent  
    ↓  javax.faces.component.UIComponentBase  
      ↓  javax.faces.component.UIGraphic  
        ↓  javax.faces.component.html.HtmlGraphicImage

All Implemented Interfaces: 
  StateHolder

public class HtmlGraphicImage
  extends UIGraphic

Extends: UICOMPONENT > UICOMPONENTBASE > UIGRAPHIC

HTML  img

    rendererType "javax.faces.Image"  setRendererType()

Represents an HTML img element, used to retrieve and render a graphical image.

By default, the rendererType property must be set to "javax.faces.Image". This value can be changed by calling the setRendererType() method.

---

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIGraphic

COMPONENT_FAMILY
### Fields inherited from class `javax.faces.component.UIComponent bindings`

### Constructor Summary

```java
HtmlGraphicImage()
```

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAlt()</code></td>
<td>Return the value of the <code>alt</code> property.</td>
</tr>
<tr>
<td><code>getDir()</code></td>
<td>Return the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>getHeight()</code></td>
<td>Return the value of the <code>height</code> property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>Return the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>getLongdesc()</code></td>
<td>Return the value of the <code>longdesc</code> property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>Return the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
</tbody>
</table>
getOnmouseout()  
Return the value of the onmouseout property.

getOnmouseover()  
Return the value of the onmouseover property.

getOnmouseup()  
Return the value of the onmouseup property.

getStyle()  
Return the value of the style property.

getStyleClass()  
Return the value of the styleClass property.

title()  
Return the value of the title property.

getUsemap()  
Return the value of the usemap property.

getWidth()  
Return the value of the width property.

isIsmap()  
Return the value of the ismap property.

restoreState(FacesContext _context, Object _state)  
Perform any processing required to restore the state from the entries in the state Object.

saveState(FacesContext _context)  
Gets the state of the instance as a Serializable Object.

setAlt(String alt)  
Set the value of the alt property.

setDir(String dir)  
Set the value of the dir property.

setHeight(String height)  
Set the value of the height property.

setIsmap(boolean ismap)  
Set the value of the ismap property.

setLang(String lang)  
Set the value of the lang property.

setLongdesc(String longdesc)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <strong>setOnclick</strong>(<em>String</em> onclick)</td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td>void <strong>setOndblclick</strong>(<em>String</em> ondblclick)</td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td>void <strong>setOnkeydown</strong>(<em>String</em> onkeydown)</td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td>void <strong>setOnkeypress</strong>(<em>String</em> onkeypress)</td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td>void <strong>setOnkeyup</strong>(<em>String</em> onkeyup)</td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td>void <strong>setOnmousedown</strong>(<em>String</em> onmousedown)</td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td>void <strong>setOnmousemove</strong>(<em>String</em> onmousemove)</td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td>void <strong>setOnmouseout</strong>(<em>String</em> onmouseout)</td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td>void <strong>setOnmouseover</strong>(<em>String</em> onmouseover)</td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td>void <strong>setOnmouseup</strong>(<em>String</em> onmouseup)</td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td>void <strong>setStyle</strong>(<em>String</em> style)</td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>void <strong>setStyleClass</strong>(<em>String</em> styleClass)</td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td>void <strong>setTitle</strong>(<em>String</em> title)</td>
<td>Set the value of the title property.</td>
</tr>
<tr>
<td>void <strong>setUsemap</strong>(<em>String</em> usemap)</td>
<td>Set the value of the usemap property.</td>
</tr>
<tr>
<td>void <strong>setWidth</strong>(<em>String</em> width)</td>
<td>Set the value of the width property.</td>
</tr>
</tbody>
</table>

Methods inherited from class *javax.faces.component.UIGraphic*:
- **getFamily**, **getUrl**, **getValue**, **getValueBinding**, **getValueExpression**, ...
Methods inherited from class javax.faces.component.UIComponentBase
addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
   Constant Field Values

Constructor Detail
public HtmlGraphicImage()

HtmlGraphicImage

public HtmlGraphicImage()

### Method Detail

**public String getAlt()**

`alt`

getAlt

**public String getAlt()**

Return the value of the alt property.

Contents: Alternate textual description of the element rendered by this component.

---

**public void setAlt(String alt)**

`alt`

setAlt

**public void setAlt(String alt)**
Set the value of the alt property.

```java
public String getDir()

  dir

 "LTR" left-to-right "RTL" right-to-left
```

getDir

```java
public String getDir()

  Return the value of the dir property.

  Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-
left).
```

```java
public void setDir(String dir)

  dir
```

setDir

```java
public void setDir(String dir)

  Set the value of the dir property.
```

```java
public String getHeight()

  height
```
**getHeigh**

```java
public String getHeight()

    Return the value of the height property.
    
    Contents: Override for the height of this image.
```

**public void setHeigh(String height)**

```java
height
```

**setHeigh**

```java
public void setHeight(String height)

    Set the value of the height property.
```

**public boolean isIsmap()**

```java
ismap

("a") false true ismap="ismap"
```

**isIsmap**

```java
public boolean isIsmap()
```
Return the value of the `ismap` property.

Contents: Flag indicating that this image is to be used as a server side image map. Such an image must be enclosed within a hyperlink ("a"). A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `ismap="ismap"`.

```java
public void setIsmap(boolean ismap)

   ismap
```

`setIsmap`

```java
public void setIsmap(boolean ismap)

   Set the value of the `ismap` property.
```

```java
public String getLang()

   lang
```

`getLang`

```java
public String getLang()

   Return the value of the `lang` property.

   Contents: Code describing the language used in the generated markup for this component.
```
public void setLang(String lang)

lang

setLang

public void setLang(String lang)

Set the value of the lang property.

public String getLongdesc()

longdesc

URI

getLongdesc

public String getLongdesc()

Return the value of the longdesc property.

Contents: URI to a long description of the image represented by this element.

public void setLongdesc(String longdesc)

longdesc
**setLongdesc**

```java
public void setLongdesc(String longdesc)
```

Set the value of the longdesc property.

---

**public String getOnclick()**

```java
public String getOnclick()
```

Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.

---

**public void setOnclick(String onclick)**

```java
public void setOnclick(String onclick)
```

---

**setOnclick**

```java
public void setOnclick(String onclick)
```

Set the value of the onclick property.
public String getOndblclick()

    ondblclick

Javascript

---

getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.

    Contents: Javascript code executed when a pointer button is double clicked over this element.

---

public void setOndblclick(String ondblclick)

    ondblclick

---

setOndblclick

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

---

public String getOnkeydown()

    onkeydown

Javascript
**getOnkeydown**

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

---

**public void setOnkeydown(String onkeydown)**

```java
onkeydown
```

---

**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.

---

**public String getOnkeypress()**

```java
onkeypress
```

---

**getOnkeypress**

```java
public String getOnkeypress()
```

Return the value of the `onkeypress` property.
Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

  onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

  Set the value of the onkeypress property.

public String getOnkeyup()

  onkeyup

Javascript

getOnkeyup

public String getOnkeyup()

  Return the value of the onkeyup property.

  Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

  onkeyup
**setOnkeyup**

public void **setOnkeyup**(*String* onkeyup)

Set the value of the onkeyup property.

---

**public String getOnmousedown()**

*onmousedown*

**Javascript**

**getOnmousedown**

public **String** **getOnmousedown**()

Return the value of the onmousedown property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

---

**public void setOnmousedown(String onmousedown)**

*onmousedown*

**setOnmousedown**

public void **setOnmousedown**(*String* onmousedown)

Set the value of the onmousedown property.
public String getOnmousemove()

  onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

  Return the value of the onmousemove property.

  Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

  onmousemove

setOnmousemove

public void setOnmousemove(String onmousemove)

  Set the value of the onmousemove property.

public String getOnmouseout()

  onmouseout

Javascript
**getOnmouseout**

```java
public String getOnmouseout()
```

Return the value of the `onmouseout` property.

Contents: Javascript code executed when a pointer button is moved away from this element.

---

**public void setOnmouseout(String onmouseout)**

```
setOnmouseout
```

---

**setOnmouseover**

```java
public void setOnmouseover(String onmouseover)
```

Set the value of the `onmouseover` property.

---

**public String getOnmouseover()**

```
getOnmouseover
```

Javascript

---

**getOnmouseover**

```java
public String getOnmouseover()
```

Return the value of the `onmouseover` property.
Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

  onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

  Set the value of the onmouseover property.

public String getOnmouseup()

  onmouseup

Javascript

getOnmouseup

public String getOnmouseup()

  Return the value of the onmouseup property.

  Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

  onmouseup
**setOnmouseup**

public void setOnmouseup(String onmouseup)

Set the value of the onmouseup property.

---

**public String getStyle()**

style

CSS

**getStyle**

public String getStyle()

Return the value of the style property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

style

**setStyle**

public void setStyle(String style)

Set the value of the style property.
public String getStyleClass()

    styleClass

    CSS "class"

getStyleClass

public String getStyleClass()  

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

    styleClass

setStyleClass

public void setStyleClass(String styleClass)  

    Set the value of the styleClass property.

public String getTitle()

    title
**getTitle**

public String getTitle()

    Return the value of the title property.

    Contents: Advisory title information about markup elements generated for this component.

**public void setTitle(String title)**

    title

**setTitle**

public void setTitle(String title)

    Set the value of the title property.

**public String getUsemap()**

    usemap

HTML "map"

**getUsemap**

public String getUsemap()
Return the value of the usemap property.

Contents: The name of a client side image map (an HTML "map" element) for which this element provides the image.

```java
public void setUsemap(String usemap)
```

**setUsemap**

```java
public void setUsemap(String usemap)
```

Set the value of the usemap property.

```java
public String getWidth()
```

**getWidth**

```java
public String getWidth()
```

Return the value of the width property.

Contents: Override for the width of this image.

```java
public void setWidth(String width)
```
width

setWidth

```java
public void setWidth(String width)
```

Set the value of the width property.

public Object saveState(FacesContext _context)

saveState

```java
public Object saveState(FacesContext _context)
```

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
    restoreState in interface StateHolder

Overrides:
    restoreState in class UIGraphic

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.faces.component.html  Class HtmlInputHidden

java.lang.Object
  ▼ javax.faces.component.UIComponent
      ▼ javax.faces.component.UIComponentBase
          ▼ javax.faces.component.UIOutput
              ▼ javax.faces.component.UIInput
                  ▼ javax.faces.component.html.HtmlInputHidden

All Implemented Interfaces:
    EditableValueHolder, StateHolder, ValueHolder

public class HtmlInputHidden
extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput

hidden HTML input

rendererType "javax.faces.Hidden" setRendererType()

Represents an HTML input element of type hidden.

By default, the rendererType property must be set to "javax.faces.Hidden". This value can be changed by calling the setRendererType() method.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>

**Fields inherited from class javax.faces.component.UIInput**
### Fields inherited from class `javax.faces.component.UIComponent` bindings

### Constructor Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HtmlInputHidden</code></td>
<td>()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.component.UIInput`

- `addValidator`, `addValueChangeListener`, `compareValues`, `decode`, `getConvertedValue`, `getConverterMessage`, `getFamily`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValidator`, `setValidatorMessage`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`, `validateValue`

### Methods inherited from class `javax.faces.component.UIOutput`

- `getConverter`, `getLocalValue`, `getValue`, `setConverter`

### Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `encodeBegin`, `encodeChildren`, `encodeChildrenBegin`, `encodeEnd`, `encodeBeginChildren`, `encodeChildrenEnd`, `encodeLocalChildren`, `encodePartial`, `encodePartialChildren`, `encodePartialLocalChildren`, `encodePartialUpdate`, `encodeSubtree`, `getComponentFactory`, `getComponentFactoryId`, `getId`, `getPartialId`, `isContainer`, `isEditable`, `isEligibleForPartialProcessing`, `isInput`, `isLocalComponent`, `isLocalId`, `isMessage`, `isPartialComponent`, `isPartialId`, `isRendered`, `isRenderedLocal`, `isRenderedLocalComponent`, `isRenderedLocalId`, `isRenderedMessage`, `isRenderedPartialComponent`, `isRenderedPartialId`, `isRenderedPartialMessage`, `isRenderedPartialView`, `isRenderedRemote`, `isRemoteComponent`, `isRemoteId`, `isRemoteMessage`, `isRemotePartialComponent`, `isRemotePartialId`, `isRemotePartialMessage`, `isRemotePartialView`, `isRemoteView`, `isRenderable`, `isSubmitDefault`, `isSubmitManual`, `isSubmitRequired`, `isSubmitValue`, `isSubmitView`, `isValidated`, `isValidationMessage`, `isValidationMessageIgnored`, `isValidationMessagesIgnored`, `isValidationType`, `isView`, `process`, `processEvent`, `processEvents`, `processInstruction`, `processRendered`, `processRenderedLocal`, `processRenderedMessage`, `processRenderedPartial`, `processRenderedPartialLocal`, `processRenderedPartialMessage`, `processRenderedView`, `processUpdate`, `processUpdates`, `processValidators`, `removeFacesListener`, `renderChildren`, `renderLocalChildren`, `renderPartial`, `renderPartialChildren`, `renderPartialLocalChildren`, `renderPartialUpdate`, `renderSubtree`, `renderUpdate`, `setComponentFactory`, `setComponentFactoryId`, `setId`, `setPartialId`, `setRendered`, `setRenderedLocal`, `setRenderedLocalComponent`, `setRenderedLocalId`, `setRenderedMessage`, `setRenderedPartialComponent`, `setRenderedPartialId`, `setRenderedPartialMessage`, `setRenderedPartialView`, `setRenderedRemote`, `setRenderedRemoteComponent`, `setRenderedRemoteId`, `setRenderedRemoteMessage`, `setRenderedRemotePartialComponent`, `setRenderedRemotePartialId`, `setRenderedRemotePartialMessage`, `setRenderedRemotePartialView`, `setRenderedRemoteView`, `setRenderable`, `setSubmitDefault`, `setSubmitManual`, `setSubmitRequired`, `setSubmitValue`, `setSubmitView`, `setValidated`, `setValidationMessage`, `setValidationMessageIgnored`, `setValidationMessagesIgnored`, `setValidationType`, `setView`, `validate`, `validateValue`
Methods inherited from class javax.faces.component.UIComponent

encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Methods inherited from interface javax.faces.component.ValueHolder

getConverter, getLocalValue, getValue, setConverter

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail
public HtmlInputHidden()

HtmlInputHidden

public HtmlInputHidden()

Method Detail

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by: saveState in interface StateHolder
Overrides:

```java
saveState in class UIInput
```

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface:** StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**

```java
restoreState in interface StateHolder
```

**Overrides:**

```java
restoreState in class UIInput
```

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.faces.component.html Class HtmlInputSecret

java.lang.Object
  ↓ javax.faces.component.UIComponent
  ↓ javax.faces.component.UIComponentBase
  ↓ javax.faces.component.UIOutput
  ↓ javax.faces.component.UIInput
  ↓ javax.faces.component.html.HtmlInputSecret

All Implemented Interfaces:
  EditableValueHolder, StateHolder, ValueHolder

public class HtmlInputSecret
extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput

password HTML input redisplay true
rendererType "javax.faces.Secret" setRendererType()

Represents an HTML input element of type password. On a redisplay, any previously entered value will not be rendered (for security reasons) unless the redisplay property is set to true.

By default, the rendererType property must be set to "javax.faces.Secret". This value can be changed by calling the setRendererType() method.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String COMPONENT_TYPE</td>
</tr>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>
### Fields inherited from class `javax.faces.component.UIInput`
- `COMPONENT_FAMILY`, `CONVERSION MESSAGE ID`, `REQUIRED MESSAGE ID`, `UPDATE MESSAGE ID`

### Fields inherited from class `javax.faces.component.UIComponent`
- `bindings`

### Constructor Summary

- **HtmlInputSecret()**

### Method Summary

- **String [getAccesskey()](#)**
  - Return the value of the accesskey property.

- **String [getAlt()](#)**
  - Return the value of the alt property.

- **String [getAutocomplete()](#)**
  - Return the value of the autocomplete property.

- **String [getDir()](#)**
  - Return the value of the dir property.

- **String [getLabel()](#)**
  - Return the value of the label property.

- **String [getLang()](#)**
  - Return the value of the lang property.

- **int [getMaxlength()](#)**
  - Return the value of the maxlength property.

- **String [getOnblur()](#)**
  - Return the value of the onblur property.

- **String [getOnchange()](#)**
  - Return the value of the onchange property.

- **String [getOnclick()](#)**
  - Return the value of the onclick property.
<table>
<thead>
<tr>
<th>String</th>
<th>getOndblclick()</th>
<th>Return the value of the ondblclick property.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getOnfocus()</td>
<td>Return the value of the onfocus property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeypress()</td>
<td>Return the value of the onkeypress property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeyup()</td>
<td>Return the value of the onkeyup property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousedown()</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousemove()</td>
<td>Return the value of the onmousemove property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseout()</td>
<td>Return the value of the onmouseout property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseover()</td>
<td>Return the value of the onmouseover property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseup()</td>
<td>Return the value of the onmouseup property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnselect()</td>
<td>Return the value of the onselect property.</td>
</tr>
<tr>
<td>int</td>
<td>getSize()</td>
<td>Return the value of the size property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyle()</td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyleClass()</td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>String</td>
<td>getTabindex()</td>
<td>Return the value of the tabindex property.</td>
</tr>
<tr>
<td>String</td>
<td>getTitle()</td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td>boolean</td>
<td>isEnabled()</td>
<td>Return the value of the disabled property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>boolean isreadonly()</td>
<td>Return the value of the readonly property.</td>
<td></td>
</tr>
<tr>
<td>boolean isredisplay()</td>
<td>Return the value of the redisplay property.</td>
<td></td>
</tr>
<tr>
<td>void restoreState(FacesContext _context, Object _state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
<td></td>
</tr>
<tr>
<td>Object saveState(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
<td></td>
</tr>
<tr>
<td>void setAccesskey(String accesskey)</td>
<td>Set the value of the accesskey property.</td>
<td></td>
</tr>
<tr>
<td>void setAlt(String alt)</td>
<td>Set the value of the alt property.</td>
<td></td>
</tr>
<tr>
<td>void setAutocomplete(String autocomplete)</td>
<td>Set the value of the autocomplete property.</td>
<td></td>
</tr>
<tr>
<td>void setDir(String dir)</td>
<td>Set the value of the dir property.</td>
<td></td>
</tr>
<tr>
<td>void setDisabled(boolean disabled)</td>
<td>Set the value of the disabled property.</td>
<td></td>
</tr>
<tr>
<td>void setLabel(String label)</td>
<td>Set the value of the label property.</td>
<td></td>
</tr>
<tr>
<td>void setLang(String lang)</td>
<td>Set the value of the lang property.</td>
<td></td>
</tr>
<tr>
<td>void setMaxlength(int maxlength)</td>
<td>Set the value of the maxlength property.</td>
<td></td>
</tr>
<tr>
<td>void setOnblur(String onblur)</td>
<td>Set the value of the onblur property.</td>
<td></td>
</tr>
<tr>
<td>void setOnchange(String onchange)</td>
<td>Set the value of the onchange property.</td>
<td></td>
</tr>
<tr>
<td>void setOnfocus(String onfocus)</td>
<td>Set the value of the onfocus property.</td>
<td></td>
</tr>
<tr>
<td>void setOnfocus(String onfocus)</td>
<td>Set the value of the onfocus property.</td>
<td></td>
</tr>
<tr>
<td>void setOndblclick(String ondblclick)</td>
<td>Set the value of the ondblclick property.</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td>Parameters</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>------------</td>
</tr>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
<td></td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setRedisplay(boolean redisplay)</code></td>
<td>Set the value of the redisplay property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setSize(int size)</code></td>
<td>Set the value of the size property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>
Methods inherited from class `javax.faces.component.UIInput`

- `addValidator`
- `addValueChangeListener`
- `compareValues`
- `decode`
- `getConvertedValue`
- `getConverterMessage`
- `getFamily`
- `getRequiredMessage`
- `getSubmittedValue`
- `getValidator`
- `getValidatorMessage`
- `getValidators`
- `getValueChangeListener`
- `getValueChangeListeners`
- `isImmediate`
- `isLocalValueSet`
- `isRequired`
- `isValid`
- `processDecodes`
- `processUpdates`
- `processValidators`
- `removeValidator`
- `removeValueChangeListener`
- `resetValue`
- `setConverterMessage`
- `setImmediate`
- `setLocalValueSet`
- `setRequired`
- `setRequiredMessage`
- `setSubmittedValue`
- `setValid`
- `setValidator`
- `setValidatorMessage`
- `setValue`
- `setValueChangeListener`

Methods inherited from class `javax.faces.component.UIOutput`

- `getConverter`
- `getLocalValue`
- `getValue`
- `setConverter`

Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`
- `broadcast`
- `encodeBegin`
- `encodeChildren`
- `encodeEnd`
- `findComponent`
- `getAttributes`
- `getChildCount`
- `getChildren`
- `getClientId`
- `getFacesContext`
- `getFacesListeners`
- `getFacet`
- `getFacetCount`
- `getFacets`
- `getFacetsAndChildren`
- `getId`
- `getParent`
- `getRenderer`
- `getRendererType`
- `getRendersChildren`
- `getValueBinding`
- `invokeOnComponent`
- `isRendered`
- `isTransient`
- `processRestoreState`
- `processSaveState`
- `queueEvent`
- `removeFacesListener`
- `restoreAttachedState`
- `saveAttachedState`
- `setId`
- `setParent`
- `setRendered`
- `setRendererType`
- `setTransient`
- `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`
- `getContainerClientId`
- `getValueExpression`
- `setValueExpression`

Methods inherited from class `java.lang.Object`

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `toString`
- `wait`

Methods inherited from interface `javax.faces.component.ValueHolder`

- `getConverter`
- `getLocalValue`
- `getValue`
- `setConverter`
### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

[Constant Field Values](#)

### Constructor Detail

```java
public HtmlInputSecret()
```

**HtmlInputSecret**

```java
public HtmlInputSecret()
```

### Method Detail

```java
public String getAccesskey()
```

accesskey

getAccesskey
public String getAccesskey()

    Return the value of the accesskey property.
    Contents: Access key that, when pressed, transfers focus to this element.

public void setAccesskey(String accesskey)

    accesskey

setAccesskey

public void setAccesskey(String accesskey)

    Set the value of the accesskey property.

public String getAlt()

    alt

gGetAlt

public String getAlt()

    Return the value of the alt property.
    Contents: Alternate textual description of the element rendered by this component.
public void setAlt(String alt)

    alt

setAlt

public void setAlt(String alt)

    Set the value of the alt property.

public String getAutocomplete()

    autocomplete

"off" "off" "on"

getAutocomplete

public String getAutocomplete()

    Return the value of the autocomplete property.

    Contents: If the value of this attribute is "off", render "off" as the
    value of the attribute. This indicates that the browser should disable
    its autocomplete feature for this component. This is useful for
    components that perform autocompletion and do not want the
    browser interfering. If this attribute is not set or the value is "on",
    render nothing.

public void setAutocomplete(String autocomplete)

    autocomplete
setAutocomplete

public void setAutocomplete(String autocomplete)

    Set the value of the autocomplete property.

public String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-
left).

public void setDir(String dir)

dir

setDir

public void setDir(String dir)
Set the value of the dir property.

```java
public boolean isDisabled()
{
    disabled
    false true disabled="disabled"
}
```

isDisabled

```java
public boolean isDisabled()
{
    Return the value of the disabled property.
    Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".
}
```

```java
public void setDisabled(boolean disabled)
{
    disabled
}
```

setDisabled

```java
public void setDisabled(boolean disabled)
{
    Set the value of the disabled property.
}
```

```java
public String getLabel()
```
public String getLabel()

Return the value of the label property.

Contents: A localized user presentable name for this component.

public void setLabel(String label)

Set the value of the label property.

public String getLang()

getLang
public String getLang()

    Return the value of the lang property.

    Contents: Code describing the language used in the generated markup for this component.

    -----------------------------------------------

public void setLang(String lang)

    lang

setLang

public void setLang(String lang)

    Set the value of the lang property.

    -----------------------------------------------

public int getMaxlength()

    maxlength

getMaxlength

public int getMaxlength()

    Return the value of the maxlength property.

    Contents: The maximum number of characters that may be entered in this field.
public void setMaxlength(int maxlength)

setMaxlength

public void setMaxlength(int maxlength)

Set the value of the maxlength property.

public String getOnblur()

getOnblur

Javascript

public String getOnblur()

Return the value of the onblur property.

Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

setOnblur

onblur
public void setOnblur(String onblur)

Set the value of the onblur property.

public String getOnchange()

onchange

Javascript

getOnchange

public String getOnchange()

Return the value of the onchange property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

public void setOnchange(String onchange)

onchange

setOnchange

public void setOnchange(String onchange)

Set the value of the onchange property.

public String getOnclick()
onclick

Javascript

getOnclick

public String getOnclick()

Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

onclick

setOnclick

public void setOnclick(String onclick)

Set the value of the onclick property.

public String getOndblclick()

 ondblclick

Javascript
**getOndblclick**

public String getOndblclick()

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

---

**public void setOndblclick(String ondblclick)**

ondblclick

**setOndblclick**

public void setOndblclick(String ondblclick)

Set the value of the ondblclick property.

---

**public String getOnfocus()**

onfocus

**Javascript**

**getOnfocus**

public String getOnfocus()

Return the value of the onfocus property.

Contents: Javascript code executed when this element receives
public void setOnfocus(String onfocus)

    onfocus

setOnfocus

public void setOnfocus(String onfocus)

    Set the value of the onfocus property.

public String getOnkeydown()

    onkeydown

Javascript

getOnkeydown

public String getOnkeydown()

    Return the value of the onkeydown property.

    Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

    onkeydown
setOnkeydown

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

---

public String getOnkeypress()

onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

Return the value of the onkeypress property.

Contents: Javascript code executed when a key is pressed and released over this element.

---

public void setOnkeypress(String onkeypress)

onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.
public String getOnkeyup()

  onkeyup

Javascript

getOnkeyup

public String getOnkeyup()

  Return the value of the onkeyup property.

  Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

  onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

  Set the value of the onkeyup property.

public String getOnmousedown()

  onmousedown

Javascript
getOnmousedown

```java
public String getOnmousedown()
```

Return the value of the `onmousedown` property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

---

public void setOnmousedown(String onmousedown)

```java
onmousedown
```

---

setOnmousedown

```java
public void setOnmousedown(String onmousedown)
```

Set the value of the `onmousedown` property.

---

public String getOnmousemove()

```java
onmousemove
```

Javascript

---

getOnmousemove

```java
public String getOnmousemove()
```

Return the value of the `onmousemove` property.
public void setOnmousemove(String onmousemove)

setOnmousemove

public void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.

public String getOnmouseout()

getOnmouseout

public String getOnmouseout()

Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

setOnmouseout

onmouseout

onmouseout
**setOnmouseout**

```java
public void setOnmouseout(String onmouseout)
```

Set the value of the onmouseout property.

**getOnmouseover**

```java
public String getOnmouseover()
```

onmouseover

Javascript

**getOnmouseover**

```java
public String getOnmouseover()
```

Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.

**setOnmouseover**

```java
public void setOnmouseover(String onmouseover)
```

onmouseover

**setOnmouseover**

```java
public void setOnmouseover(String onmouseover)
```

Set the value of the onmouseover property.
public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public String getOnmouseup()

    Return the value of the onmouseup property.

    Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getOnselect()

    onselect

Javascript
**getOnselect**

```java
public String getOnselect()
```

Return the value of the `onselect` property.

Contents: Javascript code executed when text within this element is selected by the user.

---

**public void setOnselect(String onselect)**

`onselect` setOnselect

---

**setOnselect**

```java
public void setOnselect(String onselect)
```

Set the value of the `onselect` property.

---

**public boolean isReadonly()**

`readonly`

false true readonly="readonly"

---

**isReadonly**

```java
public boolean isReadonly()
```
Return the value of the `readonly` property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `readonly="readonly"`.

```java
public void setReadonly(boolean readonly)

readonly
```

`setReadonly`

```java
public void setReadonly(boolean readonly)

Set the value of the `readonly` property.
```

```java
public boolean isRedisplay()

redisplay
```

`isRedisplay`

```java
public boolean isRedisplay()

Return the value of the `redisplay` property.

Contents: Flag indicating that any existing value in this field should be rendered when the form is created. Because this is a potential
```
security risk, password values are not displayed by default.

public void setRedisplay(boolean redisplay)

   redisplay

setRedisplay

public void setRedisplay(boolean redisplay)

   Set the value of the redisplay property.

public int getSize()

   size

getSize

public int getSize()

   Return the value of the size property.

   Contents: The number of characters used to determine the width of this field.

public void setSize(int size)

   size
**setSize**

```java
class PublicMethod {  
    public void setSize(int size) {  
        Set the value of the size property.  
    }  
}
```

**getStyle**

```java
class PublicMethod {  
    public String getStyle() {  
        style  
        Return the value of the style property.  
        Contents: CSS style(s) to be applied when this component is rendered.  
    }  
}
```

**setStyle**

```java
class PublicMethod {  
    public void setStyle(String style) {  
        style  
        Set the value of the style property.  
    }  
}
```
public String getStyleClass()

    styleClass

CSS "class"

getStyleClass

public String getStyleClass()

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

    styleClass

setStyleClass

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

public String getTabindex()

    tabindex
public String getTabindex()

    Return the value of the tabindex property.
    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

public void setTabindex(String tabindex)
    tabindex

public String getTitle()
    title

gTitle

public String getTitle()
Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

---

**public void setTitle(String title)**

Set the value of the title property.

---

**public Object saveState(FacesContext _context)**

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:
Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:  
   saveState in interface StateHolder

Overrides:  
   saveState in class UIInput

public void restoreState(FacesContext _context, Object _state)

restoreState

public void restoreState(FacesContext _context,  
                         Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext,  
java.lang.Object) method on all those instances as well.

Specified by:  
   restoreState in interface StateHolder

Overrides:  
   restoreState in class UIInput
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html Class HtmlInputText

java.lang.Object  
  ↓ java.lang.Object  
  ↓ javax.faces.component.UIComponent  
  ↓ javax.faces.component.UIComponentBase  
  ↓ javax.faces.component.UIOutput  
  ↓ javax.faces.component.UIInput  
  ↓ javax.faces.component.html.HtmlInputText

All Implemented Interfaces: EditableValueHolder, StateHolder, ValueHolder

public class HtmlInputText
extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput

text HTML input

rendererType "javax.faces.Text" setRendererType()

 Represents an HTML input element of type text.

By default, the rendererType property must be set to "javax.faces.Text". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>

Fields inherited from class javax.faces.component.UIInput

COMPONENT_FAMILY, CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID,
### Fields inherited from class javax.faces.component.UIComponent bindings

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>getAccesskey</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td></td>
<td><strong>getAlt</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the alt property.</td>
</tr>
<tr>
<td></td>
<td><strong>getAutocomplete</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the autocomplete property.</td>
</tr>
<tr>
<td></td>
<td><strong>getDir</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td></td>
<td><strong>getLabel</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td></td>
<td><strong>getLang</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td></td>
<td><strong>getMaxlength</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the maxlength property.</td>
</tr>
<tr>
<td></td>
<td><strong>getOnblur</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td></td>
<td><strong>getOnchange</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the onchange property.</td>
</tr>
<tr>
<td></td>
<td><strong>getOnfocus</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td></td>
<td><strong>getOndblclick</strong>()</td>
</tr>
<tr>
<td></td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td></td>
<td><strong>getOnfocus</strong>()</td>
</tr>
<tr>
<td>Function</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the <code>onselect</code> property.</td>
</tr>
<tr>
<td><code>getSize()</code></td>
<td>Return the value of the <code>size</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTabIndex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>isDisabled()</code></td>
<td>Return the value of the <code>disabled</code> property.</td>
</tr>
<tr>
<td><code>isReadonly()</code></td>
<td>Return the value of the <code>readonly</code> property.</td>
</tr>
</tbody>
</table>

`restoreState(FacesContext _context, Object _state)`
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void Perform any processing required to restore the state from the entries in the state Object.</code></td>
<td></td>
</tr>
<tr>
<td><code>Object saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>void setAccesskey(String accesskey)</code></td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td><code>void setAlt(String alt)</code></td>
<td>Set the value of the alt property.</td>
</tr>
<tr>
<td><code>void setAutocomplete(String autocomplete)</code></td>
<td>Set the value of the autocomplete property.</td>
</tr>
<tr>
<td><code>void setDir(String dir)</code></td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td><code>void setDisabled(boolean disabled)</code></td>
<td>Set the value of the disabled property.</td>
</tr>
<tr>
<td><code>void setLabel(String label)</code></td>
<td>Set the value of the label property.</td>
</tr>
<tr>
<td><code>void setLang(String lang)</code></td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td><code>void setMaxlength(int maxlength)</code></td>
<td>Set the value of the maxlen property.</td>
</tr>
<tr>
<td><code>void setOnblur(String onblur)</code></td>
<td>Set the value of the onblur property.</td>
</tr>
<tr>
<td><code>void setOnchange(String onchange)</code></td>
<td>Set the value of the onchange property.</td>
</tr>
<tr>
<td><code>void setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td><code>void setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td><code>void setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td><code>void setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td><code>void setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>void setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td><code>void setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td><code>void setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td><code>void setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td><code>void setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><code>void setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>void setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>void setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>void setSize(int size)</code></td>
<td>Set the value of the size property.</td>
</tr>
<tr>
<td><code>void setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>void setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>void setTabIndex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><code>void setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UISupport`

- `addValidator`, `addValueChangeListener`, `compareValues`, `decode`, `getConvertedValue`, `getConverterMessage`, `getFamily`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListener`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `isRequired`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValid`, `setValidator`, `setValidator`
<table>
<thead>
<tr>
<th>Methods inherited from class <code>javax.faces.component.UIComponent</code>. <strong>UIOutput</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getConverter</code>, <code>getLocalValue</code>, <code>getValue</code>, <code>setConverter</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>javax.faces.component.UIComponentBase</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addFacesListener</code>, <code>broadcast</code>, <code>encodeBegin</code>, <code>encodeChildren</code>, <code>encodeEnd</code>, <code>findComponent</code>, <code>getAttributes</code>, <code>getChildCount</code>, <code>getChildren</code>, <code>getClientId</code>, <code>getFacesContext</code>, <code>getFacesListeners</code>, <code>getFacet</code>, <code>getFacetCount</code>, <code>getFacets</code>, <code>getFacetsAndChildren</code>, <code>getId</code>, <code>getParent</code>, <code>getRenderer</code>, <code>getRendererType</code>, <code>getRendersChildren</code>, <code>getValueBinding</code>, <code>invokeOnComponent</code>, <code>isRendered</code>, <code>isTransient</code>, <code>processRestoreState</code>, <code>processSaveState</code>, <code>queueEvent</code>, <code>removeFacesListener</code>, <code>restoreAttachedState</code>, <code>saveAttachedState</code>, <code>setId</code>, <code>setParent</code>, <code>setRendered</code>, <code>setRendererType</code>, <code>setTransient</code>, <code>setValueBinding</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>javax.faces.component.UIComponent</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>encodeAll</code>, <code>getContainerClientId</code>, <code>getValueExpression</code>, <code>setValueExpression</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>java.lang.Object</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>toString</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface <code>javax.faces.component.ValueHolder</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getConverter</code>, <code>getLocalValue</code>, <code>getValue</code>, <code>setConverter</code></td>
</tr>
</tbody>
</table>

**Field Detail**

COMPONENT_TYPE
public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail

public HtmlInputText()

HtmlInputText

public HtmlInputText()

Method Detail

public String getAccesskey()

accesskey

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.
public void setAccesskey(String accesskey)
    accesskey

setAccesskey

public void setAccesskey(String accesskey)
    Set the value of the accesskey property.

public String getAlt()
    alt

getAlt

public String getAlt()
    Return the value of the alt property.
    Contents: Alternate textual description of the element rendered by this component.

public void setAlt(String alt)
    alt
**setAlt**

```java
public void setAlt(String alt)
```

Set the value of the `alt` property.

---

**public String getAutocomplete()**

```java
autocomplete
"off" "off" "on"
```

---

**getAutocomplete**

```java
public String getAutocomplete()
```

Return the value of the `autocomplete` property.

Contents: If the value of this attribute is "off", render "off" as the value of the attribute. This indicates that the browser should disable its autocomplete feature for this component. This is useful for components that perform autocompletion and do not want the browser interfering. If this attribute is not set or the value is "on", render nothing.

---

**public void setAutocomplete(String autocomplete)**

```java
autocomplete
```

---

**setAutocomplete**

```java
public void setAutocomplete(String autocomplete)
```
Set the value of the `autocomplete` property.

```java
public String getDir()
{
    String dir = "LTR"; // left-to-right
    return dir;
}
```

**getDir**

```java
public String getDir()
{
    return "LTR"; // left-to-right
}
```

**Contents:** Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

```java
public void setDir(String dir)
{
    String dir = "LTR"; // left-to-right
}
```

**setDir**

```java
public void setDir(String dir)
{
    Set the value of the dir property.
}
```

```java
public boolean isDisabled()
{
    boolean disabled = true;
    return disabled;
}
```

**public boolean isDisabled()**
public boolean isDisabled()

    Return the value of the disabled property.

    Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled=\"disabled\".

public void setDisabled(boolean disabled)

disabled

public void setDisabled(boolean disabled)

    Set the value of the disabled property.

public String getLabel()

    label

getLabel
public String getLabel()

    Return the value of the label property.

    Contents: A localized user presentable name for this component.

public void setLabel(String label)

    label

setLabel

public void setLabel(String label)

    Set the value of the label property.

public String getLang()

    lang

getLang

public String getLang()

    Return the value of the lang property.

    Contents: Code describing the language used in the generated markup for this component.
public void setLang(String lang)

    lang

setLang

public void setLang(String lang)

    Set the value of the lang property.

public int getMaxlength()

    maxlength

getMaxlength

public int getMaxlength()

    Return the value of the maxLength property.

    Contents: The maximum number of characters that may be entered in this field.

public void setMaxlength(int maxLength)

    maxLength
setMaxlength

public void setMaxlength(int maxlength)

    Set the value of the maxlen property.

-----------------------------------------------

getOnblur

public String getOnblur()

    onblur

Javascript

getOnblur

public String getOnblur()

    Return the value of the onblur property.
    Contents: Javascript code executed when this element loses focus.

-----------------------------------------------

setOnblur

public void setOnblur(String onblur)

    onblur

setOnblur

public void setOnblur(String onblur)

    Set the value of the onblur property.

-----------------------------------------------

public String getOnchange()
getOnchange

public String getOnchange()

Return the value of the onchange property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

public void setOnchange(String onchange)

setOnchange

public void setOnchange(String onchange)

Set the value of the onchange property.

public String getOnclick()

getOnclick

Javascrip
getOnclick

public String getOnclick()

    Return the value of the onclick property.

    Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

    onclick

setOnclick

public void setOnclick(String onclick)

    Set the value of the onclick property.

public String getOndblclick()

    ondblclick

Javascript

getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.

    Contents: Javascript code executed when a pointer button is double
public void setOndblclick(String ondblclick)

    ondblclick

setOndblclick

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

public String getOnfocus()

    onfocus

Javascript

getOnfocus

public String getOnfocus()

    Return the value of the onfocus property.

    Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onfocus)

    onfocus
setOnfocus

public void setOnfocus(String onfocus)

Set the value of the onfocus property.

public String getOnkeydown()

onkeydown

Javascript

getOnkeydown

public String getOnkeydown()

Return the value of the onkeydown property.

Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

onkeydown

setOnkeydown

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.
public String getOnkeypress()

   onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

   Return the value of the onkeypress property.

   Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

   onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

   Set the value of the onkeypress property.

public String getOnkeyup()

   onkeyup

Javascript
**getOnkeyup**

```java
public String getOnkeyup()
```

Return the value of the `onkeyup` property.

Contents: Javascript code executed when a key is released over this element.

**public void setOnkeyup(String onkeyup)**

```java
onkeyup
```

**setOnkeyup**

```java
public void setOnkeyup(String onkeyup)
```

Set the value of the `onkeyup` property.

**public String getOnmousedown()**

```java
onmousedown
```

**Javascrip**

**getOnmousedown**

```java
public String getOnmousedown()
```

Return the value of the `onmousedown` property.
Contents: Javascript code executed when a pointer button is pressed down over this element.

```java
public void setOnmousedown(String onmousedown)

  onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

  Set the value of the onmousedown property.

public String getOnmousemove()

  onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

  Return the value of the onmousemove property.

  Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

  onmousemove
```
**setOnmousemove**

public void **setOnmousemove**(String onmousemove)

Set the value of the **onmousemove** property.

---

**public String getOnmouseout()**

**onmouseout**

**Javascript**

**getOnmouseout**

public **String** **getOnmouseout**()

Return the value of the **onmouseout** property.

Contents: Javascript code executed when a pointer button is moved away from this element.

---

**public void setOnmouseout(String onmouseout)**

**onmouseout**

**setOnmouseout**

public void **setOnmouseout**(String onmouseout)

Set the value of the **onmouseout** property.
public String getOnmouseover()

  onmouseover

Javascript

getOnmouseover

public String getOnmouseover()

  Return the value of the onmouseover property.

  Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

  onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

  Set the value of the onmouseover property.

public String getOnmouseup()

  onmouseup

Javascript
**getOnmouseup**

```java
public String getOnmouseup()
```

Return the value of the `onmouseup` property.

Contents: Javascript code executed when a pointer button is released over this element.

---

**public void setOnmouseup(String onmouseup)**

```java
onmouseup
```

---

**setOnmouseup**

```java
public void setOnmouseup(String onmouseup)
```

Set the value of the `onmouseup` property.

---

**public String getOnselect()**

```java
onselect
```

**Javascript**

---

**getOnselect**

```java
public String getOnselect()
```

Return the value of the `onselect` property.
Contents: Javascript code executed when text within this element is selected by the user.

```java
public void setOnselect(String onSelect)
    onSelect

setOnselect
public void setOnselect(String onSelect)
    Set the value of the onSelect property.

public boolean isReadonly()
    readonly

false true readonly="readonly"

isReadonly
public boolean isReadonly()
    Return the value of the readonly property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".
public void setReadonly(boolean readonly)

    readonly

setReadonly

public void setReadonly(boolean readonly)

    Set the value of the readonly property.

public int getSize()

    size

getSize

public int getSize()

    Return the value of the size property.

    Contents: The number of characters used to determine the width of this field.

public void setSize(int size)

    size
**setSize**

```java
public void setSize(int size)
```

Set the value of the size property.

---

**public String getStyle()**

**style**

CSS

**getStyle**

```java
public String getStyle()
```

Return the value of the style property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

**style**

**setStyle**

```java
public void setStyle(String style)
```

Set the value of the style property.
public String getStyleClass()

    styleClass

CSS "class"

getStyleClass

public String getStyleClass()

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

    styleClass

setStyleClass

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

public String getTabindex()

    tabindex

Tab 0 32767
**getTabindex**

public **String** getTabindex()

Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

---

**public void setTabindex(String tabindex)**

tabindex

**setTabindex**

public void **setTabindex**(String `tabindex`)

Set the value of the `tabindex` property.

---

**public String getTitle()**

title

**getTitle**

public **String** getTitle()

Return the value of the `title` property.
Contents: Advisory title information about markup elements generated for this component.

---

public void setTitle(String title)

title

setTitle

public void setTitle(String title)

Set the value of the title property.

---

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);
component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
   saveState in interface StateHolder

Overrides:
   saveState in class UIInput

public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIClientComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
   restoreState in interface StateHolder

Overrides:
   restoreState in class UIInput
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html Class HtmlInputTextarea

java.lang.Object
   ▼ javax.faces.component.UIComponent
       ▼ javax.faces.component.UIComponentBase
           ▼ javax.faces.component.UIOutput
               ▼ javax.faces.component.UIInput
                   ▼ javax.faces.component.html.HtmlInputTextarea

All Implemented Interfaces:
   EditableValueHolder, StateHolder, ValueHolder

public class HtmlInputTextarea
    extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput

HTML textarea

    rendererType "javax.faces.Textarea"  setRendererType()

Represents an HTML textarea element.

By default, the rendererType property must be set to "javax.faces.Textarea". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>Component Type</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>

Fields inherited from class javax.faces.component.UIInput
Fields inherited from class javax.faces.component.UIComponent
bindings

## Constructor Summary

```java
HtmlInputTextarea()
```

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAccesskey()</code></td>
<td>String</td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td><code>getCols()</code></td>
<td>int</td>
<td>Return the value of the cols property.</td>
</tr>
<tr>
<td><code>getDir()</code></td>
<td>String</td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td><code>getLabel()</code></td>
<td>String</td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>String</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td><code>getOnblur()</code></td>
<td>String</td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td><code>getOnchange()</code></td>
<td>String</td>
<td>Return the value of the onchange property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>String</td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>String</td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>String</td>
<td>Return the value of the onfocus property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>String</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the onkeypress property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the onkeyup property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the onmousedown property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the onmousemove property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the onmouseout property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the onmouseover property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the onmouseup property.</td>
<td></td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the onselect property.</td>
<td></td>
</tr>
<tr>
<td><code>getRows()</code></td>
<td>Return the value of the rows property.</td>
<td></td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the style property.</td>
<td></td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the styleClass property.</td>
<td></td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the tabindex property.</td>
<td></td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the title property.</td>
<td></td>
</tr>
<tr>
<td><code>isDisabled()</code></td>
<td>Return the value of the disabled property.</td>
<td></td>
</tr>
<tr>
<td><code>isReadonly()</code></td>
<td>Return the value of the readonly property.</td>
<td></td>
</tr>
<tr>
<td><code>restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
<td></td>
</tr>
<tr>
<td><code>saveState(FacesContext _context)</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><code>getAccesskey()</code></td>
<td>Gets the state of the instance as a <code>Serializable</code> Object.</td>
<td></td>
</tr>
<tr>
<td><code>setAccesskey(String accesskey)</code></td>
<td>Set the value of the <code>accesskey</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setCols(int cols)</code></td>
<td>Set the value of the <code>cols</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setDir(String dir)</code></td>
<td>Set the value of the <code>dir</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setDisabled(boolean disabled)</code></td>
<td>Set the value of the <code>disabled</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setLabel(String label)</code></td>
<td>Set the value of the <code>label</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setLang(String lang)</code></td>
<td>Set the value of the <code>lang</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnblur(String onblur)</code></td>
<td>Set the value of the <code>onblur</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnchange(String onchange)</code></td>
<td>Set the value of the <code>onchange</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the <code>onfocus</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the <code>onkeydown</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the <code>onkeypress</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the <code>onkeyup</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the <code>onmousedown</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the <code>onmousemove</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the <code>onmouseout</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the <code>onmouseover</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the <code>onmousemove</code> property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the <code>onmouseout</code> property.</td>
<td></td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>setRows(int rows)</code></td>
<td>Set the value of the rows property.</td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

#### Methods inherited from class `javax.faces.component.UIInput`

- `addValidator`, `addValueChangeListener`, `compareValues`, `decode`, `getConvertedValue`, `getConverterMessage`, `getFamily`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValid`, `setValidator`, `setValidatorMessage`, `setTitle`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`, `validateValue`

#### Methods inherited from class `javax.faces.component.UIOutput`

- `getConverter`, `getLocalValue`, `getValue`, `setConverter`
Methods inherited from class javax.faces.component.UIComponentBase
addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getID, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.faces.component.ValueHolder
getConverter, getLocalValue, getValue, setConverter

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values
Constructor Detail

public HtmlInputTextarea()

HtmlInputTextarea

public HtmlInputTextarea()

Method Detail

public String getAccesskey()

accesskey

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

public void setAccesskey(String accesskey)

accesskey

setAccesskey
public void setAccesskey(String accesskey)

    Set the value of the accesskey property.

public int getCols()

cols

getCols

public int getCols()

    Return the value of the cols property.

    Contents: The number of columns to be displayed.

public void setCols(int cols)

cols

setCols

public void setCols(int cols)

    Set the value of the cols property.

public String getDir()

dir
getDir

```java
public String getDir()

    Return the value of the dir property.
```

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

---

public void setDir(String dir)

dir

---

setDir

```java
public void setDir(String dir)

    Set the value of the dir property.
```

---

public boolean isDisabled()

```java
    disabled

false true disabled="disabled"
```

---

isDisabled
public boolean isDisabled()

    Return the value of the disabled property.

    Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled

setDisabled

public void setDisabled(boolean disabled)

    Set the value of the disabled property.

public String getLabel()

    label

getLabel

public String getLabel()

    Return the value of the label property.

    Contents: A localized user presentable name for this component.
public void setLabel(String label)

setLabel

public void setLabel(String label)

Set the value of the label property.

public String getLang()

getLang

public String getLang()

Return the value of the lang property.

Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

setLang

public void setLang(String lang)

lang
**setLang**

```java
public void setLang(String lang)
```

Set the value of the `lang` property.

**public String getOnblur()**

```java
onblur
```

**Javascript**

**getOnblur**

```java
public String getOnblur()
```

Return the value of the `onblur` property.

Contents: Javascript code executed when this element loses focus.

**public void setOnblur(String onblur)**

```java
onblur
```

**setOnblur**

```java
public void setOnblur(String onblur)
```

Set the value of the `onblur` property.

**public String getOnchange()**
onchange

Javascript

getOnchange

public String getOnchange()

   Return the value of the onchange property.
   Contents: Javascript code executed when this element loses focus
   and its value has been modified since gaining focus.

public void setOnchange(String onchange)

   onchange

setOnchange

public void setOnchange(String onchange)

   Set the value of the onchange property.

public String getOnclick()

   onclick

Javascript
**getOnclick**

```java
public String getOnclick()
```

Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.

---

**public void setOnclick(String onclick)**

`onclick`

---

**setOnclick**

```java
public void setOnclick(String onclick)
```

Set the value of the onclick property.

---

**public String getOndblclick()**

`ondblclick`

**Javascript**

---

**getOndblclick**

```java
public String getOndblclick()
```

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double
public void setOnDbLclick(String onDbLclick)

  onDbLclick

setOnDbLclick

public void setOnDbLclick(String onDbLclick)

  Set the value of the onDbLclick property.

public String getOnfocus()

  onFocus

Javascript

getOnfocus

public String getOnfocus()

  Return the value of the onFocus property.

  Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onFocus)

  onFocus
**setOnfocus**

```java
public void setOnfocus(String onFocus)
```

Set the value of the `onfocus` property.

---

**public String getOnkeydown()**

`onkeydown`

**Javascript**

**getOnkeydown**

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

---

**public void setOnkeydown(String onkeydown)**

`onkeydown`

**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.
public String getOnkeypress()

   onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

    Return the value of the onkeypress property.

    Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

   onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

    Set the value of the onkeypress property.

public String getOnkeyup()

   onkeyup

Javascript
getOnkeyup

public String getOnkeyup()

Return the value of the onkeyup property.

Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

Set the value of the onkeyup property.

public String getOnmousedown()

onmousedown

Javascript

getOnmousedown

public String getOnmousedown()

Return the value of the onmousedown property.
Contents: Javascript code executed when a pointer button is pressed down over this element.

```java
public void setOnmousedown(String onmousedown)
    onmousedown
```

`setOnmousedown`

```java
public void setOnmousedown(String onmousedown)
    Set the value of the onmousedown property.
```

```java
public String getOnmousemove()
    onmousemove
```

`getOnmousemove`

```java
public String getOnmousemove()
    Return the value of the onmousemove property.
    Contents: Javascript code executed when a pointer button is moved within this element.
```

```java
public void setOnmousemove(String onmousemove)
    onmousemove
```

`setOnmousemove`
setOnmousemove

public void setOnmousemove(String onmousemove)

    Set the value of the onmousemove property.

getOnmouseout

public String getOnmouseout()

    onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

    Return the value of the onmouseout property.

    Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

    onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

    Set the value of the onmouseout property.
public String getOnmouseover()

    onmouseover

Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

public String getOnmouseup()

    onmouseup

Javascript
getOnmouseup

public String getOnmouseup()

Return the value of the onmouseup property.

Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

Set the value of the onmouseup property.

public String getOnselect()

onselect

Javascript

getOnselect

public String getOnselect()

Return the value of the onselect property.
Contents: Javascript code executed when text within this element is selected by the user.

```java
public void setOnselect(String onSelect)
    onSelect
```

setOnselect

```java
public void setOnselect(String onSelect)
    Set the value of the onSelect property.
```

```java
public boolean isReadonly()
    readonly

false true readonly="readonly"
```

isReadonly

```java
public boolean isReadonly()
    Return the value of the readonly property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".
```
public void setReadonly(boolean readonly)

    readonly

setReadonly

public void setReadonly(boolean readonly)
    Set the value of the readonly property.

public int getRows()

    rows

getRows

public int getRows()
    Return the value of the rows property.
    Contents: The number of rows to be displayed.

public void setRows(int rows)

    rows

setRows
public void setRows(int rows)
   
   Set the value of the rows property.

public String getStyle()
   
   style

CSS

getStyle

public String getStyle()
   
   Return the value of the style property.
   
   Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)
   
   style

setStyle

public void setStyle(String style)
   
   Set the value of the style property.

public String getStyleClass()
getStyleClass

public String getStyleClass()

Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

setStyleClass

public void setStyleClass(String styleClass)

Set the value of the styleClass property.

public String getTabindex()

tabindex

Tab 0 32767
getTabindex

public String getTabindex()

    Return the value of the \texttt{tabindex} property.

    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

_____________________________

public void setTabindex(String tabindex)

    \texttt{tabindex}

_____________________________

setTabindex

public void setTabindex(String \texttt{tabindex})

    Set the value of the \texttt{tabindex} property.

_____________________________

public String getTitle()

    \texttt{title}

_____________________________

getTitle

public String getTitle()

    Return the value of the \texttt{title} property.

    Contents: Advisory title information about markup elements
public void setTitle(String title)

title

setTitle

public void setTitle(String title)

Set the value of the title property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);
component should be the same as before executing it.

The return from this method must be `Serializable`

Specified by:
`saveState` in interface `StateHolder`

Overrides:
`saveState` in class `UIInput`

---

```java
public void restoreState(FacesContext _context, Object _state)
```

**restoreState**

```java
public void restoreState(FacesContext _context, Object _state)
```

Description copied from interface: `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UICOMPONENT with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

Specified by:
`restoreState` in interface `StateHolder`

Overrides:
`restoreState` in class `UIInput`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html  Class HtmlMessage

java.lang.Object
  ↓ javax.faces.component.UIComponent
      ↓ javax.faces.component.UIComponentBase
          ↓ javax.faces.component.UIMessage
              ↓ javax.faces.component.html.HtmlMessage

All Implemented Interfaces:
  StateHolder

public class HtmlMessage
  extends UIMessage

Extends: UIComponent > UIComponentBase > UIMessage

rendererType "javax.faces.Message"  setRendererType()

By default, the rendererType property must be set to "javax.faces.Message". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>

Fields inherited from class javax.faces.component.UIMessage
COMPONENT_FAMILY

Fields inherited from class javax.faces.component.UIComponent
bindings
### Constructor Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Constructor Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>HtmlMessage</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getDir()</td>
</tr>
<tr>
<td>Return the value of the dir property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getErrorClass()</td>
</tr>
<tr>
<td>Return the value of the errorClass property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getErrorStyle()</td>
</tr>
<tr>
<td>Return the value of the errorStyle property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getFatalClass()</td>
</tr>
<tr>
<td>Return the value of the fatalError property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getFatalStyle()</td>
</tr>
<tr>
<td>Return the value of the fatalError property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getInfoClass()</td>
</tr>
<tr>
<td>Return the value of the infoClass property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getInfoStyle()</td>
</tr>
<tr>
<td>Return the value of the infoStyle property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getLang()</td>
</tr>
<tr>
<td>Return the value of the lang property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getStyle()</td>
</tr>
<tr>
<td>Return the value of the style property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getStyleClass()</td>
</tr>
<tr>
<td>Return the value of the styleClass property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getTitle()</td>
</tr>
<tr>
<td>Return the value of the title property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getWarnClass()</td>
</tr>
<tr>
<td>Return the value of the warnClass property.</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>getWarnStyle()</td>
</tr>
<tr>
<td>Return the value of the warnStyle property.</td>
<td></td>
</tr>
<tr>
<td>boolean</td>
<td>isTooltip()</td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>Object saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>void setDir(String dir)</code></td>
<td>Set the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>void setErrorClass(String errorClass)</code></td>
<td>Set the value of the <code>errorClass</code> property.</td>
</tr>
<tr>
<td><code>void setErrorStyle(String errorStyle)</code></td>
<td>Set the value of the <code>errorStyle</code> property.</td>
</tr>
<tr>
<td><code>void setFatalClass(String fatalClass)</code></td>
<td>Set the value of the <code>fatalClass</code> property.</td>
</tr>
<tr>
<td><code>void setFatalStyle(String fatalStyle)</code></td>
<td>Set the value of the <code>fatalStyle</code> property.</td>
</tr>
<tr>
<td><code>void setInfoClass(String infoClass)</code></td>
<td>Set the value of the <code>infoClass</code> property.</td>
</tr>
<tr>
<td><code>void setInfoStyle(String infoStyle)</code></td>
<td>Set the value of the <code>infoStyle</code> property.</td>
</tr>
<tr>
<td><code>void setLang(String lang)</code></td>
<td>Set the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>void setStyle(String style)</code></td>
<td>Set the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>void setStyleClass(String styleClass)</code></td>
<td>Set the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>void setTitle(String title)</code></td>
<td>Set the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>void setTooltip(boolean tooltip)</code></td>
<td>Set the value of the <code>tooltip</code> property.</td>
</tr>
<tr>
<td><code>void setWarnClass(String warnClass)</code></td>
<td>Set the value of the <code>warnClass</code> property.</td>
</tr>
<tr>
<td><code>void setWarnStyle(String warnStyle)</code></td>
<td>Set the value of the <code>warnStyle</code> property.</td>
</tr>
</tbody>
</table>
Methods inherited from class `javax.faces.component.UIMessage`

- `getFamily`
- `getFor`
- `isShowDetail`
- `isShowSummary`
- `setFor`
- `setShowDetail`
- `setShowSummary`

Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`
- `broadcast`
- `decode`
- `encodeBegin`
- `encodeChildren`
- `encodeEnd`
- `findComponent`
- `getAttributes`
- `getChildCount`
- `getChildren`
- `getClientId`
- `getFacesContext`
- `getFacesListeners`
- `getFacet`
- `getFacetCount`
- `getFacets`
- `getFacetsAndChildren`
- `getId`
- `getParent`
- `getRenderer`
- `getRendererType`
- `getRendersChildren`
- `getValueBinding`
- `invokeOnComponent`
- `isRendered`
- `isTransient`
- `processDecodes`
- `processRestoreState`
- `processSaveState`
- `processUpdates`
- `processValidators`
- `queueEvent`
- `removeFacesListener`
- `restoreAttachedState`
- `saveAttachedState`
- `setId`
- `setParent`
- `setRendered`
- `setRendererType`
- `setTransient`
- `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`
- `getContainerClientId`
- `getValueExpression`
- `setValueExpression`

Methods inherited from class `java.lang.Object`

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `toString`
- `wait`
- `wait`
- `wait`

Field Detail

**COMPONENT_TYPE**

- `public static final String COMPONENT_TYPE`

  The standard component type for this component.

  **See Also:**
Constant Field Values

Constructor Detail

public HtmlMessage()

HtmlMessage

public HtmlMessage()

Method Detail

public String getDir()  
  
  "LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()  
  
  Return the value of the dir property.
  
  Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

public void setDir(String dir)  
  
  dir
**setDir**

```java
public void setDir(String dir)
```

Set the value of the `dir` property.

---

**public String getErrorClass()**

```java
errorClass
"ERROR" CSS
```

**getErrorClass**

```java
public String getErrorClass()
```

Return the value of the `errorClass` property.

Contents: CSS style class to apply to any message with a severity class of "ERROR".

---

**public void setErrorClass(String errorClass)**

```java
errorClass
```

**setErrorClass**

```java
public void setErrorClass(String errorClass)
```

Set the value of the `errorClass` property.
public String getErrorStyle()

    errorStyle

"ERROR" CSS

g.getErrorStyle

public String getErrorStyle()

    Return the value of the errorStyle property.

    Contents: CSS style(s) to apply to any message with a severity class of "ERROR".

public void setErrorStyle(String errorStyle)

    errorStyle

setErrorStyle

public void setErrorStyle(String errorStyle)

    Set the value of the errorStyle property.

public String getFatalClass()

    fatalClass

"FATAL" CSS
getFatalClass

public String getFatalClass()

Return the value of the fatalClass property.

Contents: CSS style class to apply to any message with a severity class of "FATAL".

---------------------------------------------

public void setFatalClass(String fatalClass)

fatalClass

setFatalClass

public void setFatalClass(String fatalClass)

Set the value of the fatalClass property.

---------------------------------------------

public String getFatalStyle()

fatalStyle

"FATAL" CSS

getFatalStyle

public String getFatalStyle()

Return the value of the fatalStyle property.
Contents: CSS style(s) to apply to any message with a severity class of "FATAL".

public void setFatalStyle(String fatalStyle)

    fatalError

setFatalStyle

public void setFatalStyle(String fatalStyle)

    Set the value of the fatalError property.

public String getInfoClass()

    infoClass

"INFO" CSS

getInfoClass

public String getInfoClass()

    Return the value of the infoClass property.

    Contents: CSS style class to apply to any message with a severity class of "INFO".

public void setInfoClass(String infoClass)

    infoClass
setInfoClass

public void setInfoClass(String infoClass)

    Set the value of the infoClass property.

public String getInfoStyle()

    infoStyle

"INFO" CSS

getInfoStyle

public String getInfoStyle()

    Return the value of the infoStyle property.

    Contents: CSS style(s) to apply to any message with a severity class of "INFO".

public void setInfoStyle(String infoStyle)

    infoStyle

setInfoStyle

public void setInfoStyle(String infoStyle)

    Set the value of the infoStyle property.
public String getLang()

    lang

getLang

public String getLang()

    Return the value of the lang property.

    Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

    lang

setLang

public void setLang(String lang)

    Set the value of the lang property.

public String getStyle()

    style

CSS
**getStyle**

```java
public String getStyle()
```

Return the value of the `style` property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

- `style`

---

**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.

---

**public String getStyleClass()**

- `styleClass`

**CSS "class"**

---

**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.
Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

```
public void setStyleClass(String styleClass)

styleClass
```

```
setStyleClass

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.
```

```
public String getTitle()

title
```

```
getTitle

public String getTitle()

    Return the value of the title property.

    Contents: Advisory title information about markup elements generated for this component.
```

```
public void setTitle(String title)
```
setTitle

public void setTitle(String title)

Set the value of the title property.

public boolean isTooltip()

tooltip

isTooltip

public boolean isTooltip()

Return the value of the tooltip property.

Contents: Flag indicating whether the detail portion of the message should be displayed as a tooltip.

public void setTooltip(boolean tooltip)

tooltip

setTooltip

public void setTooltip(boolean tooltip)
Set the value of the `tooltip` property.

public String getWarnClass()

    warnClass

"WARN" CSS

getWarnClass

public String getWarnClass()

    Return the value of the `warnClass` property.

    Contents: CSS style class to apply to any message with a severity class of "WARN".

public void setWarnClass(String warnClass)

    warnClass

setWarnClass

public void setWarnClass(String warnClass)

    Set the value of the `warnClass` property.

public String getWarnStyle()

    warnStyle
"WARN" CSS

getWarnStyle

public String getWarnStyle()

    Return the value of the warnStyle property.

    Contents: CSS style(s) to apply to any message with a severity class of "WARN".

-----------------------------------------------

public void setWarnStyle(String warnStyle)

    warnStyle

-----------------------------------------------

setWarnStyle

public void setWarnStyle(String warnStyle)

    Set the value of the warnStyle property.

-----------------------------------------------

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

    Description copied from interface: StateHolder

    Gets the state of the instance as a Serializable Object.
If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIMessage`

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface:** `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the
StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
   restoreState in interface StateHolder

Overrides:
   restoreState in class UIMessage
javax.faces.component.html Class HtmlMessages

java.lang.Object
    ▼ javax.faces.component.UIComponent
        ▼ javax.faces.component.UIComponentBase
            ▼ javax.faces.component.UIMessages
                ▼ javax.faces.component.html.HtmlMessages

All Implemented Interfaces:
    StateHolder

public class HtmlMessages
    extends UIMessages

Extends: UIComponent > UIComponentBase > UIMessages

    rendererType "javax.faces.Messages"    setRendererType()

By default, the rendererType property must be set to "javax.faces.Messages". This value can be changed by calling the setRendererType() method.

---

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String COMPONENT_TYPE</td>
</tr>
</tbody>
</table>

| Fields inherited from class javax.faces.component.UIMessages |
| COMPONENT_FAMILY |

| Fields inherited from class javax.faces.component.UIComponent |
| bindings |
## Constructor Summary

**HtmlMessages()**

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getDir()</strong></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td><strong>getErrorCode()</strong></td>
<td>Return the value of the errorClass property.</td>
</tr>
<tr>
<td><strong>getErrorStyle()</strong></td>
<td>Return the value of the errorStyle property.</td>
</tr>
<tr>
<td><strong>getFatalClass()</strong></td>
<td>Return the value of the fatalClass property.</td>
</tr>
<tr>
<td><strong>getFatalStyle()</strong></td>
<td>Return the value of the fatalStyle property.</td>
</tr>
<tr>
<td><strong>getInfoClass()</strong></td>
<td>Return the value of the infoClass property.</td>
</tr>
<tr>
<td><strong>getInfoStyle()</strong></td>
<td>Return the value of the infoStyle property.</td>
</tr>
<tr>
<td><strong>getLang()</strong></td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td><strong>getLayout()</strong></td>
<td>Return the value of the layout property.</td>
</tr>
<tr>
<td><strong>getStyle()</strong></td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td><strong>getStyleClass()</strong></td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td><strong>getTitle()</strong></td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td><strong>getWarnClass()</strong></td>
<td>Return the value of the warnClass property.</td>
</tr>
<tr>
<td><strong>getWarnStyle()</strong></td>
<td>Return the value of the warnStyle property.</td>
</tr>
</tbody>
</table>
**Return the value of the warnStyle property.**

```java
boolean isTooltip()
    Return the value of the tooltip property.
```

**Perform any processing required to restore the state from the entries in the state Object.**

```java
void restoreState(FacesContext _context, Object _state)
```

**Gets the state of the instance as a Serializable Object.**

```java
Object saveState(FacesContext _context)
```

**Set the value of the dir property.**

```java
void setDir(String dir)
```

**Set the value of the errorClass property.**

```java
void setErrorClass(String errorClass)
```

**Set the value of the errorStyle property.**

```java
void setErrorStyle(String errorStyle)
```

**Set the value of the fatalClass property.**

```java
void setFatalClass(String fatalClass)
```

**Set the value of the fatalStyle property.**

```java
void setFatalStyle(String fatalStyle)
```

**Set the value of the infoClass property.**

```java
void setInfoClass(String infoClass)
```

**Set the value of the infoStyle property.**

```java
void setInfoStyle(String infoStyle)
```

**Set the value of the lang property.**

```java
void setLang(String lang)
```

**Set the value of the layout property.**

```java
void setLayout(String layout)
```

**Set the value of the style property.**

```java
void setStyle(String style)
```

**Set the value of the styleClass property.**

```java
void setStyleClass(String styleClass)
```

**Set the value of the title property.**

```java
void setTitle(String title)
```

**Set the value of the tooltip property.**

```java
void setTooltip(boolean tooltip)
```
<table>
<thead>
<tr>
<th>void</th>
<th><code>setWarnClass(String warnClass)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set the value of the warnClass property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th><code>setWarnStyle(String warnStyle)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set the value of the warnStyle property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UIMessages`

- `getFamily`, `isGlobalOnly`, `isShowDetail`, `isShowSummary`, `setGlobalOnly`, `setShowDetail`, `setShowSummary`

Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

COMPONENT_TYPE
public static final String COMPONENT_TYPE

   The standard component type for this component.

See Also:
   Constant Field Values

Constructor Detail

public HtmlMessages()

HtmlMessages

public HtmlMessages()

Method Detail

public String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
public void setDir(String dir)

    dir

setDir

public void setDir(String dir)

    Set the value of the dir property.

public String getErrorClass()

    errorClass

"ERROR" CSS

getErrorClass

public String getErrorClass()

    Return the value of the errorClass property.

    Contents: CSS style class to apply to any message with a severity class of "ERROR".

public void setErrorClass(String errorClass)

    errorClass
**setErrorClass**

public void setErrorClass(String errorClass)

Set the value of the `errorClass` property.

---

**public String getErrorStyle()**

`errorStyle`

"ERROR" CSS

---

**getErrorStyle**

public String getErrorStyle()

Return the value of the `errorStyle` property.

Contents: CSS style(s) to apply to any message with a severity class of "ERROR".

---

**public void setErrorStyle(String errorStyle)**

`errorStyle`

---

**setErrorStyle**

public void setErrorStyle(String errorStyle)

Set the value of the `errorStyle` property.
public String getFatalClass()

    fatalClass

"FATAL" CSS

getFatalClass

public String getFatalClass()

    Return the value of the fatalClass property.

   Contents: CSS style class to apply to any message with a severity class of "FATAL".

public void setFatalClass(String fatalClass)

    fatalClass

setFatalClass

public void setFatalClass(String fatalClass)

    Set the value of the fatalClass property.

public String getFatalStyle()

    fatalStyle

"FATAL" CSS
**getFatalStyle**

```java
public String getFatalStyle()
```

Return the value of the `fatalStyle` property.

Contents: CSS style(s) to apply to any message with a severity class of "FATAL".

---

**public void setFatalStyle(String fatalStyle)**

```java

fatalStyle
```

---

**setFatalStyle**

```java
public void setFatalStyle(String fatalStyle)
```

Set the value of the `fatalStyle` property.

---

**public String getInfoClass()**

```java

"INFO" CSS
```

---

**getInfoClass**

```java
public String getInfoClass()
```

Return the value of the `infoClass` property.
Contents: CSS style class to apply to any message with a severity class of "INFO".

public void setInfoClass(String infoClass)

  infoClass

setInfoClass

public void setInfoClass(String infoClass)

  Set the value of the infoClass property.

public String getInfoStyle()

  infoStyle

"INFO"  CSS

getInfoStyle

public String getInfoStyle()

  Return the value of the infoStyle property.

  Contents: CSS style(s) to apply to any message with a severity class of "INFO".

public void setInfoStyle(String infoStyle)

  infoStyle
**setInfoStyle**

public void setInfoStyle(String infoStyle)

Set the value of the infoStyle property.

---

**public String getLang()**

lang

---

**getLang**

public String getLang()

Return the value of the lang property.

Contents: Code describing the language used in the generated markup for this component.

---

**public void setLang(String lang)**

lang

---

**setLang**

public void setLang(String lang)

Set the value of the lang property.
public String getLayout()

    Return the value of the layout property.

    Contents: The type of layout markup to use when rendering error messages. Valid values are "table" (an HTML table) and "list" (an HTML list). If not specified, the default value is "list".

public void setLayout(String layout)

    Set the value of the layout property.

public String getStyle()
**getStyle**

```java
public String getStyle()
```

Return the value of the `style` property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

`style`

---

**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.

---

**public String getStyleClass()**

`styleClass`

CSS "class"

---

**getStyleClass**

```java
public String getStyleClass()
```
Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

```java
public void setStyleClass(String styleClass)

    styleClass
```

**setStyleClass**

```java
public void setStyleClass(String styleClass)

    Set the value of the styleClass property.
```

```java
public String getTitle()

    title
```

**getTitle**

```java
public String getTitle()

    Return the value of the title property.

    Contents: Advisory title information about markup elements generated for this component.
```
public void setTitle(String title)

  title

setTitle

public void setTitle(String title)

  Set the value of the title property.

public boolean isTooltip()

  tooltip

isTooltip

public boolean isTooltip()

  Return the value of the tooltip property.

  Contents: Flag indicating whether the detail portion of the message should be displayed as a tooltip.

public void setTooltip(boolean tooltip)

  tooltip

setTooltip
**setTooltip**

```java
public void setTooltip(boolean tooltip)
    Set the value of the tooltip property.
```

---

**public String getWarnClass()**

```java
public String getWarnClass()
    warnClass
    "WARN"  CSS
```

---

**getWarnClass**

```java
public String getWarnClass()
    Return the value of the warnClass property.
    Contents: CSS style class to apply to any message with a severity class of "WARN".
```

---

**public void setWarnClass(String warnClass)**

```java
public void setWarnClass(String warnClass)
    warnClass
```

---

**setWarnClass**

```java
public void setWarnClass(String warnClass)
    Set the value of the warnClass property.
```
public String getWarnStyle()

    warnStyle

"WARN"  CSS

getWarnStyle

public String getWarnStyle()

    Return the value of the warnStyle property.

    Contents: CSS style(s) to apply to any message with a severity class of "WARN".

public void setWarnStyle(String warnStyle)

    warnStyle

setWarnStyle

public void setWarnStyle(String warnStyle)

    Set the value of the warnStyle property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)
**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIMessages`

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface: StateHolder**

Perform any processing required to restore the state from the entries in the state Object.
If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
- `restoreState` in interface `StateHolder`

**Overrides:**
- `restoreState` in class `UIMessages`
javax.faces.component.html  Class HtmlOutputFormat

java.lang.Object  
  ▼ javax.faces.component.UIComponent  
      ▼ javax.faces.component.UIComponentBase  
          ▼ javax.faces.component.UIOutput  
              ▼ javax.faces.component.html.HtmlOutputFormat

All Implemented Interfaces:
  StateHolder, ValueHolder

public class HtmlOutputFormat
extends UIOutput

Extends: UIComponent > UIComponentBase > UIOutput

  MessageFormat  UIParameter "dir" "lang"  
  span  span

  rendererType "javax.faces.Format"  setRendererType()

Represents a component that looks up a localized message in a resource bundle, optionally uses it as a MessageFormat pattern string and substitutes in parameter values from nested UIParameter components, and renders the result. If the "dir" or "lang" attributes are present, render a span element and pass them through as attributes on the span.

By default, the rendererType property must be set to "javax.faces.Format". This value can be changed by calling the setRendererType() method.

---

**Field Summary**
### static String COMPONENT_TYPE
The standard component type for this component.

### Fields inherited from class javax.faces.component.UIOutput
COMPONENT_FAMILY

### Fields inherited from class javax.faces.component.UIComponent
bindings

### Constructor Summary

- **HtmlOutputFormat()**

### Method Summary

- **String getDir()**
  Return the value of the dir property.

- **String getLang()**
  Return the value of the lang property.

- **String getStyle()**
  Return the value of the style property.

- **String getStyleClass()**
  Return the value of the styleClass property.

- **String getTitle()**
  Return the value of the title property.

- **boolean isEscape()**
  Return the value of the escape property.

- **void restoreState(FacesContext _context, Object _state)**
  Perform any processing required to restore the state from the entries in the state Object.

- **Object saveState(FacesContext _context)**
  Gets the state of the instance as a Serializable Object.

- **void setDir(String dir)**
  Set the value of the dir property.
<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setEscape</td>
<td>boolean escape</td>
<td>Set the value of the escape property.</td>
</tr>
<tr>
<td>setLang</td>
<td>String lang</td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td>setStyle</td>
<td>String style</td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>setStyleClass</td>
<td>String styleClass</td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td>setTitle</td>
<td>String title</td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.component.UIOutput:
- getConverter, getFamily, getLocalValue, getValue, setConverter, setValue

Methods inherited from class javax.faces.component.UIComponentBase:
- addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent:
- encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
**Field Detail**

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

*See Also:*
Constant Field Values

**Constructor Detail**

```java
public HtmlOutputFormat()
```

HtmlOutputFormat

```java
public HtmlOutputFormat()
```

**Method Detail**

```java
public String getDir()
```

```java
dir
```

"LTR"left-to-right "RTL"right-to-left

```java
getDir
```

```java
public String getDir()
```
Return the value of the `dir` property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

```java
public void setDir(String dir)
```

`dir`

```java
setDir
```

```java
public void setDir(String dir)
```

Set the value of the `dir` property.

```java
public boolean isEscape()
```

`escape`

```java
HTML XML "true"
```

```java
isEscape
```

```java
public boolean isEscape()
```

Return the value of the `escape` property.

Contents: Flag indicating that characters that are sensitive in HTML and XML markup must be escaped. This flag is set to "true" by default.
**public void setEscape(boolean escape)**

    escape

**setEscape**

```java
public void setEscape(boolean escape)
```

Set the value of the escape property.

**public String getLang()**

    lang

**getLang**

```java
public String getLang()
```

Return the value of the lang property.

Contents: Code describing the language used in the generated markup for this component.

**public void setLang(String lang)**

    lang
**setLang**

```java
public void setLang(String lang)
```

Set the value of the `lang` property.

---

**public String getStyle()**

```java
public String getStyle()
```

- **style**

CSS

**getStyle**

```java
public String getStyle()
```

Return the value of the `style` property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

```java
public void setStyle(String style)
```

**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.
public String getStyleClass()

    styleClass

    CSS "class"

getStyleClass

public String getStyleClass()

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

    styleClass

setStyleClass

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

public String getTitle()

    title
**getTitle**

public String getTitle()

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

**public void setTitle(String title)**

setTitle

title

**setTitle**

public void setTitle(String title)

Set the value of the title property.

**public Object saveState(FacesContext _context)**

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the
**StateHolder.saveState(javax.faces.context.FacesContext)** method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

**component** should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**
- saveState in interface **StateHolder**

**Overrides:**
- saveState in class **UIOutput**

```
public void restoreState(FacesContext _context, Object _state)
```

**restoreState**

```
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.
Specified by:

`restoreState` in interface `StateHolder`

Overrides:

`restoreState` in class `UIOutput`
javax.faces.component.html  **Class HtmlOutputLabel**

**java.lang.Object**
- javax.faces.component.UIComponent
  - javax.faces.component.UIComponentBase
  - javax.faces.component.UIOutput
    - javax.faces.component.html.HtmlOutputLabel

**All Implemented Interfaces:**
- StateHolder, ValueHolder

---

public class **HtmlOutputLabel**

extends **UIOutput**

**Extends:** **UIComponent** > **UIComponentBase** > **UIOutput**

**HTML label**

    rendererType "javax.faces.Label"    setRendererType()

Represents an HTML label element, used to define an accessible label for a corresponding input element.

By default, the rendererType property must be set to "javax.faces.Label". This value can be changed by calling the setRendererType() method.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
</table>

The standard component type for this component.

**Fields inherited from class javax.faces.component.UIOutput**

COMPONENT_FAMILY
### Fields inherited from class javax.faces.component.UIComponent

bindings

### Constructor Summary

**HtmlOutputLabel()**

### Method Summary

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getAccesskey()</td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td>String</td>
<td>getDir()</td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td>String</td>
<td>getFor()</td>
<td>Return the value of the for property.</td>
</tr>
<tr>
<td>String</td>
<td>getLang()</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnblur()</td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnclick()</td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td>String</td>
<td>getOndblclick()</td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnfocus()</td>
<td>Return the value of the onfocus property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeypress()</td>
<td>Return the value of the onkeypress property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeyup()</td>
<td>Return the value of the onkeyup property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousedown()</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>String getOnmousemove()</td>
<td>Return the value of the onmousemove property.</td>
<td></td>
</tr>
<tr>
<td>String getOnmouseout()</td>
<td>Return the value of the onmouseout property.</td>
<td></td>
</tr>
<tr>
<td>String getOnmouseover()</td>
<td>Return the value of the onmouseover property.</td>
<td></td>
</tr>
<tr>
<td>String getOnmouseup()</td>
<td>Return the value of the onmouseup property.</td>
<td></td>
</tr>
<tr>
<td>String getStyle()</td>
<td>Return the value of the style property.</td>
<td></td>
</tr>
<tr>
<td>String getStyleClass()</td>
<td>Return the value of the styleClass property.</td>
<td></td>
</tr>
<tr>
<td>String getTabIndex()</td>
<td>Return the value of the tabindex property.</td>
<td></td>
</tr>
<tr>
<td>String getTitle()</td>
<td>Return the value of the title property.</td>
<td></td>
</tr>
<tr>
<td>boolean isEscape()</td>
<td>Return the value of the escape property.</td>
<td></td>
</tr>
<tr>
<td>void restoreState(FacesContext _context, Object _state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
<td></td>
</tr>
<tr>
<td>Object saveState(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
<td></td>
</tr>
<tr>
<td>void setAccesskey(String accesskey)</td>
<td>Set the value of the accesskey property.</td>
<td></td>
</tr>
<tr>
<td>void setDir(String dir)</td>
<td>Set the value of the dir property.</td>
<td></td>
</tr>
<tr>
<td>void setEscape(boolean escape)</td>
<td>Set the value of the escape property.</td>
<td></td>
</tr>
<tr>
<td>void setFor(String _for)</td>
<td>Set the value of the for property.</td>
<td></td>
</tr>
<tr>
<td>void setLang(String lang)</td>
<td>Set the value of the lang property.</td>
<td></td>
</tr>
<tr>
<td>void setOnblur(String onblur)</td>
<td>Set the value of the onblur property.</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
<td></td>
</tr>
<tr>
<td><code>setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
<td></td>
</tr>
<tr>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
<td></td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
<td></td>
</tr>
<tr>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
<td></td>
</tr>
<tr>
<td><code>setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
<td></td>
</tr>
<tr>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UIOutput`:
- `getConverter`, `getFamily`, `getLocalValue`, `getValue`, `setConverter`,...
setValue

Methods inherited from class javax.faces.component.UICOMPONENTBASE
addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UICOMPONENT
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail
public HtmlOutputLabel()

HtmlOutputLabel

public HtmlOutputLabel()

### Method Detail

#### public String getAccesskey()

accesskey

---

**getAccesskey**

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

---

#### public void setAccesskey(String accesskey)

accesskey

---

**setAccesskey**

public void setAccesskey(String accesskey)
Set the value of the accesskey property.

```
public String getDir()
{
    dir
    "LTR"left-to-right "RTL"right-to-left
}
```

getDir

```
public String getDir()
{
    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
}
```

```
public void setDir(String dir)
{
    dir
}
```

setDir

```
public void setDir(String dir)
{
    Set the value of the dir property.
}
```

```
public boolean isEscape()
```
isEscape

public boolean isEscape()

  Return the value of the escape property.

  Contents: Flag indicating that characters that are sensitive in HTML and XML markup must be escaped. If omitted, this flag is assumed to be "true".

-----------------------------------------------

public void setEscape(boolean escape)

  escape

setEscape

public void setEscape(boolean escape)

  Set the value of the escape property.

-----------------------------------------------

public String getFor()

  for
getFor

```java
public String getFor()
```

Return the value of the `for` property.

Contents: Client identifier of the component for which this element is a label.

---

setFor

```java
public void setFor(String _for)
```

for

---

getLang

```java
public String getLang()
```

lang

---

getLang

```java
public String getLang()
```

Return the value of the `lang` property.

Contents: Code describing the language used in the generated
public void setLang(String lang)

    lang

setLang

public void setLang(String lang)

    Set the value of the lang property.

public String getOnblur()

    onblur

Javascript

getOnblur

public String getOnblur()

    Return the value of the onblur property.

    Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

    onblur
setOnblur

public void setOnblur(String onblur)

    Set the value of the onblur property.

public String getOnclick()

    onclick

Javascript

getOnclick

public String getOnclick()

    Return the value of the onclick property.

    Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

    onclick

setOnclick

public void setOnclick(String onclick)

    Set the value of the onclick property.
public String getOndblclick()

    ondblclick

Javascript

getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.

    Contents: Javascript code executed when a pointer button is double clicked over this element.

---------------------------------------------

public void setOndblclick(String ondblclick)

    ondblclick

setOndblclick

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

---------------------------------------------

public String getOnfocus()

    onfocus

Javascript
getOnfocus

```java
public String getOnfocus()
```

Return the value of the `onfocus` property.

Contents: Javascript code executed when this element receives focus.

---

public void setOnfocus(String onFocus)

```java
onfocus
```

---

setOnfocus

```java
public void setOnfocus(String onFocus)
```

Set the value of the `onfocus` property.

---

public String getOnkeydown()

```java
onkeydown
```

Javascript

---

getOnkeydown

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.
Contents: Javascript code executed when a key is pressed down over this element.

```java
public void setOnkeydown(String onkeydown)

  onkeydown

setOnkeydown

public void setOnkeydown(String onkeydown)

  Set the value of the onkeydown property.

```}

```java
public String getOnkeypress()

  onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

  Return the value of the onkeypress property.

  Contents: Javascript code executed when a key is pressed and released over this element.

```}

```java
public void setOnkeypress(String onkeypress)

  onkeypress

```
**setOnkeypress**

public void **setOnkeypress**(*String* onkeypress)

Set the value of the onkeypress property.

---

**public String getOnkeyup()**

    onkeyup

**Javascript**

**getOnkeyup**

public **String** **getOnkeyup**()

    Return the value of the onkeyup property.
    
    Contents: Javascript code executed when a key is released over this element.

---

**public void setOnkeyup(String onkeyup)**

    onkeyup

**setOnkeyup**

public void **setOnkeyup**(*String* onkeyup)

    Set the value of the onkeyup property.
public String getOnmousedown()

    onmousedown

Javascript

getOnmousedown

public String getOnmousedown()

    Return the value of the onmousedown property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.

public void setOnmousedown(String onmousedown)

    onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

public String getOnmousemove()

    onmousemove

Javascript
**getOnmousemove**

```java
public String getOnmousemove()
```

Return the value of the `onmousemove` property.

Contents: Javascript code executed when a pointer button is moved within this element.

**setOnmousemove(String onmousemove)**

```java
public void setOnmousemove(String onmousemove)
```

Set the value of the `onmousemove` property.

**getOnmouseout()**

```java
public String getOnmouseout()
```

Return the value of the `onmouseout` property.
Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

  onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

  Set the value of the onmouseout property.

public String getOnmouseover()

  onmouseover

Javascript

getonmouseover

public String getOnmouseover()

  Return the value of the onmouseover property.
  Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

  onmouseover
**setOnmouseover**

**public void setOnmouseover(String onmouseover)**

Set the value of the `onmouseover` property.

---

**public String getOnmouseup()**

`onmouseup`

**Javascript**

---

**getOnmouseup**

**public String getOnmouseup()**

Return the value of the `onmouseup` property.

Contents: Javascript code executed when a pointer button is released over this element.

---

**public void setOnmouseup(String onmouseup)**

`onmouseup`

---

**setOnmouseup**

**public void setOnmouseup(String onmouseup)**

Set the value of the `onmouseup` property.
public String getStyle()

    style

CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style

setStyle

public void setStyle(String style)

    Set the value of the style property.

public String getStyleClass()

    styleClass

CSS "class"
**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**public void setStyleClass(String styleClass)**

```
styleClass
```

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.

---

**public String getTabindex()**

```
tabindex
```

Tab  0  32767

**getTabindex**

```java
public String getTabindex()
```
Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

```
public void setTabindex(String tabindex)
```

`tabindex`

```
setTabindex
```

```
public void setTabindex(String tabindex)
```

Set the value of the `tabindex` property.

```
public String getTitle()
```

`title`

```
getTitle
```

```
public String getTitle()
```

Return the value of the `title` property.

Contents: Advisory title information about markup elements generated for this component.

```
public void setTitle(String title)
```
setTitle

public void setTitle(String title)

Set the value of the title property.

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
restoreState in interface StateHolder

Overrides:
restoreState in class UIOutput

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.faces.component.html  Class HtmlOutputLink

java.lang.Object
  ↓ javax.faces.component.UIComponent
     ↓ javax.faces.component.UIComponentBase
        ↓ javax.faces.component.UIOutput
           ↓ javax.faces.component.html.HtmlOutputLink

All Implemented Interfaces:
  StateHolder, ValueHolder

```
public class HtmlOutputLink
extends UIOutput

Extends: UIComponent > UIComponentBase > UIOutput

HTML  a  value URL

    rendererType "javax.faces.Link"  setRendererType()
```

Represents an HTML a (hyperlink) element that may be used to link to an arbitrary URL defined by the value property.

By default, the rendererType property must be set to "javax.faces.Link". This value can be changed by calling the setRendererType() method.

---

**Field Summary**

```
static String COMPONENT_TYPE
  The standard component type for this component.

Fields inherited from class javax.faces.component.UIOutput
COMPONENT_FAMILY
```
### Fields inherited from class `javax.faces.component.UIComponent bindings`

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HtmlOutputLink()</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getAccesskey()</td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td>String getCharset()</td>
<td>Return the value of the charset property.</td>
</tr>
<tr>
<td>String getCoords()</td>
<td>Return the value of the coords property.</td>
</tr>
<tr>
<td>String getDir()</td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td>String getHreflang()</td>
<td>Return the value of the hreflang property.</td>
</tr>
<tr>
<td>String getLang()</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String getOnblur()</td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>String getOnclick()</td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td>String getOndblclick()</td>
<td>Return the value of the ondblclick property.</td>
</tr>
<tr>
<td>String getOnfocus()</td>
<td>Return the value of the onfocus property.</td>
</tr>
<tr>
<td>String getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String getOnkeypress()</td>
<td>Return the value of the onkeypress property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>getRel()</code></td>
<td>Return the value of the <code>rel</code> property.</td>
</tr>
<tr>
<td><code>getRev()</code></td>
<td>Return the value of the <code>rev</code> property.</td>
</tr>
<tr>
<td><code>getShape()</code></td>
<td>Return the value of the <code>shape</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>getTarget()</code></td>
<td>Return the value of the <code>target</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Return the value of the <code>type</code> property.</td>
</tr>
<tr>
<td><code>isDisabled()</code></td>
<td>Return the value of the <code>disabled</code> property.</td>
</tr>
<tr>
<td><code>restoreState(FacesContext context, Object state)</code></td>
<td>Perform any processing required to restore the state from</td>
</tr>
</tbody>
</table>
the entries in the state Object.

**Object**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>setAccesskey(String accesskey)</code></td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td><code>setCharset(String charset)</code></td>
<td>Set the value of the charset property.</td>
</tr>
<tr>
<td><code>setCoords(String coords)</code></td>
<td>Set the value of the coords property.</td>
</tr>
<tr>
<td><code>setDir(String dir)</code></td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td><code>setDisabled(boolean disabled)</code></td>
<td>Set the value of the disabled property.</td>
</tr>
<tr>
<td><code>setHreflang(String hreflang)</code></td>
<td>Set the value of the hreflang property.</td>
</tr>
<tr>
<td><code>setLang(String lang)</code></td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td><code>setOnblur(String onblur)</code></td>
<td>Set the value of the onblur property.</td>
</tr>
<tr>
<td><code>setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td><code>setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>setRel(String rel)</code></td>
<td>Set the value of the <code>rel</code> property.</td>
</tr>
<tr>
<td><code>setRev(String rev)</code></td>
<td>Set the value of the <code>rev</code> property.</td>
</tr>
<tr>
<td><code>setShape(String shape)</code></td>
<td>Set the value of the <code>shape</code> property.</td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>setTabIndex(String tabindex)</code></td>
<td>Set the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>setTarget(String target)</code></td>
<td>Set the value of the <code>target</code> property.</td>
</tr>
<tr>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>setType(String type)</code></td>
<td>Set the value of the <code>type</code> property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UIOutput`:
- `getConverter`, `getFamily`, `getLocaleValue`, `getValue`, `setConverter`, `setValue`

Methods inherited from class `javax.faces.component.UIComponentBase`:
- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, ...
getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail

public HtmlOutputLink()
public HtmlOutputLink()

Method Detail

public String getAccesskey()

accesskey

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

public void setAccesskey(String accesskey)

accesskey

setAccesskey

public void setAccesskey(String accesskey)

Set the value of the accesskey property.

public String getCharset()
getCharset

public String getCharset()

Return the value of the charset property.

Contents: The character encoding of the resource designated by this hyperlink.

public void setCharset(String charset)

ccharset

setCharset

public void setCharset(String charset)

Set the value of the charset property.

public String getCoords()

cords
getCoords

public String getCoords()

    Return the value of the coords property.

    Contents: The position and shape of the hot spot on the screen (for use in client-side image maps).

public void setCoords(String coords)

    coords

setCoords

public void setCoords(String coords)

    Set the value of the coords property.

public String getDir()

    dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

public void setDir(String dir)

dir

setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isDisabled()

disabled

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit.

public void setDisabled(boolean disabled)

disabled
**setDisabled**

public void setDisabled(boolean disabled)

Set the value of the disabled property.

---

**public String getHreflang()**

hreflang

**getHreflang**

public String getHreflang()

Return the value of the hreflang property.

Contents: The language code of the resource designated by this hyperlink.

---

**public void setHreflang(String hreflang)**

hreflang

**setHreflang**

public void setHreflang(String hreflang)

Set the value of the hreflang property.
public String getLang()

  lang

getLang

public String getLang()

  Return the value of the lang property.
  Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

  lang

setLang

public void setLang(String lang)

  Set the value of the lang property.

public String getOnblur()

  onblur

Javascript
getOnblur

public String getOnblur()

    Return the value of the onblur property.

    Contents: Javascript code executed when this element loses focus.

------------------------------------------

public void setOnblur(String onblur)

    onblur

setOnblur

public void setOnblur(String onblur)

    Set the value of the onblur property.

------------------------------------------

public String getOnclick()

    onclick

Javascript

getOnclick

public String getOnclick()

    Return the value of the onclick property.
Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnonclick(String onclick)

  onclick

setOnonclick

public void setOnonclick(String onclick)

  Set the value of the onclick property.

public String getOndblclick()

  ondblclick

Javascript

getOndblclick

public String getOndblclick()

  Return the value of the ondblclick property.

  Contents: Javascript code executed when a pointer button is double clicked over this element.

public void setOndblclick(String ondblclick)

  ondblclick
**setOndblclick**

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

---

**public String getOnfocus()**

    onFocus

*Javascript*

**getOnfocus**

public String getOnfocus()

    Return the value of the onFocus property.

    Contents: Javascript code executed when this element receives focus.

---

**public void setOnfocus(String onFocus)**

    onFocus

**setOnfocus**

public void setOnfocus(String onFocus)

    Set the value of the onFocus property.
public String getOnkeydown()

    onkeydown

Javascript

getonkeydown

public String getOnkeydown()

    Return the value of the onkeydown property.

    Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

    onkeydown

setonkeydown

public void setOnkeydown(String onkeydown)

    Set the value of the onkeydown property.

public String getOnkeypress()

    onkeypress

Javascript
**getOnkeypress**

```java
public String getOnkeypress()
```

Return the value of the onkeypress property.

Contents: Javascript code executed when a key is pressed and released over this element.

---

**public void setOnkeypress(String onkeypress)**

```java
onkeypress
```

---

**setOnkeypress**

```java
public void setOnkeypress(String onkeypress)
```

Set the value of the onkeypress property.

---

**public String getOnkeyup()**

```java
onkeyup
```

**Javascript**

---

**getOnkeyup**

```java
public String getOnkeyup()
```

Return the value of the onkeyup property.
Contents: Javascript code executed when a key is released over this element.

---

**public void setOnkeyup(String onkeyup)**

`onkeyup`

---

**setOnkeyup**

**public void setOnkeyup(String onkeyup)**

Set the value of the `onkeyup` property.

---

**public String getOnmousedown()**

`onmousedown`

**Javascript**

---

**getOnmousedown**

**public String getOnmousedown()**

Return the value of the `onmousedown` property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

---

**public void setOnmousedown(String onmousedown)**

`onmousedown`
**setOnmousedown**

public void setOnmousedown(String onmousedown)

Set the value of the onmousedown property.

---

**public String getOnmousemove()**

onmousemove

Javascript

**getOnmousemove**

public String getOnmousemove()

Return the value of the onmousemove property.

Contents: Javascript code executed when a pointer button is moved within this element.

---

**public void setOnmousemove(String onmousemove)**

onmousemove

**setOnmousemove**

public void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.
public String getOnmouseout()

  onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

  Return the value of the onmouseout property.

  Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseover)

  onmouseover

setOnmouseout

public void setOnmouseout(String onmouseover)

  Set the value of the onmouseover property.

public String getOnmouseover()

  onmouseover

Javascript
getOnmouseover

class public String getOnmouseover()

Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

onmouseover

setOnmouseover

class public void setOnmouseover(String onmouseover)

Set the value of the onmouseover property.

public String getOnmouseup()

onmouseup

Javascript

getOnmouseup

class public String getOnmouseup()

Return the value of the onmouseup property.
Contents: Javascript code executed when a pointer button is released over this element.

```java
public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getRel()

    rel

getRel

public String getRel()

    Return the value of the rel property.

    Contents: The relationship from the current document to the anchor specified by this hyperlink. The value of this attribute is a space-separated list of link types.

public void setRel(String rel)
```
rel

setRel

public void setRel(String rel)

    Set the value of the rel property.

-------------------------------

public String getRev()

    rev

getRev

public String getRev()

    Return the value of the rev property.

    Contents: A reverse link from the anchor specified by this hyperlink to the current document. The value of this attribute is a space-separated list of link types.

-------------------------------

public void setRev(String rev)

    rev

setRev
public void setRev(String rev)

    Set the value of the rev property.

public String getShape()

    shape
default rect circle
poly

getShape

public String getShape()

    Return the value of the shape property.

    Contents: The shape of the hot spot on the screen (for use in client-side image maps). Valid values are: default (entire region); rect (rectangular region); circle (circular region); and poly (polygonal region).

public void setShape(String shape)

    shape

setShape

public void setShape(String shape)

    Set the value of the shape property.
public String getStyle()

  style

  CSS

getStyle

public String getStyle()

  Return the value of the style property.
  Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

  style

setStyle

public void setStyle(String style)

  Set the value of the style property.

public String getStyleClass()

  styleClass

CSS "class"
**getStyleClass**

public String getStyleClass()

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

**public void setStyleClass(String styleClass)**

    styleClass

**setStyleClass**

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

**public String getTabindex()**

    tabindex

    **Tab 0 32767**

**getTabindex**

public String getTabindex()
Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

```java
public void setTabindex(String tabindex)
```

Set the value of the `tabindex` property.

```java
public String getTarget()
```

Return the value of the `target` property.

Contents: Name of a frame where the resource retrieved via this hyperlink is to be displayed.

```java
public void setTarget(String target)
```
**setTarget**

```java
public void setTarget(String target)
```

Set the value of the `target` property.

---

**getTitle**

```java
public String getTitle()
```

Return the value of the `title` property.

Contents: Advisory title information about markup elements generated for this component.

---

**setTitle**

```java
public void setTitle(String title)
```

---
public String getType()

getType

public String getType()

    Return the value of the type property.
    
    Contents: The content type of the resource designated by this hyperlink.

public void setType(String type)

setType

public void setType(String type)

    Set the value of the type property.

public Object saveState(FacesContext _context)

saveState
public Object saveState(FacesContext _context)

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

*component* should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**

saveState in interface StateHolder

**Overrides:**

saveState in class UIOutput

---

public void restoreState(FacesContext _context, Object _state)

**restoreState**

public void restoreState(FacesContext _context, Object _state)

**Description copied from interface: StateHolder**
Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: restoreState in interface StateHolder
Overrides: restoreState in class UIOutput
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
javax.faces.component.html  Class HtmlOutputText

java.lang.Object
  ▼ javax.faces.component.UIComponent
     ▼ javax.faces.component.UIComponentBase
        ▼ javax.faces.component.UIOutput
           ▼ javax.faces.component.html.HtmlOutputText

All Implemented Interfaces:
  StateHolder, ValueHolder

public class HtmlOutputText
extends UIOutput

Extends: UIComponent > UIComponentBase > UIOutput

I18N CSS  span

  rendererType "javax.faces.Text"  setRendererType()

Renders the component value as text, optionally wrapping in a span element if I18N attributes, CSS styles or style classes are specified.

By default, the rendererType property must be set to "javax.faces.Text". This value can be changed by calling the setRendererType() method.

---

Field Summary

static String COMPONENT_TYPE
  The standard component type for this component.

Fields inherited from class javax.faces.component.UIOutput
COMPONENT_FAMILY
### Constructor Summary

**HtmlOutputText()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDir()</code></td>
<td>Return the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>Return the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>isEscape()</code></td>
<td>Return the value of the <code>escape</code> property.</td>
</tr>
<tr>
<td><code>restoreState()</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>saveState()</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>setDir(String dir)</code></td>
<td>Set the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>setEscape(boolean escape)</code></td>
<td>Set the value of the <code>escape</code> property.</td>
</tr>
<tr>
<td><code>setLang(String lang)</code></td>
<td>Set the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the <code>style</code> property.</td>
</tr>
</tbody>
</table>
Set the value of the style property.

<table>
<thead>
<tr>
<th>void setStyleClass(String styleClass)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the styleClass property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void setTitle(String title)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.faces.component.UIOutput

getConverter, getFamily, getLocalValue, getValue, setConverter, setValue

### Methods inherited from class javax.faces.component.UIComponentBase

addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getFace, getRender, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

### Methods inherited from class javax.faces.component.UIComponent

encodeAll, getContainerClientId, getValueExpression, setValueExpression

### Methods inherited from class java.lang.Object

cache, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

COMPONENT_TYPE
public static final String COMPONENT_TYPE

    The standard component type for this component.

    See Also:
    Constant Field Values

Constructor Detail

class HtmlOutputText()

HtmlOutputText

class HtmlOutputText()

Method Detail

class public String getDir()

    dir

    "LTR" left-to-right "RTL" right-to-left

getDir

class public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
public void setDir(String dir)

dir

setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isEscape()

escape

HTML  XML  "true"

isEscape

public boolean isEscape()

Return the value of the escape property.

Contents: Flag indicating that characters that are sensitive in HTML and XML markup must be escaped. This flag is set to "true" by default.

public void setEscape(boolean escape)

escape
setEscape

public void setEscape(boolean escape)

  Set the value of the escape property.

public String getLang()

  lang

getLang

public String getLang()

  Return the value of the lang property.

  Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

  lang

setLang

public void setLang(String lang)

  Set the value of the lang property.
public String getStyle()

    style

CSS

getStyle

public String getStyle()

    Return the value of the style property.
    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style

setStyle

public void setStyle(String style)

    Set the value of the style property.

public String getStyleClass()

    styleClass

CSS "class"
**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**public void setStyleClass(String styleClass)**

```java
styleClass
```

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.

---

**public String getTitle()**

```java
title
```

---

**getTitle**

```java
public String getTitle()
```
Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

```java
public void setTitle(String title)
```

setTitle

```java
public void setTitle(String title)
```

Set the value of the title property.

```java
public Object saveState(FacesContext _context)
```

saveState

```java
public Object saveState(FacesContext _context)
```

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:
Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
   saveState in interface StateHolder
Overrides:
   saveState in class UIOutput

public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UICOMPONENT with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
   restoreState in interface StateHolder
Overrides:
   restoreState in class UIOutput
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html  Class HtmlPanelGrid

java.lang.Object
   ↓ javax.faces.component.UIComponent
      ↓ javax.faces.component.UIComponentBase
         ↓ javax.faces.component.UIPanel
             ↓ javax.faces.component.html.HtmlPanelGrid

All Implemented Interfaces:
   StateHolder

public class HtmlPanelGrid
    extends UIPanel

Extends: UIComponent >UIComponentBase > UIPanel

   rendererType "javax.faces.Grid"     setRendererType()

Renders child components in a table, starting a new row after the specified number of columns.

By default, the rendererType property must be set to "javax.faces.Grid". This value can be changed by calling the setRendererType() method.

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIPanel

COMPONENT_FAMILY
### Fields inherited from class javax.faces.component.UIComponent bindings

### Constructor Summary

**HtmlPanelGrid()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String getBgcolor()</strong></td>
<td>Return the value of the bgcolor property.</td>
</tr>
<tr>
<td><strong>int getBorder()</strong></td>
<td>Return the value of the border property.</td>
</tr>
<tr>
<td><strong>String getCaptionClass()</strong></td>
<td>Return the value of the captionClass property.</td>
</tr>
<tr>
<td><strong>String getCaptionStyle()</strong></td>
<td>Return the value of the captionStyle property.</td>
</tr>
<tr>
<td><strong>String getCellpaddings()</strong></td>
<td>Return the value of the cellpaddings property.</td>
</tr>
<tr>
<td><strong>String getCellspacing()</strong></td>
<td>Return the value of the cellspacings property.</td>
</tr>
<tr>
<td><strong>String getColumnClasses()</strong></td>
<td>Return the value of the columnClasses property.</td>
</tr>
<tr>
<td><strong>int getColumns()</strong></td>
<td>Return the value of the columns property.</td>
</tr>
<tr>
<td><strong>String getDir()</strong></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td><strong>String getFooterClass()</strong></td>
<td>Return the value of the footerClass property.</td>
</tr>
<tr>
<td><strong>String getFrame()</strong></td>
<td>Return the value of the frame property.</td>
</tr>
<tr>
<td><strong>String getHeaderClass()</strong></td>
<td>Return the value of the headerClass property.</td>
</tr>
</tbody>
</table>
String `getLang()`
    Return the value of the lang property.

String `getOnClick()`
    Return the value of the onclick property.

String `getOnDblick()`
    Return the value of the ondblclick property.

String `getOnKeyDown()`
    Return the value of the onkeydown property.

String `getOnKeyPress()`
    Return the value of the onkeypress property.

String `getOnKeyUp()`
    Return the value of the onkeyup property.

String `getOnMouseDown()`
    Return the value of the onmousedown property.

String `getOnMouseMove()`
    Return the value of the onmousemove property.

String `getOnMouseOut()`
    Return the value of the onmouseout property.

String `getOnMouseOver()`
    Return the value of the onmouseover property.

String `getOnMouseUp()`
    Return the value of the onmouseup property.

String `getRowClasses()`
    Return the value of the rowClasses property.

String `getRules()`
    Return the value of the rules property.

String `getStyle()`
    Return the value of the style property.

String `getStyleClass()`
    Return the value of the styleClass property.

String `getSummary()`
    Return the value of the summary property.

String `getTitle()`
    Return the value of the title property.
<table>
<thead>
<tr>
<th><strong>String</strong> <code>getWidth()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the value of the <code>width</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>restoreState(FacesContext _context, Object _state)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Object</strong> <code>saveState(FacesContext _context)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setBackground(String bgcolor)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>bgcolor</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setBorder(int border)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>border</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setCaptionClass(String captionClass)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>captionClass</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setCaptionStyle(String captionStyle)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>captionStyle</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setCellpadding(String cellpadding)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>cellpadding</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setCellspacing(String cellspacing)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>cellspacing</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setColumnClasses(String columnClasses)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>columnClasses</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setColumns(int columns)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>columns</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setDir(String dir)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>dir</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setFooterClass(String footerClass)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>footerClass</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setFrame(String frame)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>frame</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setHeaderClass(String headerClass)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>headerClass</code> property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>void</strong> <code>setLang(String lang)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Set the value of the <code>lang</code> property.</td>
</tr>
</tbody>
</table>

| **void** `setOnclick(String onclick)` |
Set the value of the onclick property.

```java
void setOndblclick(String ondblclick)
    Set the value of the ondblclick property.
```

Set the value of the onkeydown property.

```java
void setOnkeydown(String onkeydown)
    Set the value of the onkeydown property.
```

Set the value of the onkeypress property.

```java
void setOnkeypress(String onkeypress)
    Set the value of the onkeypress property.
```

Set the value of the onkeyup property.

```java
void setOnkeyup(String onkeyup)
    Set the value of the onkeyup property.
```

Set the value of the onmousedown property.

```java
void setOnmousedown(String onmousedown)
    Set the value of the onmousedown property.
```

Set the value of the onmousemove property.

```java
void setOnmousemove(String onmousemove)
    Set the value of the onmousemove property.
```

Set the value of the onmouseout property.

```java
void setOnmouseout(String onmouseout)
    Set the value of the onmouseout property.
```

Set the value of the onmouseover property.

```java
void setOnmouseover(String onmouseover)
    Set the value of the onmouseover property.
```

Set the value of the onmouseup property.

```java
void setOnmouseup(String onmouseup)
    Set the value of the onmouseup property.
```

Set the value of the rowClasses property.

```java
void setRowClasses(String rowClasses)
    Set the value of the rowClasses property.
```

Set the value of the rules property.

```java
void setRules(String rules)
    Set the value of the rules property.
```

Set the value of the style property.

```java
void setStyle(String style)
    Set the value of the style property.
```

Set the value of the styleClass property.

```java
void setStyleClass(String styleClass)
    Set the value of the styleClass property.
```

Set the value of the summary property.

```java
void setSummary(String summary)
    Set the value of the summary property.
```

Set the value of the title property.

```java
void setTitle(String title)
    Set the value of the title property.
```

Set the value of the width property.

```java
void setWidth(String width)
    Set the value of the width property.
```
### Methods inherited from class `javax.faces.component.UIPanel`

- `getFamily`

### Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`

### Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

- [Constant Field Values](#)

---
Constructor Detail

public HtmlPanelGrid()

HtmlPanelGrid

public HtmlPanelGrid()

Method Detail

public String getBgcolor()

bgcolor

getBgcolor

public String getBgcolor()

Return the value of the bgcolor property.

Contents: Name or code of the background color for this table.

public void setBgcolor(String bgcolor)

bgcolor

setBgcolor
public void setBackground(String bgcolor)

    Set the value of the bgcolor property.

public int getBorder()

    border

getBorder

public int getBorder()

    Return the value of the border property.

    Contents: Width (in pixels) of the border to be drawn around this table.

public void setBorder(int border)

    border

setBorder

public void setBorder(int border)

    Set the value of the border property.

public String getCaptionClass()
getCaptionClass

public String getCaptionClass()

Return the value of the captionClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any caption generated for this table.

setCaptionClass

public void setCaptionClass(String captionClass)

captionClass

setCaptionClass

public void setCaptionClass(String captionClass)

Set the value of the captionClass property.

public String getCaptionStyle()

captionStyle

CSS
**getCaptionStyle**

```java
public String getCaptionStyle()
```

Return the value of the `captionStyle` property.

Contents: CSS style(s) to be applied when this caption is rendered.

**public void setCaptionStyle(String captionStyle)**

```java
captionStyle
```

**setCaptionStyle**

```java
public void setCaptionStyle(String captionStyle)
```

Set the value of the `captionStyle` property.

**public String getCellpadding()**

```java
cellpadding
```

**getCellpadding**

```java
public String getCellpadding()
```

Return the value of the `cellpadding` property.

Contents: Definition of how much space the user agent should leave between the border of each cell and its contents.
public void setCellpadding(String cellpadding)

cellpadding

setCellpadding

public void setCellpadding(String cellpadding)

Set the value of the cellpadding property.

public String getCellspacing()

cellspacing

gCellSpacing

public String getCellspacing()

Return the value of the cellspacing property.

Contents: Definition of how much space the user agent should leave between the left side of the table and the leftmost column, the top of the table and the top of the top side of the topmost row, and so on for the right and bottom of the table. It also specifies the amount of space to leave between cells.

public void setCellspacing(String cellspacing)

cellspacing
**setCellspacing**

```java
public void setCellspacing(String cellspacing)

Set the value of the cellspacing property.
```

---

**public String getColumnClasses()**

```java
getColumnClasses

CSS "columns" "class"
"columns" "columns"
```

---

**getColumnClasses**

```java
public String getColumnClasses()

Return the value of the columnClasses property.

Contents: Comma-delimited list of CSS style classes that will be applied to the columns of this table. A space separated list of classes may also be specified for any individual column. If the number of elements in this list is less than the number of columns specified in the "columns" attribute, no "class" attribute is output for each column greater than the number of elements in the list. If the number of elements in the list is greater than the number of columns specified in the "columns" attribute, the elements at the position in the list after the value of the "columns" attribute are ignored.
```

---

**public void setColumnClasses(String columnClasses)**

```java
getColumnClasses
```

```java
columnClasses
```
**setColumnClasses**

```java
class {
    public void setColumnClasses(String columnClasses) {
        Set the value of the columnClasses property.
    }
}
```

**getColumnCount**

```java
class {
    public int getColumnCount() {
        return columns;
    }
}
```

**getColumns**

```java
class {
    public int getColumns() {
        return columns;
    }
}
```

Contents: The number of columns to render before starting a new row.

**setColumns**

```java
class {
    public void setColumns(int columns) {
        columns = columns;
    }
}
```

**setColumns**

```java
class {
    public void setColumns(int columns) {
        Set the value of the columns property.
    }
}
```
public String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

Return the value of the dir property.

Contents: Direction indication for text that does not inherit
directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-
left).

public void setDir(String dir)

dir

setDir

public void setDir(String dir)

Set the value of the dir property.

public String getFooterClass()

footerClass
CSS

**getFooterClass**

```java
public String getFooterClass()
```

Return the value of the footerClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any footer generated for this table.

---

**public void setFooterClass(String footerClass)**

```
footerClass
```

---

**setFooterClass**

```java
public void setFooterClass(String footerClass)
```

Set the value of the footerClass property.

---

**public String getFrame()**

```
frame
```

`noneabovelowersidesvsideslhrsrsboxborder`
getFrame

public String getFrame()

Return the value of the frame property.

Contents: Code specifying which sides of the frame surrounding this table will be visible. Valid values are: none (no sides, default value); above (top side only); below (bottom side only); hsides (top and bottom sides only); vsides (right and left sides only); lhs (left hand side only); rhs (right hand side only); box (all four sides); and border (all four sides).

public void setFrame(String frame)

frame

setFrame

public void setFrame(String frame)

Set the value of the frame property.

public String getHeaderClass()

headerClass

CSS

getHeaderClass

public String getHeaderClass()
Return the value of the headerClass property.

Contents: Space-separated list of CSS style class(es) that will be applied to any header generated for this table.

```java
public void setHeaderClass(String headerClass)

headerClass
```

**setHeaderClass**

```java
public void setHeaderClass(String headerClass)

Set the value of the headerClass property.
```

**getLang**

```java
public String getLang()

  lang
```

**getLang**

```java
public String getLang()

  Return the value of the lang property.

  Contents: Code describing the language used in the generated markup for this component.
```

```java
public void setLang(String lang)
```
**lang**

**setLang**

```java
public void setLang(String lang)
```

Set the value of the `lang` property.

**public String getOnclick()**

**onclick**

**Javascript**

**getOnclick**

```java
public String getOnclick()
```

Return the value of the `onclick` property.

Contents: Javascript code executed when a pointer button is clicked over this element.

**public void setOnclick(String onclick)**

**onclick**

**setOnclick**

```java
public void setOnclick(String onclick)
```
Set the value of the `onclick` property.

```java
public String getOndblclick()

  ondblclick

Javascript

getOndblclick

public String getOndblclick()

  Return the value of the `ondblclick` property.

  Contents: Javascript code executed when a pointer button is double clicked over this element.

public void setOndblclick(String ondblclick)

  ondblclick

setOndblclick

public void setOndblclick(String ondblclick)

  Set the value of the `ondblclick` property.

public String getOnkeydown()

  onkeydown
getOnkeydown

public String getOnkeydown()

Return the value of the onkeydown property.

Contents: Javascript code executed when a key is pressed down over this element.

setOnkeydown

public void setOnkeydown(String onkeydown)

onkeydown

getOnkeypress

public String getOnkeypress()

onkeypress

getOnkeypress

public String getOnkeypress()
Return the value of the `onkeypress` property.

Contents: Javascript code executed when a key is pressed and released over this element.

```
public void setOnkeypress(String onkeypress)
```

```
setOnkeypress
```

```
public void setOnkeypress(String onkeypress)
```

Set the value of the `onkeypress` property.

```
public String getOnkeyup()
```

```
getOnkeyup
```

```
public String getOnkeyup()
```

Return the value of the `onkeyup` property.

Contents: Javascript code executed when a key is released over this element.

```
public void setOnkeyup(String onkeyup)
```
onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

Set the value of the onkeyup property.

getOnmousedown

public String getOnmousedown()

onmousedown

Javascript

getOnmousedown

public String getOnmousedown()

Return the value of the onmousedown property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

public void setOnmousedown(String onmousedown)

onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)
public String getOnmousemove()

  onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

  Return the value of the onmousemove property.

  Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

  onmousemove

setOnmousemove

public void setOnmousemove(String onmousemove)

  Set the value of the onmousemove property.

public String getOnmouseout()

  onmouseout
Javascript

getOnmouseout

public String getOnmouseout()

    Return the value of the onmouseout property.

    Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

    onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

    Set the value of the onmouseout property.

public String getOnmouseover()

    onmouseover

Javascript

getOnmouseover

public String getOnmouseover()
Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.

```
public void setOnmouseover(String onmouseover)
```

```
setOnmouseover
```

```
public void setOnmouseover(String onmouseover)
```

Set the value of the onmouseover property.

```
public String getOnmouseup()
```

```
getOnmouseup
```

```
public String getOnmouseup()
```

Return the value of the onmouseup property.

Contents: Javascript code executed when a pointer button is released over this element.

```
public void setOnmouseup(String onmouseup)
```
onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getRowClasses()

    rowClasses

CSS

getRowClasses

public String getRowClasses()

    Return the value of the rowClasses property.

    Contents: Comma-delimited list of CSS style classes that will be applied to the rows of this table. A space separated list of classes may also be specified for any individual row. Those styles are applied, in turn, to each row in the table. For example, if the list has two elements, the first style class in the list is applied to the first row, the second to the second row, the first to the third row, the second to the fourth row, etc. In other words, we keep iterating through the list until we reach the end, and then we start at the beginning again.

public void setRowClasses(String rowClasses)

    rowClasses
**setRowClasses**

```
public void setRowClasses(String rowClasses)
```

Set the value of the rowClasses property.

---

**public String getRules()**

```
    rules
```

**nongroups**

**rows**

**cols**

**all**

---

**getRules**

```
public String getRules()
```

Return the value of the rules property.

Contents: Code specifying which rules will appear between cells within this table. Valid values are: none (no rules, default value); groups (between row groups); rows (between rows only); cols (between columns only); and all (between all rows and columns).

---

**public void setRules(String rules)**

```
   rules
```

**setRules**
public void setRules(String rules)

    Set the value of the rules property.

public String getStyle()

    style

    CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style

setStyle

public void setStyle(String style)

    Set the value of the style property.

public String getStyleClass()
getStyleClass

public String getStyleClass()

Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

-------------------------------

public void setStyleClass(String styleClass)

styleClass

setStyleClass

public void setStyleClass(String styleClass)

Set the value of the styleClass property.

-------------------------------

public String getSummary()

summary

Braille
**getSummary**

```java
public String getSummary()
```

Return the value of the summary property.

Contents: Summary of this table's purpose and structure, for user agents rendering to non-visual media such as speech and Braille.

---

**public void setSummary(String summary)**

summary

---

**setSummary**

```java
public void setSummary(String summary)
```

Set the value of the summary property.

---

**public String getTitle()**

title

---

**getTitle**

```java
public String getTitle()
```

Return the value of the title property.

Contents: Advisory title information about markup elements
generated for this component.

public void setTitle(String title)

setTitle

public void setTitle(String title)

Set the value of the title property.

public String getWidth()

getWidth

public String getWidth()

Return the value of the width property.

Contents: Width of the entire table, for visual user agents.

public void setWidth(String width)

setWidth

public void setWidth(String width)
**setWidth**

```java
public void setWidth(String width)
```

Set the value of the `width` property.

**saveState**

```java
public Object saveState(FacesContext _context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a `Serializable` Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a `UIComponent` with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**
`saveState` in interface `StateHolder`

**Overrides:**
`saveState` in class `UIComponentBase`
public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: restoreState in interface StateHolder

Overrides: restoreState in class UIComponentBase
javax.faces.component.html  Class HtmlPanelGroup

java.lang.Object
   \ javax.faces.component.UIComponent
      \ javax.faces.component.UIComponentBase
         \ javax.faces.component.UIPanel
            \ javax.faces.component.html.HtmlPanelGroup

All Implemented Interfaces:
   StateHolder

public class HtmlPanelGroup
   extends JPanel

Extends: UIComponent > UIComponentBase > JPanel

   rendererType "javax.faces.Group"    setRendererType()

Causes all child components of this component to be rendered. This is useful in scenarios where a parent component is expecting a single component to be present, but the application wishes to render more than one.

By default, the rendererType property must be set to "javax.faces.Group". This value can be changed by calling the setRendererType() method.

---

Field Summary

| static String COMPONENT_TYPE |
| The standard component type for this component. |
### Fields inherited from class javax.faces.component.UIPanel

**COMPONENT_FAMILY**

### Fields inherited from class javax.faces.component.UIComponent

**bindings**

### Constructor Summary

**HtmlPanelGroup()**

### Method Summary

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Method Name</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getLayout()</td>
<td></td>
<td>Return the value of the layout property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyle()</td>
<td></td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>String</td>
<td>getStyleClass()</td>
<td></td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>void</td>
<td>restoreState(FacesContext _context, Object _state)</td>
<td></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td>saveState(FacesContext _context)</td>
<td></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td>setLayout(String layout)</td>
<td></td>
<td>Set the value of the layout property.</td>
</tr>
<tr>
<td>void</td>
<td>setStyle(String style)</td>
<td></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>void</td>
<td>setStyleClass(String styleClass)</td>
<td></td>
<td>Set the value of the styleClass property.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.faces.component.UIPanel

**getFamily**

### Methods inherited from class
javax.faces.component.UIComponentBase

Methods inherited from class javax.faces.component.UIComponent

class javax.faces.component.UIComponentBase

encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object

class java.lang.Object

class HTMLPanelGroup

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:

Constant Field Values

Constructor Detail

public HTMLPanelGroup()
HtmlPanelGroup

public HtmlPanelGroup()

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
</table>

public String getLayout()

  layout

  "block" renderer HTML "div"
  HTML "span"

getLayout

public String getLayout()

  Return the value of the layout property.

  Contents: The type of layout markup to use when rendering this group. If the value is "block" the renderer must produce an HTML "div" element. Otherwise HTML "span" element must be produced.

public void setLayout(String layout)

  layout

setLayout

public void setLayout(String layout)
Set the value of the layout property.

public String getStyle()

    style

  CSS

getStyle

public String getStyle()

    Return the value of the style property.
    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style

setStyle

public void setStyle(String style)

    Set the value of the style property.

public String getStyleClass()

    styleClass
CSS "class"

**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" property on generated markup.

**public void setStyleClass(String styleClass)**

styleClass

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the styleClass property.

**public Object saveState(FacesContext _context)**

saveState

```java
public Object saveState(FacesContext _context)
```

Description copied from interface: `StateHolder`

Gets the state of the instance as a Serializable Object.
If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIComponentBase`

```java
public void restoreState(FacesContext _context, Object _state)
```

**restoreState**

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface: StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the...
StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:

restoreState in interface StateHolder

Overrides:

restoreState in class UIComponentBase
javax.faces.component.html  **Class**  
**HtmlSelectBooleanCheckbox**

java.lang.Object  
  ↳ javax.faces.component.UIComponent  
  ↳ javax.faces.component.UIComponentBase  
  ↳ javax.faces.component.UIOutput  
  ↳ javax.faces.component.UIInput  
  ↳ javax.faces.component.html.HtmlSelectBoolean

**All Implemented Interfaces:**
  EditableValueHolder, StateHolder, ValueHolder

---

```java
class HtmlSelectBooleanCheckbox
extends UISelectBoolean

Extends: UIComponent > UIComponentBase > UIOutput > UIInput > UISelectBoolean
```

```java
checkbox HTML input value

rendererType "javax.faces.Checkbox" setRendererType()

```

Represents an HTML input element of type checkbox. The checkbox will be rendered as checked, or not, based on the value of the value property.

By default, the rendererType property must be set to "javax.faces.Checkbox". This value can be changed by calling the setRendererType() method.

---

**Field Summary**
static String COMPONENT_TYPE
The standard component type for this component.

Fields inherited from class
javax.faces.component.UISelectBoolean
COMPONENT_FAMILY

Fields inherited from class
javax.faces.component.UIInput
CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID, UPDATE_MESSAGE_ID

Fields inherited from class
javax.faces.component.UIComponent
bindings

Constructor Summary

(HtmlSelectBooleanCheckbox())

Method Summary

String getAccesskey()
Return the value of the accesskey property.

String getDir()
Return the value of the dir property.

String getLabel()
Return the value of the label property.

String getLang()
Return the value of the lang property.

String getOnblur()
Return the value of the onblur property.

String getOnchange()
Return the value of the onchange property.

String getOnclick()
Return the value of the onclick property.

getOndblclick()
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>Return the value of the <code>onfocus</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the <code>onselect</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>boolean isDisabled()</code></td>
<td>Return the value of the <code>disabled</code> property.</td>
</tr>
<tr>
<td><code>boolean isReadonly()</code></td>
<td>Return the value of the <code>readonly</code> property.</td>
</tr>
<tr>
<td><code>restoreState()</code></td>
<td></td>
</tr>
</tbody>
</table>

`restoreState(FacesContext _context, Object _state)`
void Perform any processing required to restore the state from the entries in the state Object.

Object

saveState(FacesContext _context)

Gets the state of the instance as a Serializable Object.

void setAccesskey(String accesskey)

Set the value of the accesskey property.

void setDir(String dir)

Set the value of the dir property.

void setDisabled(boolean disabled)

Set the value of the disabled property.

void setLabel(String label)

Set the value of the label property.

void setLang(String lang)

Set the value of the lang property.

void setOnblur(String onblur)

Set the value of the onblur property.

void setOnChange(String onchange)

Set the value of the onchange property.

void setOnclick(String onclick)

Set the value of the onclick property.

void setOndblclick(String ondblclick)

Set the value of the ondblclick property.

void setOnfocus(String onfocus)

Set the value of the onfocus property.

void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.

void setOnkeyup(String onkeyup)

Set the value of the onkeyup property.

void setOnmousedown(String onmousedown)

Set the value of the onmousedown property.

void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.

void setOnmouseover(String onmouseover)

Set the value of the onmouseover property.

void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.

void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.
Methods inherited from class `javax.faces.component.UISelectBoolean`

getFamily, getValueBinding, getValueExpression, isSelected, setSelected, setValueBinding, setValueExpression

Methods inherited from class `javax.faces.component.UIInput`

addValidator, addValueChangeListener, compareValues, decode, getConvertedValue, getConverterMessage, getRequiredMessage, getSubmittedValue, getValidator, getValidatorMessage, getValidators, getValueChangeListener, getValueChangeListeners, isImmediate, isLocalValueSet, isRequired, isValid, processDecodes, processUpdates, processValidators, removeValidator, removeValueChangeListener, resetValue, setConverterMessage, setImmediate, setLocalValueSet, setRequired, setRequiredMessage, setSubmittedValue, setValidator, setValidatorMessage, setValue, setValueChangeListeners, updateModel, validate, validateValue

Methods inherited from class `javax.faces.component.UIOutput`
### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

[Constant Field Values](#)
Constructor Detail

public HtmlSelectBooleanCheckbox()

HtmlSelectBooleanCheckbox

public HtmlSelectBooleanCheckbox()

Method Detail

public String getAccesskey()

accesskey

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

public void setAccesskey(String accesskey)

accesskey
**setAccesskey**

```java
public void setAccesskey(String accesskey)
```

Set the value of the accesskey property.

---

**public String getDir()**

```java
dir
"LTR" left-to-right "RTL" right-to-left
```

**getDir**

```java
public String getDir()
```

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

---

**public void setDir(String dir)**

```java
dir
```

**setDir**

```java
public void setDir(String dir)
```

Set the value of the dir property.
public boolean isDisabled()

disabled

false  true  disabled="disabled"

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled

setDisabled

public void setDisabled(boolean disabled)

Set the value of the disabled property.

public String getLabel()

label
**getLabel**

public String getLabel()

Return the value of the label property.

Contents: A localized user presentable name for this component.

---

**public void setLabel(String label)**

label

**setLabel**

public void setLabel(String label)

Set the value of the label property.

---

**public String getLang()**

lang

**getLang**

public String getLang()

Return the value of the lang property.
Public void setLang(String lang)

setLang

public void setLang(String lang)

Set the value of the lang property.

Public String getOnblur()

getOnblur

public String getOnblur()

Return the value of the onblur property.

Contents: Javascript code executed when this element loses focus.

Public void setOnblur(String onblur)

setOnblur

public void setOnblur(String onblur)

onblur
**setOnblur**

```java
public void setOnblur(String onblur)
```

Set the value of the `onblur` property.

---

**getOnchange()**

```java
public String getOnchange()
```

Return the value of the `onchange` property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

---

**setOnchange(String onchange)**

```java
public void setOnchange(String onchange)
```

---

**setOnchange**

```java
public void setOnchange(String onchange)
```

Set the value of the `onchange` property.
public String getOnclick()

  onclick

Javascript

getOnclick

public String getOnclick()

  Return the value of the onclick property.

  Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

  onclick

setOnclick

public void setOnclick(String onclick)

  Set the value of the onclick property.

public String getOndblclick()

  ondblclick

Javascript
getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.

    Contents: Javascript code executed when a pointer button is double clicked over this element.

public void setOndblclick(String ondblclick)

    ondblclick

setOndblclick

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

public String getOnfocus()

    onfocus

Javascript

getOnfocus

public String getOnfocus()

    Return the value of the onfocus property.
Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onfocus)

  onfocus

setOnfocus

public void setOnfocus(String onfocus)

  Set the value of the onfocus property.

public String getOnkeydown()

  onkeydown

Javascript

getOnkeydown

public String getOnkeydown()

  Return the value of the onkeydown property.

  Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

  onkeydown
setOnkeydown

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

public String getOnkeypress()

onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

Return the value of the onkeypress property.

Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.
public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup

public String getOnkeyup()

    Return the value of the onkeyup property.

    Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

    onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.

public String getOnmousedown()

    onmousedown

Javascript
getOnmousedown

public String getOnmousedown()

    Return the value of the onmousedown property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.

_____________________________________________________

public void setOnmousedown(String onmousedown)

  onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

_____________________________________________________

public String getOnmousemove()

  onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

    Return the value of the onmousemove property.
Contents: Javascript code executed when a pointer button is moved within this element.

```java
public void setOnmousemove(String onmousemove)

  onmousemove

setOnmousemove
```

```java
public void setOnmousemove(String onmousemove)

  Set the value of the onmousemove property.
```

```java
public String getOnmouseout()

  onmouseout

Javascript
```

```java
public String getOnmouseout()

  Return the value of the onmouseout property.

  Contents: Javascript code executed when a pointer button is moved away from this element.
```

```java
public void setOnmouseout(String onmouseout)

  onmouseout
```
setOnmouseout

public void setOnmouseout(String onmouseout)

    Set the value of the onmouseout property.

__________________________________________________________

getOnmouseover

public String getOnmouseover()

    onmouseover

Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

__________________________________________________________

setOnmouseover

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.
public String getOnmouseup()
  
  onmouseup

Javascript

getOnmouseup

public String getOnmouseup()

  Return the value of the onmouseup property.
  
  Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

  onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

  Set the value of the onmouseup property.

public String getOnselect()

  onselect

Javascript
getOnselect

public String getOnselect()

Return the value of the onselect property.

Contents: Javascript code executed when text within this element is selected by the user.

public void setOnselect(String onselect)

onselect

setOnselect

public void setOnselect(String onselect)

Set the value of the onselect property.

public boolean isReadonly()

readonly

false true readonly="readonly"

isReadonly

public boolean isReadonly()
Return the value of the `readonly` property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `readonly="readonly"`.

---

**public void setReadonly(boolean readonly)**

```java
readonly
```

---

**public String getStyle()**

```java
style
```

---

**CSS**

```java
public String getStyle()
```

Return the value of the `style` property.

Contents: CSS style(s) to be applied when this component is rendered.
public void setStyle(String style)

setState

public void setStyle(String style)

Set the value of the style property.

public String getStyleClass()

getStyleClass

public String getStyleClass()

Return the value of the styleClass property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

setStyleClass

public void setStyleClass(String styleClass)

styleClass
**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the styleClass property.

---

**public String getTabindex()**

`tabindex`

**Tab 0 32767**

---

**getTabindex**

```java
public String getTabindex()
```

Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

---

**public void setTabindex(String tabindex)**

`tabindex`

---

**setTabindex**

```java
public void setTabindex(String tabindex)
```

Set the value of the `tabindex` property.
public String getTitle()

title

getTitle

public String getTitle()

    Return the value of the title property.

    Contents: Advisory title information about markup elements generated for this component.

public void setTitle(String title)

title

setTitle

public void setTitle(String title)

    Set the value of the title property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)
Description copied from interface: **StateHolder**

Gets the state of the instance as a **Serializable** Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the **StateManager**

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**
`saveState` in interface **StateHolder**

**Overrides:**
`saveState` in class **UIInput**

---

**public void restoreState(FacesContext _context, Object _state)**

**restoreState**

```java
public void restoreState(FacesContext _context,
Object _state)
```

**Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.
If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the 
StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

** Specified by:**
restoreState in interface StateHolder

** Overrides:**
restoreState in class UIInput

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public class HtmlSelectManyCheckbox

extends UISelectMany

**Extends:** UIComponent > UIComponentBase > UIOutput > UIInput > UISelectMany

checkbox HTML input

rendererType "javax.faces.Checkbox" setRendererType()

Represents a multiple-selection component that is rendered as a set of HTML input elements of type checkbox.

By default, the rendererType property must be set to "javax.faces.Checkbox". This value can be changed by calling the setRendererType() method.

---

**Field Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>
### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HtmlSelectManyCheckbox()</code></td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getAccesskey()</code></td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td><code>int getBorder()</code></td>
<td>Return the value of the border property.</td>
</tr>
<tr>
<td><code>String getDir()</code></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td><code>String getDisabledClass()</code></td>
<td>Return the value of the disabledClass property.</td>
</tr>
<tr>
<td><code>String getEnabledClass()</code></td>
<td>Return the value of the enabledClass property.</td>
</tr>
<tr>
<td><code>String getLabel()</code></td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td><code>String getLang()</code></td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td><code>String getLayout()</code></td>
<td>Return the value of the layout property.</td>
</tr>
<tr>
<td><code>String getOnblur()</code></td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>String</td>
<td>getOnchange()</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
</tr>
<tr>
<td>String</td>
<td>getOnclick()</td>
</tr>
<tr>
<td>String</td>
<td>getOndblclick()</td>
</tr>
<tr>
<td>String</td>
<td>getOnfocus()</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeydown()</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeypress()</td>
</tr>
<tr>
<td>String</td>
<td>getOnkeyup()</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousedown()</td>
</tr>
<tr>
<td>String</td>
<td>getOnmousemove()</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseout()</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseover()</td>
</tr>
<tr>
<td>String</td>
<td>getOnmouseup()</td>
</tr>
<tr>
<td>String</td>
<td>getOnselect()</td>
</tr>
<tr>
<td>String</td>
<td>getStyle()</td>
</tr>
<tr>
<td>String</td>
<td>getStyleClass()</td>
</tr>
<tr>
<td>String</td>
<td>getTabIndex()</td>
</tr>
<tr>
<td>String</td>
<td>getTitle()</td>
</tr>
</tbody>
</table>
boolean isDisabled()
    Return the value of the disabled property.

boolean isReadonly()
    Return the value of the readonly property.

void restoreState(FacesContext _context, Object _state)
    Perform any processing required to restore the state from the entries in the state Object.

Object saveState(FacesContext _context)
    Gets the state of the instance as a Serializable Object.

void setAccesskey(String accesskey)
    Set the value of the accesskey property.

void setBorder(int border)
    Set the value of the border property.

void setDir(String dir)
    Set the value of the dir property.

void setDisabled(boolean disabled)
    Set the value of the disabled property.

void setDisabledClass(String disabledClass)
    Set the value of the disabledClass property.

void setEnabledClass(String enabledClass)
    Set the value of the enabledClass property.

void setLabel(String label)
    Set the value of the label property.

void setLang(String lang)
    Set the value of the lang property.

void setLayout(String layout)
    Set the value of the layout property.

void setOnblur(String onblur)
    Set the value of the onblur property.

void setOnchange(String onchange)
    Set the value of the onchange property.

void setOnclick(String onclick)
    Set the value of the onclick property.

void setOndblclick(String ondblclick)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td>void <code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td>void <code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td>void <code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td>void <code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td>void <code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td>void <code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td>void <code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td>void <code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td>void <code>setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td>void <code>setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td>void <code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
<tr>
<td>void <code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>void <code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td>void <code>setTabIndex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.component.UISelectMany:
- `compareValues`,
- `getFamily`,
- `getSelectedValues`,
- `getValueBinding`.
**Methods inherited from class javax.faces.component.UIInput**

getValueExpression, setSelectedValues, setValueBinding, setValueExpression, validateValue

**Methods inherited from class javax.faces.component.UIOutput**
getConverter, getLocalValue, getValue, setConverter

**Methods inherited from class javax.faces.component.UIComponentBase**
addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient

**Methods inherited from class javax.faces.component.UIComponent**
encodeAll, getContainerClientId

**Methods inherited from class java.lang.Object**
cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Methods inherited from interface javax.faces.component.ValueHolder**
getConverter, getLocalValue, getValue, setConverter
Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**
- [Constant Field Values](#)

Constructor Detail

```java
public HtmlSelectManyCheckbox()
```

**HtmlSelectManyCheckbox**

```java
public HtmlSelectManyCheckbox()
```

Method Detail

```java
public String getAccesskey()
```

accesskey

getAccesskey
public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

public void setAccesskey(String accesskey)

accesskey

setAccesskey

public void setAccesskey(String accesskey)

Set the value of the accesskey property.

public int getBorder()

border

g.Border

public int getBorder()

Return the value of the border property.

Contents: Width (in pixels) of the border to be drawn around the table containing the options list.
public void setBorder(int border)

border

setBorder

class setBorder

public void setBorder(int border)

Set the value of the border property.

public String getDir()

dir

"LTR"left-to-right "RTL"right-to-left

getDir

class getDir

public String getDir()

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

public void setDir(String dir)

dir
setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isDisabled()

disabled

false true disabled="disabled"

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled

setDisabled

public void setDisabled(boolean disabled)

Set the value of the disabled property.
public String getDisabledClass()

    disabledClass

CSS

getDisabledClass

public String getDisabledClass()

    Return the value of the disabledClass property.
    
    Contents: CSS style class to apply to the rendered label on disabled options.

public void setDisabledClass(String disabledClass)

    disabledClass

setDisabledClass

public void setDisabledClass(String disabledClass)

    Set the value of the disabledClass property.

public String getEnabledClass()

    enabledClass

CSS
**getEnabledClass**

public String getEnabledClass()

Return the value of the enabledClass property.

Contents: CSS style class to apply to the rendered label on enabled options.

**public void setEnabledClass(String enabledClass)**

enabledClass

**setLabel**

public void setEnabledClass(String enabledClass)

Set the value of the enabledClass property.

**public String getLabel()**

label

**getLabel**

public String getLabel()

Return the value of the label property.
public void setLabel(String label)

setLabel

public String getLang()

getLang

Contents: Code describing the language used in the generated markup for this component.
setLang

public void setLang(String lang)

    Set the value of the lang property.

getLayout

public String getLayout()

    layout

"pageDirection" "lineDirection"
"lineDirection"

getLayout

public String getLayout()

    Return the value of the layout property.

    Contents: Orientation of the options list to be created. Valid values are "pageDirection" (list is laid out vertically), or "lineDirection" (list is laid out horizontally). If not specified, the default value is "lineDirection".

setLayout

public void setLayout(String layout)

    layout

setLayout
public void setLayout(String layout)

    Set the value of the layout property.

public String getOnblur()

    onblur

    Javascript

getOnblur

public String getOnblur()

    Return the value of the onblur property.

    Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

    onblur

setOnblur

public void setOnblur(String onblur)

    Set the value of the onblur property.

public String getOnchange()

    onchange
getOnchange

```java
public String getOnchange()
```

Return the value of the onchange property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

---

setOnchange

```java
public void setOnchange(String onchange)
```

Events:

```
onchange
```

setOnchange

```java
public void setOnchange(String onchange)
```

Set the value of the onchange property.

---

getOnclick

```java
public String getOnclick()
```

Events:

```
onclick
```

getOnclick

```java
public String getOnclick()
```
Return the value of the `onclick` property.

Contents: Javascript code executed when a pointer button is clicked over this element.

```java
public void setOnclick(String onclick)
```

`onclick`

`setOnclick`

```java
public void setOnclick(String onclick)
```

Set the value of the `onclick` property.

---

```java
public String getOndblclick()
```

`ondblclick`

`Javascript`

`getOndblclick`

```java
public String getOndblclick()
```

Return the value of the `ondblclick` property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

```java
public void setOndblclick(String ondblclick)
```
setOndblclick

public void setOndblclick(String ondblclick)

Set the value of the ondblclick property.

public String getOnfocus()

onfocus

Javascript

getOnfocus

public String getOnfocus()

Return the value of the onfocus property.

Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onfocus)

onfocus

setOnfocus

public void setOnfocus(String onfocus)
Set the value of the onfocus property.

public String getOnkeydown()

onkeydown

Javascript

getOnkeydown

public String getOnkeydown()

Return the value of the onkeydown property.

Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

onkeydown

setOnkeydown

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

public String getOnkeypress()
getOnkeypress

public String getOnkeypress()

    Return the value of the onkeypress property.

    Contents: Javascript code executed when a key is pressed and released over this element.

-----------------------------------------------

public void setOnkeypress(String onkeypress)

    onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

    Set the value of the onkeypress property.

-----------------------------------------------

public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup

public String getOnkeyup()
Return the value of the onkeyup property.

Contents: Javascript code executed when a key is released over this element.

```java
public void setOnkeyup(String onkeyup)
    onkeyup
```

### setOnkeyup

```java
public void setOnkeyup(String onkeyup)
    Set the value of the onkeyup property.
```

### public String getOnmousedown()

```java
public String getOnmousedown()
    onmousedown

Javascript
```

### getOnmousedown

```java
public String getOnmousedown()
    Return the value of the onmousedown property.

Contents: Javascript code executed when a pointer button is pressed down over this element.
```

```java
public void setOnmousedown(String onmousedown)
```

onmousedown

**setOnmousedown**

```java
public void setOnmousedown(String onmousedown)
```

Set the value of the `onmousedown` property.

---

**public String getOnmousemove()**

```java
onmousemove
```

Javascript

**getOnmousemove**

```java
public String getOnmousemove()
```

Return the value of the `onmousemove` property.

Contents: Javascript code executed when a pointer button is moved within this element.

---

**public void setOnmousemove(String onmousemove)**

```java
onmousemove
```

**setOnmousemove**

```java
public void setOnMouseMove(String onmousemove)
```
Set the value of the onmousemove property.

public String getOnmouseout()

onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.

public String getOnmouseover()
Javascript

getOnmouseover

public String getOnmouseover()

  Return the value of the onmouseover property.
  Contents: Javascript code executed when a pointer button is moved onto this element.

-----------------------------------------------

public void setOnmouseover(String onmouseover)

  onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

  Set the value of the onmouseover property.

-----------------------------------------------

public String getOnmouseup()

  onmouseup

Javascript

getOnmouseup

public String getOnmouseup()
Return the value of the onmouseup property.

Contents: Javascript code executed when a pointer button is released over this element.

```java
public void setOnmouseup(String onmouseup)
```

`onmouseup`

---

**setOnmouseup**

public void setOnmouseup(String onmouseup)

Set the value of the onmouseup property.

---

**public String getOnselect()**

`onselect`

**Javascript**

---

**getOnselect**

public String getOnselect()

Return the value of the onselect property.

Contents: Javascript code executed when text within this element is selected by the user.

---

```java
public void setOnselect(String onselect)
```
onselect

setOnselect

public void setOnselect(String onSelect)

Set the value of the onSelect property.

------

public boolean isReadonly()

readonly

false true readonly="readonly"

------

isReadonly

public boolean isReadonly()

Return the value of the readonly property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".

------

public void setReadonly(boolean readonly)

readonly
setReadonly

public void setReadonly(boolean readonly)

Set the value of the readonly property.

public String getStyle()

style

CSS

getAddress

public String getStyle()

Return the value of the style property.

Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

style

setStyle

public void setStyle(String style)

Set the value of the style property.
public String getStyleClass()

    styleClass

    CSS "class"

getStyleClass

public String getStyleClass()

    Return the value of the styleClass property.

    Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

    styleClass

setStyleClass

public void setStyleClass(String styleClass)

    Set the value of the styleClass property.

public String getTabindex()

    tabindex

Tab  0  32767
**getTabindex**

```java
public String getTabindex()
```

Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

**public void setTabindex(String tabindex)**

```java
public void setTabindex(String tabindex)
```

Set the value of the `tabindex` property.

**getTitle**

```java
public String getTitle()
```

**getTitle**

```java
public String getTitle()
```

Return the value of the `title` property.
public void setTitle(String title)

setTitle

public void setTitle(String title)

Set the value of the title property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to
instances that implement StateHolder (such as a UIComponent with
event handlers, validators, etc.) this method must call the
StateHolder.saveState(javax.faces.context.FacesContext) method
on all those instances as well. This method must not save the
state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In
other words, after executing this code:

Object state = component.saveState(facesContext);
component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
    saveState in interface StateHolder
Overrides:
    saveState in class UIInput

public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
    restoreState in interface StateHolder
Overrides:
    restoreState in class UIInput

Overview Package Tree Deprecated Index Help
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component.html  Class HtmlSelectManyListbox

java.lang.Object
    ↓ javax.faces.component.UIComponent
       ↓ javax.faces.component.UIComponentBase
          ↓ javax.faces.component.UIOutput
             ↓ javax.faces.component.UIInput
                ↓ javax.faces.component.html.HtmlSelectMany

All Implemented Interfaces:
    EditableValueHolder, StateHolder, ValueHolder

public class HtmlSelectManyListbox

extends UISelectMany

Extends: UIComponent > UIComponentBase > UIOutput > UIInput > UISelectMany

HTML  select

    rendererType "javax.faces.Listbox"  setRendererType()

Represents a multiple-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.

By default, the rendererType property must be set to "javax.faces.Listbox". This value can be changed by calling the setRendererType() method.

Field Summary

| COMPONENT_TYPE |
The standard component type for this component.

Fields inherited from class javax.faces.component.UISelectMany
COMPONENT_FAMILY, INVALID_MESSAGE_ID

Fields inherited from class javax.faces.component.UlInput
CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID, UPDATE_MESSAGE_ID

Fields inherited from class javax.faces.component.UIComponent bindings

Constructor Summary

HtmlSelectManyListbox()

Method Summary

String getAccesskey()
Return the value of the accesskey property.

String getDir()
Return the value of the dir property.

String getDisabledClass()
Return the value of the disabledClass property.

String getEnabledClass()
Return the value of the enabledClass property.

String getLabel()
Return the value of the label property.

String getLang()
Return the value of the lang property.

String getOnblur()
Return the value of the onblur property.

String getOnchange()
Return the value of the onchange property.
String `getOnclick()`
Return the value of the `onclick` property.

String `getOndblclick()`
Return the value of the `ondblclick` property.

String `getOnfocus()`
Return the value of the `onfocus` property.

String `getOnkeydown()`
Return the value of the `onkeydown` property.

String `getOnkeypress()`
Return the value of the `onkeypress` property.

String `getOnkeyup()`
Return the value of the `onkeyup` property.

String `getOnmousedown()`
Return the value of the `onmousedown` property.

String `getOnmousemove()`
Return the value of the `onmousemove` property.

String `getOnmouseout()`
Return the value of the `onmouseout` property.

String `getOnmouseover()`
Return the value of the `onmouseover` property.

String `getOnmouseup()`
Return the value of the `onmouseup` property.

String `getOnselect()`
Return the value of the `onselect` property.

int `getSize()`
Return the value of the `size` property.

String `getStyle()`
Return the value of the `style` property.

String `getStyleClass()`
Return the value of the `styleClass` property.

String `getTabindex()`
Return the value of the `tabindex` property.

String `getTitle()`
Return the value of the `title` property.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean <code>isDisabled()</code></td>
<td>Return the value of the disabled property.</td>
</tr>
<tr>
<td>boolean <code>isReadonly()</code></td>
<td>Return the value of the readonly property.</td>
</tr>
<tr>
<td>void <code>restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object <code>saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void <code>setAccesskey(String accesskey)</code></td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td>void <code>setDir(String dir)</code></td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td>void <code>setDisabled(boolean disabled)</code></td>
<td>Set the value of the disabled property.</td>
</tr>
<tr>
<td>void <code>setDisabledClass(String disabledClass)</code></td>
<td>Set the value of the disabledClass property.</td>
</tr>
<tr>
<td>void <code>setEnabledClass(String enabledClass)</code></td>
<td>Set the value of the enabledClass property.</td>
</tr>
<tr>
<td>void <code>setLabel(String label)</code></td>
<td>Set the value of the label property.</td>
</tr>
<tr>
<td>void <code>setLang(String lang)</code></td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td>void <code>setOnblur(String onblur)</code></td>
<td>Set the value of the onblur property.</td>
</tr>
<tr>
<td>void <code>setOnchange(String onchange)</code></td>
<td>Set the value of the onchange property.</td>
</tr>
<tr>
<td>void <code>setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td>void <code>setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td>void <code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td>void <code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
</tbody>
</table>
Set the value of the `onkeydown` property.

```java
void setOnkeypress(String onkeypress)
    Set the value of the `onkeypress` property.
```

Set the value of the `onkeyup` property.

```java
void setOnkeyup(String onkeyup)
    Set the value of the `onkeyup` property.
```

Set the value of the `onmousedown` property.

```java
void setOnmousedown(String onmousedown)
    Set the value of the `onmousedown` property.
```

Set the value of the `onmousemove` property.

```java
void setOnmousemove(String onmousemove)
    Set the value of the `onmousemove` property.
```

Set the value of the `onmouseout` property.

```java
void setOnmouseout(String onmouseout)
    Set the value of the `onmouseout` property.
```

Set the value of the `onmouseover` property.

```java
void setOnmouseover(String onmouseover)
    Set the value of the `onmouseover` property.
```

Set the value of the `onmouseup` property.

```java
void setOnmouseup(String onmouseup)
    Set the value of the `onmouseup` property.
```

Set the value of the `onselect` property.

```java
void setOnselect(String onselect)
    Set the value of the `onselect` property.
```

Set the value of the `readonly` property.

```java
void setReadonly(boolean readonly)
    Set the value of the `readonly` property.
```

Set the value of the `size` property.

```java
void setSize(int size)
    Set the value of the `size` property.
```

Set the value of the `style` property.

```java
void setStyle(String style)
    Set the value of the `style` property.
```

Set the value of the `styleClass` property.

```java
void setStyleClass(String styleClass)
    Set the value of the `styleClass` property.
```

Set the value of the `tabindex` property.

```java
void setTabindex(String tabindex)
    Set the value of the `tabindex` property.
```

Set the value of the `title` property.

```java
void setTitle(String title)
    Set the value of the `title` property.
```

Methods inherited from class `javax.faces.component.UISelectMany`

- `compareValues`, `getFamily`, `getSelectedValues`, `getValueBinding`, `getValueExpression`, `setSelectedValues`, `setValueBinding`, `setValueExpression`, `validateValue`
### Methods inherited from class `javax.faces.component.UIInput`  
- `addValidator`, `addValueChangeListener`, `decode`, `getConvertedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`  

### Methods inherited from class `javax.faces.component.UIOutput`  
- `getConverter`, `getLocalValue`, `getValue`, `setConverter`  

### Methods inherited from class `javax.faces.component.UIComponentBase`  
- `addFacesListener`, `broadcast`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getContainerClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `invokeOnComponent`, `isRendered`, `isTransient`, `processRestoreState`, `processSaveState`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`  

### Methods inherited from class `javax.faces.component.UIComponent`  
- `encodeAll`, `getContainerClientId`  

### Methods inherited from class `java.lang.Object`  
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`  

### Methods inherited from interface `javax.faces.component.ValueHolder`  
- `getConverter`, `getLocalValue`, `getValue`, `setConverter`
Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

See Also:
- [Constant Field Values](Constant Field Values)

Constructor Detail

```java
public HtmlSelectManyListbox()
```

```java
HtmlSelectManyListbox
```

```java
public HtmlSelectManyListbox()
```

Method Detail

```java
public String getAccesskey()
```

```java
accesskey
```

getAccesskey

```java
public String getAccesskey()
```
Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

```java
public void setAccesskey(String accesskey)
```

setAccesskey

```java
public void setAccesskey(String accesskey)
```

Set the value of the accesskey property.

```java
public String getDir()
```

dir

"LTR" left-to-right "RTL" right-to-left

getDir

```java
public String getDir()
```

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
public void setDir(String dir)

dir

setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isDisabled()

disabled

false true disabled="disabled"

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled
setDisabled

public void setDisabled(boolean disabled)

Set the value of the disabled property.

public String getDisabledClass()

disabledClass

CSS

getDisabledClass

public String getDisabledClass()

Return the value of the disabledClass property.

Contents: CSS style class to apply to the rendered label on disabled options.

public void setDisabledClass(String disabledClass)

disabledClass

setDisabledClass

public void setDisabledClass(String disabledClass)

Set the value of the disabledClass property.
public String getEnabledClass()

    enabledClass

CSS

getEnabledClass

public String getEnabledClass()

    Return the value of the enabledClass property.
    Contents: CSS style class to apply to the rendered label on enabled options.

public void setEnabledClass(String enabledClass)

    enabledClass

setLabel

public String getLabel()

    label
**getLabel**

public String getLabel()

    Return the value of the label property.

    Contents: A localized user presentable name for this component.

---

**public void setLabel(String label)**

      label

**setLabel**

public void setLabel(String label)

    Set the value of the label property.

---

**public String getLang()**

      lang

**getLang**

public String getLang()

    Return the value of the lang property.
public void setLang(String lang)

    lang

setLang

public void setLang(String lang)

    Set the value of the lang property.

public String getOnblur()

    onblur

Javascript

getOnblur

public String getOnblur()

    Return the value of the onblur property.

    Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

    onblur
**setOnblur**

```java
public void setOnblur(String onblur)
```

Set the value of the `onblur` property.

---

**public String getOnchange()**

```java
public String getOnchange()
```

Return the value of the `onchange` property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

---

**public void setOnchange(String onchange)**

```java
public void setOnchange(String onchange)
```

---

**setOnchange**

```java
public void setOnchange(String onchange)
```

Set the value of the `onchange` property.
public String getOnclick()

    onclick

Javascript

getOnclick

public String getOnclick()

    Return the value of the onclick property.

    Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

    onclick

setOnclick

public void setOnclick(String onclick)

    Set the value of the onclick property.

public String getOndblclick()

    ondblclick

Javascript
**getOndblclick**

public String getOndblclick()

Return the value of the ondblclick property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

---

**public void setOndblclick(String ondblclick)**

  ondblclick

**setOndblclick**

public void setOndblclick(String ondblclick)

  Set the value of the ondblclick property.

---

**public String getOnfocus()**

  onfocus

Javascript

**getOnfocus**

public String getOnfocus()

  Return the value of the onfocus property.
Contents: Javascript code executed when this element receives focus.

---

`public void setOnfocus(String onfocus)`

`onfocus`

 **setOnfocus**

`public void setOnfocus(String onfocus)`

Set the value of the `onfocus` property.

---

`public String getOnkeydown()`

`onkeydown`

**Javascript**

**getOnkeydown**

`public String getOnkeydown()`

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

---

`public void setOnkeydown(String onkeydown)`

`onkeydown`
**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.

---

**public String getOnkeypress()**

```java
    onkeypress
```

**Javascript**

**getOnkeypress**

```java
public String getOnkeypress()
```

Return the value of the `onkeypress` property.

Contents: Javascript code executed when a key is pressed and released over this element.

---

**public void setOnkeypress(String onkeypress)**

```java
    onkeypress
```

**setOnkeypress**

```java
public void setOnkeypress(String onkeypress)
```

Set the value of the `onkeypress` property.
public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup

public String getOnkeyup()

    Return the value of the onkeyup property.

    Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

    onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.

public String getOnmousedown()

    onmousedown

Javascript
getOnmousedown

public String getOnmousedown()

Return the value of the onmousedown property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

---------------------------------

public void setOnmousedown(String onmousedown)

onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

Set the value of the onmousedown property.

---------------------------------

public String getOnmousemove()

onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

Return the value of the onmousemove property.
Contents: Javascript code executed when a pointer button is moved within this element.

```java
public void setOnmousemove(String onmousemove)

    onmousemove

setOnmousemove
```

Set the value of the `onmousemove` property.

```java
public String getOnmouseout()

    onmouseout

Javascript
```

Return the value of the `onmouseout` property.

Contents: Javascript code executed when a pointer button is moved away from this element.

```java
public void setOnmouseout(String onmouseout)

    onmouseout
```
setOnmouseout

public void setOnmouseout(String onmouseout)

    Set the value of the onmouseout property.

-----------------------------------------------

getOnmouseover

public String getOnmouseover()

    onmouseover

    Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

-----------------------------------------------

setOnmouseover

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.
public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public String getOnmouseup()

    Return the value of the onmouseup property.

    Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getOnselect()

    onselect

Javascript
getOnselect

public String getOnselect()

    Return the value of the onclick property.
    Contents: Javascript code executed when text within this element is selected by the user.

public void setOnselect(String onclick)

    onclick

setOnselect

public void setOnselect(String onclick)

    Set the value of the onclick property.

public boolean isReadonly()

    readonly

    false true readonly="readonly"

isReadonly

public boolean isReadonly()
Return the value of the `readonly` property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `readonly="readonly"`.

---

**public void setReadonly(boolean readonly)**

```
readonly
```

---

**setReadonly**

**public void setReadonly(boolean readonly)**

Set the value of the `readonly` property.

---

**public int getSize()**

```
size
```

---

**getSize**

**public int getSize()**

Return the value of the `size` property.

Contents: Number of available options to be shown at all times. If not specified, all available options are shown.
public void setSize(int size)

    size

setSize

public void setSize(int size)

    Set the value of the size property.

public String getStyle()

    style

CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style
**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.

---

**public String getStyleClass()**

```
styleClass
```

CSS "class"

---

**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**public void setStyleClass(String styleClass)**

```
styleClass
```

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.
public String getTabindex()

    tabindex

    Tab  0  32767

getTabindex

public String getTabindex()

    Return the value of the tabindex property.
    
    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

----------------------------------------------

public void setTabindex(String tabindex)

    tabindex

setTabindex

public void setTabindex(String tabindex)

    Set the value of the tabindex property.

----------------------------------------------

public String getTitle()

    title
**getTitle**

public `String` `getTitle()`

    Return the value of the `title` property.

    Contents: Advisory title information about markup elements generated for this component.

---

**public void setTitle(String title)**

    `title`

---

**setTitle**

public `void` `setTitle(String title)`

    Set the value of the `title` property.

---

**public Object saveState(FacesContext _context)**

**saveState**

public `Object` `saveState(FacesContext _context)`

    Description copied from interface: `StateHolder`

    Gets the state of the instance as a `Serializable` Object.

    If the class that implements this interface has references to instances that implement `StateHolder` (such as a `UIComponent` with event handlers, validators, etc.) this method must call the
**StateHolder.saveState(javax.faces.context.FacesContext)** method on all those instances as well. **This method must not save the state of children and facets.** That is done via the **StateManager**

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

*component* should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**

- **saveState** in interface **StateHolder**

**Overrides:**

- **saveState** in class **UIInput**

---

**public void restoreState(FacesContext _context, Object _state)**

**restoreState**

```java
public void restoreState(FacesContext _context, Object _state)
```

**Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the **StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)** method on all those instances as well.
Specified by:  
restoreState in interface StateHolder

Overrides:  
restoreState in class UIInput
**javax.faces.component.html**  
**Class** HtmlSelectManyMenu

```java
java.lang.Object
  └─javax.faces.component.UIComponent
      └─javax.faces.component.UIComponentBase
          └─javax.faces.component.UIOutput
              └─javax.faces.component.UIInput
                  └─javax.faces.component.UISelectMany
                      └─javax.faces.component.html.HtmlSelectManyMenu
```

**All Implemented Interfaces:**  
EditableValueHolder, StateHolder, ValueHolder

---

```java
public class HtmlSelectManyMenu

extends UISelectMany

**Extends:** UIComponent > UIComponentBase > UIOutput > UInput > UISelectMany

HTML select

    rendererType "javax.faces.Menu"    setRendererType()
```

Represents a multiple-selection component that is rendered as an HTML select element, showing a single available option at a time.

By default, the rendererType property must be set to "javax.faces.Menu". This value can be changed by calling the setRendererType() method.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>
### Fields inherited from class `javax.faces.component.UISelectMany`  
- `COMPONENT_FAMILY`, `INVALID_MESSAGE_ID`

### Fields inherited from class `javax.faces.component.UIInput`  
- `CONVERSION_MESSAGE_ID`, `REQUIRED_MESSAGE_ID`, `UPDATE_MESSAGE_ID`

### Fields inherited from class `javax.faces.component.UIComponent`  
- `bindings`

### Constructor Summary

**HtmlSelectManyMenu()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String <code>getAccesskey()</code></td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td>String <code>getDir()</code></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td>String <code>getDisabledClass()</code></td>
<td>Return the value of the disabledClass property.</td>
</tr>
<tr>
<td>String <code>getEnabledClass()</code></td>
<td>Return the value of the enabledClass property.</td>
</tr>
<tr>
<td>String <code>getLabel()</code></td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td>String <code>getLang()</code></td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String <code>getOnblur()</code></td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>String <code>getOnchange()</code></td>
<td>Return the value of the onchange property.</td>
</tr>
<tr>
<td>String <code>getonclick()</code></td>
<td>Return the value of the onclick property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>Return the value of the <code>onfocus</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the <code>onselect</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
<tr>
<td><code>isDisabled()</code></td>
<td>Return the value of the <code>disabled</code> property.</td>
</tr>
<tr>
<td><code>isReadonly()</code></td>
<td>Return the value of the <code>readonly</code> property.</td>
</tr>
</tbody>
</table>
```java
void restoreState(FacesContext _context, Object _state)
    Perform any processing required to restore the state from
    the entries in the state Object.

Object saveState(FacesContext _context)
    Gets the state of the instance as a Serializable Object.

void setAccesskey(String accesskey)
    Set the value of the accesskey property.

void setDir(String dir)
    Set the value of the dir property.

void setDisabled(boolean disabled)
    Set the value of the disabled property.

void setDisabledClass(String disabledClass)
    Set the value of the disabledClass property.

void setEnabledClass(String enabledClass)
    Set the value of the enabledClass property.

void setLabel(String label)
    Set the value of the label property.

void setLang(String lang)
    Set the value of the lang property.

void setOnblur(String onblur)
    Set the value of the onblur property.

void setOnchange(String onchange)
    Set the value of the onchange property.

void setOnclick(String onclick)
    Set the value of the onclick property.

void setOndblclick(String ondblclick)
    Set the value of the ondblclick property.

void setOnfocus(String onfocus)
    Set the value of the onfocus property.

void setOnkeydown(String onkeydown)
    Set the value of the onkeydown property.

void setOnkeypress(String onkeypress)
    Set the value of the onkeypress property.

void setOnkeyup(String onkeyup)
    Set the value of the onkeyup property.
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UISelectMany`:
- `compareValues`, `getFamily`, `getSelectedValues`, `getValueBinding`, `getValueExpression`, `setSelectedValues`, `setValueBinding`, `setValueExpression`, `validateValue`  

Methods inherited from class `javax.faces.component.UITextField`:
- `addValidator`, `addValueChangeListener`, `decode`, `getConvertedValue`, `getConverterMessage`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`

| Methods inherited from class javax.faces.component.UIOutput |
| getConverter, getLocalValue, getValue, setConverter |

| Methods inherited from class javax.faces.component.UIComponentBase |
| addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient |

| Methods inherited from class javax.faces.component.UIComponent |
| encodeAll, getContainerClientId |

| Methods inherited from class java.lang.Object |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait |

| Methods inherited from interface javax.faces.component.ValueHolder |
| getConverter, getLocalValue, getValue, setConverter |

**Field Detail**

COMPONENT_TYPE
public static final String COMPONENT_TYPE

The standard component type for this component.

See Also: Constant Field Values

Constructor Detail

public HtmlSelectManyMenu()

HtmlSelectManyMenu

public HtmlSelectManyMenu()

Method Detail

public String getAccesskey()

accesskey

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.
public void setAccesskey(String accesskey)

accesskey

setAccesskey

public void setAccesskey(String accesskey)

Set the value of the accesskey property.

public String getDir()

dir

"LTR" left-to-right "RTL" right-to-left

getDir

public String getDir()

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
**setDir**

```java
public void setDir(String dir)
```

Set the value of the `dir` property.

---

**public boolean isDisabled()**

```java
disabled
false true disabled="disabled"
```

**isDisabled**

```java
public boolean isDisabled()
```

Return the value of the `disabled` property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

---

**public void setDisabled(boolean disabled)**

```java
disabled
```

**setDisabled**

```java
public void setDisabled(boolean disabled)
```

Set the value of the `disabled` property.
public String getDisabledClass()

disabledClass

CSS

getDisabledClass

public String getDisabledClass()

Return the value of the disabledClass property.

Contents: CSS style class to apply to the rendered label on disabled options.

public void setDisabledClass(String disabledClass)

disabledClass

setDisabledClass

public void setDisabledClass(String disabledClass)

Set the value of the disabledClass property.

public String getEnabledClass()

enabledClass

CSS
**getEnabledClass**

```java
public String getEnabledClass()
```

Return the value of the enabledClass property.

Contents: CSS style class to apply to the rendered label on enabled options.

**public void setEnabledClass(String enabledClass)**

```java
  enabledClass
```

**setLabel**

```java
public String setEnabledClass(String enabledClass)
```

Set the value of the enabledClass property.

---

**getLabel**

```java
public String getLabel()
```

```java
  label
```

Return the value of the label property.
public void setLabel(String label)

  label

setLabel

public void setLabel(String label)

  Set the value of the label property.

getLang

public String getLang()

  lang

getLang

public String getLang()

  Return the value of the lang property.

  Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

  lang
**setLang**

public void setLang(String lang)

Set the value of the lang property.

---

**public String getOnblur()**

onblur

Javascript

**getOnblur**

public String getOnblur()

Return the value of the onblur property.

Contents: Javascript code executed when this element loses focus.

---

**public void setOnblur(String onblur)**

onblur

**setOnblur**

public void setOnblur(String onblur)

Set the value of the onblur property.
public String getOnchange()

    onchange

Javascript

getOnchange

public String getOnchange()

    Return the value of the onchange property.

    Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

public void setOnchange(String onchange)

    onchange

setOnchange

public void setOnchange(String onchange)

    Set the value of the onchange property.

public String getOnclick()

    onclick

Javascript
getOnclick

public String getOnclick()

    Return the value of the onclick property.

    Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

    onclick

setOnclick

public void setOnclick(String onclick)

    Set the value of the onclick property.

public String getOndblclick()

    ondblclick

Javascript

getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.
Contents: Javascript code executed when a pointer button is double clicked over this element.

```java
public void setOndblclick(String ondblclick)

    ondblclick
```

**setOndblclick**

```java
public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.
```

```java
public String getOnfocus()

    onfocus
```

**getOnfocus**

```java
public String getOnfocus()

    Return the value of the onfocus property.
    
    Contents: Javascript code executed when this element receives focus.
```

```java
public void setOnfocus(String onfocus)

    onfocus
```
**setOnfocus**

```java
public void setOnfocus(String onFocus)
```

Set the value of the `onFocus` property.

---

**public String getOnkeydown()**

```java
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

---

**public void setOnkeydown(String onkeydown)**

```java
public void setOnkeydown(String onkeydown)
```

---

**setOnkeydown**

```java
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.
public String getOnkeypress()

    onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

    Return the value of the onkeypress property.

    Contents: Javascript code executed when a key is pressed and released over this element.

public void setOnkeypress(String onkeypress)

    onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

    Set the value of the onkeypress property.

public String getOnkeyup()

    onkeyup

Javascript
getOnkeyup

public String getOnkeyup()

Return the value of the onkeyup property.

Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

setOnkeyup

public void setOnkeyup(String onkeyup)

Set the value of the onkeyup property.

public String getOnmousedown()

getOnmousedown

public String getOnmousedown()

Return the value of the onmousedown property.
Contents: Javascript code executed when a pointer button is pressed down over this element.

public void setOnmousedown(String onmousedown)

    onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

public String getOnmousemove()

    onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

    Return the value of the onmousemove property.

    Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

    onmousemove
setOnmousemove

public void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.

public String getOnmouseout()

onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.
public String getOnmouseover()

    onmouseover

Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.
    Contents: Javascript code executed when a pointer button is moved onto this element.

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

public String getOnmouseup()
getOnmouseup

public String getOnmouseup()

    Return the value of the onmouseup property.

    Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getOnselect()

    onselect

Javascript

getOnselect

public String getOnselect()

    Return the value of the onselect property.
Contents: Javascript code executed when text within this element is selected by the user.

```java
public void setOnselect(String onSelect)

    onSelect
```

setOnselect

```java
public void setOnselect(String onSelect)

    Set the value of the onSelect property.
```

```java
public boolean isReadonly()

    readonly
```

false true readonly="readonly"

isReadonly

```java
public boolean isReadonly()

    Return the value of the readonly property.

    Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".
```
public void setReadonly(boolean readonly)

readonly

setReadonly

Set the value of the readonly property.

public String getStyle()

style

CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

style
**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.

---

**public String getStyleClass()**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**public void setStyleClass(String styleClass)**

```java
public void setStyleClass(String styleClass)
```

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.
public String getTabindex()

    tabindex

    Tab 0 32767

getTabindex

public String getTabindex()

    Return the value of the tabindex property.

    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

public void setTabindex(String tabindex)

    tabindex

setTabindex

public void setTabindex(String tabindex)

    Set the value of the tabindex property.

public String getTitle()

    title
**getTitle**

```java
public String getTitle()
```

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

---

**public void setTitle(String title)**

title

---

**setTitle**

```java
public void setTitle(String title)
```

Set the value of the title property.

---

**public Object saveState(FacesContext _context)**

**saveState**

```java
public Object saveState(FacesContext _context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the
**StateHolder.saveState(javax.faces.context.FacesContext)** method on all those instances as well. **This method must not save the state of children and facets.** That is done via the **StateManager**

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**
- `saveState` in interface **StateHolder**

**Overrides:**
- `saveState` in class **UIInput**

---

**public void restoreState(FacesContext _context, Object _state)**

**restoreState**

**public void restoreState(FacesContext _context,**
- **Object _state)**

**Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UICOMPONENT with event handlers, validators, etc.) this method must call the **StateHolder.restoreState(javax.faces.context.FacesContext,**
- **java.lang.Object)** method on all those instances as well.
Specified by:  
restoreState in interface StateHolder

Overrides:  
restoreState in class UIInput
javax.faces.component.html  Class HtmlSelectOneListbox

java.lang.Object  
  ❏ javax.faces.component.UIComponent  
    ❏ javax.faces.component.UIComponentBase  
      ❏ javax.faces.component.UIOutput  
        ❏ javax.faces.component.UIInput  
          ❏ javax.faces.component.html.HtmlSelectOne

All Implemented Interfaces:
  EditableValueHolder, StateHolder, ValueHolder

public class HtmlSelectOneListbox
  extends UILSelectOne

Extends: UIComponent > UICOMPONENTBASE > UIOutput > UINput > UILSelectOne

HTML  select
  rendererType "javax.faces.Listbox"  setRendererType()

Represents a single-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.

By default, the rendererType property must be set to "javax.faces.Listbox". This value can be changed by calling the setRendererType() method.

Field Summary

| COMPONENT_TYPE |
The standard component type for this component.

**Fields inherited from class javax.faces.component.UISelectOne**
- COMPONENT_FAMILY, INVALID_MESSAGE_ID

**Fields inherited from class javax.faces.component.UlInput**
- CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID, UPDATE_MESSAGE_ID

**Fields inherited from class javax.faces.component.UIComponent**
- bindings

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>HtmlSelectOneListbox()</td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAccesskey()</strong></td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td><strong>getDir()</strong></td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td><strong>getDisabledClass()</strong></td>
<td>Return the value of the disabledClass property.</td>
</tr>
<tr>
<td><strong>getEnabledClass()</strong></td>
<td>Return the value of the enabledClass property.</td>
</tr>
<tr>
<td><strong>getLabel()</strong></td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td><strong>getLang()</strong></td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td><strong>getOnblur()</strong></td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td><strong>getOnchange()</strong></td>
<td>Return the value of the onchange property.</td>
</tr>
<tr>
<td>String</td>
<td>getonclick()</td>
</tr>
<tr>
<td>--------</td>
<td>--------------</td>
</tr>
<tr>
<td></td>
<td>Return the value of the onclick property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getondblclick()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the ondblclick property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonfocus()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onfocus property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonkeydown()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onkeydown property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonkeypress()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onkeypress property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonkeyup()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onkeyup property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonmousedown()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onmousedown property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonmousemove()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onmousemove property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonmouseout()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onmouseout property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonmouseover()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onmouseover property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonmouseup()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onmouseup property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getonselect()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the onselect property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>int</th>
<th>getSize()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the size property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getstyle()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the style property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getstyleclass()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the styleClass property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>gettabindex()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the tabindex property.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>gettitle()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>boolean isDisabled()</code></td>
<td>Return the value of the disabled property.</td>
</tr>
<tr>
<td><code>boolean isReadonly()</code></td>
<td>Return the value of the readonly property.</td>
</tr>
<tr>
<td><code>void restoreState(FacesContext _context, Object _state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>Object saveState(FacesContext _context)</code></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td><code>void setAccesskey(String accesskey)</code></td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td><code>void setDir(String dir)</code></td>
<td>Set the value of the dir property.</td>
</tr>
<tr>
<td><code>void setDisabled(boolean disabled)</code></td>
<td>Set the value of the disabled property.</td>
</tr>
<tr>
<td><code>void setDisabledClass(String disabledClass)</code></td>
<td>Set the value of the disabledClass property.</td>
</tr>
<tr>
<td><code>void setEnabledClass(String enabledClass)</code></td>
<td>Set the value of the enabledClass property.</td>
</tr>
<tr>
<td><code>void setLabel(String label)</code></td>
<td>Set the value of the label property.</td>
</tr>
<tr>
<td><code>void setLang(String lang)</code></td>
<td>Set the value of the lang property.</td>
</tr>
<tr>
<td><code>void setOnblur(String onblur)</code></td>
<td>Set the value of the onblur property.</td>
</tr>
<tr>
<td><code>void setOnchange(String onchange)</code></td>
<td>Set the value of the onchange property.</td>
</tr>
<tr>
<td><code>void setOnclick(String onclick)</code></td>
<td>Set the value of the onclick property.</td>
</tr>
<tr>
<td><code>void setOndblclick(String ondblclick)</code></td>
<td>Set the value of the ondblclick property.</td>
</tr>
<tr>
<td><code>void setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td><code>void setOnkeydown(String onkeydown)</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>setName(String name)</td>
<td>Set the value of the name property.</td>
</tr>
<tr>
<td>setOnkeydown(String onkeydown)</td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td>setOnkeypress(String onkeypress)</td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td>setOnkeyup(String onkeyup)</td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td>setOnmousedown(String onmousedown)</td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td>setOnmousemove(String onmousemove)</td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td>setOnmouseout(String onmouseout)</td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td>setOnmouseover(String onmouseover)</td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td>setOnmouseup(String onmouseup)</td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td>setOnselect(String onselect)</td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td>setReadonly(boolean readonly)</td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td>setSize(int size)</td>
<td>Set the value of the size property.</td>
</tr>
<tr>
<td>setStyle(String style)</td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td>setStyleClass(String styleClass)</td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td>setTabindex(String tabindex)</td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td>setTitle(String title)</td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.component.UISelectOne: getFamily, validateValue
Methods inherited from class `javax.faces.component.UIInput`
- `addValidator`, `addValueChangeListener`, `compareValues`, `decode`, `getConvertedValue`, `getConverterMessage`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValid`, `setValidator`, `setValidatorMessage`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`

Methods inherited from class `javax.faces.component.UIOutput`
- `getConverter`, `getLocalValue`, `getValue`, `setConverter`

Methods inherited from class `javax.faces.component.UIComponentBase`
- `addFacesListener`, `broadcast`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processRestoreState`, `processSaveState`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`
- `encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `javax.faces.component.ValueHolder`
- `getConverter`, `getLocalValue`, `getValue`, `setConverter`
Field Detail

COMPONENT_TYPE

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

See Also:

Constant Field Values

Constructor Detail

```java
public HtmlSelectOneListbox()
```

```java
HtmlSelectOneListbox
```

```java
public HtmlSelectOneListbox()
```

Method Detail

```java
public String getAccesskey()
```

```java
accesskey
```

getAccesskey

```java
public String getAccesskey()
```
Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

```java
public void setAccesskey(String accesskey)
```

```java
setAccesskey
```

Set the value of the accesskey property.

```java
public String getDir()
```

```java
getDir()
```

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).
public void setDir(String dir)

dir

setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isDisabled()

disabled

false true disabled="disabled"

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled
**setDisabled**

public void setDisabled(boolean disabled)

Set the value of the disabled property.

---

**public String getDisabledClass()**

disabledClass

CSS

**getDisabledClass**

public String getDisabledClass()

Return the value of the disabledClass property.

Contents: CSS style class to apply to the rendered label on disabled options.

---

**public void setDisabledClass(String disabledClass)**

disabledClass

**setDisabledClass**

public void setDisabledClass(String disabledClass)

Set the value of the disabledClass property.
public String getEnabledClass()

    enabledClass

    CSS

getEnabledClass

public String getEnabledClass()

    Return the value of the enabledClass property.
    Contents: CSS style class to apply to the rendered label on enabled options.

public void setEnabledClass(String enabledClass)

    enabledClass

setEnabledClass

public void setEnabledClass(String enabledClass)

    Set the value of the enabledClass property.

public String getLabel()

    label
**getLabel**

```java
public String getLabel()
```

Return the value of the `label` property.

Contents: A localized user presentable name for this component.

---

**public void setLabel(String label)**

```java
label
```

---

**setLabel**

```java
public void setLabel(String label)
```

Set the value of the `label` property.

---

**public String getLang()**

```java
lang
```

---

**getLang**

```java
public String getLang()
```

Return the value of the `lang` property.
Contents: Code describing the language used in the generated markup for this component.

```
public void setLang(String lang)

  lang

setLang
```

Set the value of the `lang` property.

```
public String getOnblur()

  onblur

Javascript
```

Return the value of the `onblur` property.

Contents: Javascript code executed when this element loses focus.

```
public void setOnblur(String onblur)

  onblur
```
setOnblur

public void setOnblur(String onblur)

    Set the value of the onblur property.

public String getOnchange()

    onchange

Javascript

getOnchange

public String getOnchange()

    Return the value of the onchange property.
    Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

public void setOnchange(String onchange)

    onchange

setOnchange

public void setOnchange(String onchange)

    Set the value of the onchange property.
public String getOnclick()

    onclick

Javascript

getOnclick

public String getOnclick()

    Return the value of the onclick property.

    Contents: Javascript code executed when a pointer button is clicked over this element.

public void setOnclick(String onclick)

    onclick

setOnclick

public void setOnclick(String onclick)

    Set the value of the onclick property.

public String getOndblclick()

    ondblclick

Javascript
getOndblclick

public String getOndblclick()

    Return the value of the ondblclick property.
    
    Contents: Javascript code executed when a pointer button is double clicked over this element.

public void setOndblclick(String ondblclick)

    ondblclick

setOndblclick

public void setOndblclick(String ondblclick)

    Set the value of the ondblclick property.

public String getOnfocus()

    onfocus

Javascript

getOnfocus

public String getOnfocus()

    Return the value of the onfocus property.
Contents: Javascript code executed when this element receives focus.

public void setOnfocus(String onfocus)

  onfocus

setOnfocus

public void setOnfocus(String onfocus)

  Set the value of the onfocus property.

public String getOnkeydown()

  onkeydown

Javascript

getOnkeydown

public String getOnkeydown()

  Return the value of the onkeydown property.

  Contents: Javascript code executed when a key is pressed down over this element.

public void setOnkeydown(String onkeydown)

  onkeydown
setOnkeydown

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

---

public String getOnkeypress()

onkeypress

Javascript

getOnkeypress

public String getOnkeypress()

Return the value of the onkeypress property.

Contents: Javascript code executed when a key is pressed and released over this element.

---

public void setOnkeypress(String onkeypress)

onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.
public String getOnkeyup()

    onkeyup

Javascript

getOnkeyup

public String getOnkeyup()

    Return the value of the onkeyup property.

    Contents: Javascript code executed when a key is released over this element.

public void setOnkeyup(String onkeyup)

    onkeyup

setOnkeyup

public void setOnkeyup(String onkeyup)

    Set the value of the onkeyup property.

public String getOnmousedown()

    onmousedown

Javascript
**getOnmousedown**

public String getOnmousedown()

    Return the value of the onmousedown property.

    Contents: Javascript code executed when a pointer button is pressed down over this element.

**public void setOnmousedown(String onmousedown)**

    onmousedown

**setOnmousedown**

public void setOnmousedown(String onmousedown)

    Set the value of the onmousedown property.

**public String getOnmousemove()**

    onmousemove

**Javascript**

**getOnmousemove**

public String getOnmousemove()

    Return the value of the onmousemove property.
Contents: Javascript code executed when a pointer button is moved within this element.

```java
public void setOnmousemove(String onmousemove)

  onmousemove
```

`setOnmousemove`

```java
public void setOnmousemove(String onmousemove)

  Set the value of the onmousemove property.
```

```java
public String getOnmouseout()

  onmouseout
```

`getOnmouseout`

```java
public String getOnmouseout()

  Return the value of the onmouseout property.

  Contents: Javascript code executed when a pointer button is moved away from this element.

  ```java
  public void setOnmouseout(String onmouseout)

    onmouseout
  ```
**setOnmouseout**

public void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.

**public String getOnmouseover()**

onmouseover

Javascript

**getOnmouseover**

public String getOnmouseover()

Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.

**public void setOnmouseover(String onmouseover)**

onmouseover

**setOnmouseover**

public void setOnmouseover(String onmouseover)

Set the value of the onmouseover property.
public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public  String  getOnmouseup()

    Return the value of the onmouseup property.

    Contents: Javascript code executed when a pointer button is released over this element.

public void setOnmouseup(String onmouseup)

    onmouseup

setOnmouseup

public void setOnmouseup(String onmouseup)

    Set the value of the onmouseup property.

public String getOnselect()

    onselect

Javascript
getOnselect

public String getOnselect()

Return the value of the `onselect` property.

Contents: Javascript code executed when text within this element is selected by the user.

setOnselect

public void setOnselect(String onselect)

onselect

isReadonly

public boolean isReadonly()

readonly

false true readonly="readonly"
Return the value of the `readonly` property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as `readonly="readonly"`.

```java
public void setReadonly(boolean readonly)
    readonly
```

`setReadonly`

```java
public void setReadonly(boolean readonly)
    Set the value of the `readonly` property.
```

```java
public int getSize()
    size
```

`getSize`

```java
public int getSize()
    Return the value of the `size` property.

    Contents: Number of available options to be shown at all times. If not specified, all available options are shown.
```
public void setSize(int size)

    size

setSize

public void setSize(int size)

    Set the value of the size property.

public String getStyle()

    style

CSS

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

public void setStyle(String style)

    style
**setStyle**

```java
public void setStyle(String style)
```

Set the value of the `style` property.

---

**getStyleClass**

```java
public String getStyleClass()
```

### styleClass

CSS "class"

---

**getStyleClass**

```java
public String getStyleClass()
```

Return the value of the `styleClass` property.

Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

### styleClass

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the `styleClass` property.
public String getTabindex()

    tabindex

Tab 0 32767

getTabindex

public String getTabindex()

    Return the value of the tabindex property.

    Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

public void setTabindex(String tabindex)

    tabindex

setTabindex

public void setTabindex(String tabindex)

    Set the value of the tabindex property.

public String getTitle()

    title
getTitle

```java
public String getTitle()
```

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.

---

public void setTitle(String title)

title

setTitle

```java
public void setTitle(String title)
```

Set the value of the title property.

---

public Object saveState(FacesContext _context)

saveState

```java
public Object saveState(FacesContext _context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the
**`StateHolder.saveState(javax.faces.context.FacesContext)`** method on all those instances as well. **This method must not save the state of children and facets.** That is done via the **StateManager**

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**

- `saveState` in interface **StateHolder**

**Overrides:**

- `saveState` in class **UIInput**

---

```java
public void restoreState(FacesContext _context, Object _state)
```

**restoreState**

```java
public void restoreState(FacesContext _context,
                        Object _state)
```

**Description copied from interface: ** **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a `UIComponent` with event handlers, validators, etc.) this method must call the **`StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)`** method on all those instances as well.
Specified by:

restoreState in interface StateHolder

Overrides:

restoreState in class UIInput
javax.faces.component.html  Class HtmlSelectOneMenu

java.lang.Object
  ▼ javax.faces.component.UIComponent
      ▼ javax.faces.component.UIComponentBase
          ▼ javax.faces.component.UIOutput
              ▼ javax.faces.component.UIInput
                  ▼ javax.faces.component.html.HtmlSelectOne
                      ▼ javax.faces.component.html.HtmlSelectOneMenu

All Implemented Interfaces:
   EditableValueHolder, StateHolder, ValueHolder

public class HtmlSelectOneMenu
extends UISelectOne

Represents a single-selection component that is rendered as an HTML select element, showing a single available option at a time.

By default, the rendererType property must be set to "javax.faces.Menu". This value can be changed by calling the setRendererType() method.

---

Field Summary

<table>
<thead>
<tr>
<th></th>
<th>COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td></td>
</tr>
</tbody>
</table>

The standard component type for this component.

Fields inherited from class javax.faces.component.UISelectOne
COMPONENT_FAMILY, INVALID_MESSAGE_ID

Fields inherited from class javax.faces.component.UIInput
CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID, UPDATE_MESSAGE_ID

Fields inherited from class javax.faces.component.UIComponent
bindings
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HtmlSelectOneMenu()</code></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAccesskey()</code></td>
<td>Return the value of the <code>accesskey</code> property.</td>
</tr>
<tr>
<td><code>getDir()</code></td>
<td>Return the value of the <code>dir</code> property.</td>
</tr>
<tr>
<td><code>getDisabledClass()</code></td>
<td>Return the value of the <code>disabledClass</code> property.</td>
</tr>
<tr>
<td><code>getEnabledClass()</code></td>
<td>Return the value of the <code>enabledClass</code> property.</td>
</tr>
<tr>
<td><code>getLabel()</code></td>
<td>Return the value of the <code>label</code> property.</td>
</tr>
<tr>
<td><code>getLang()</code></td>
<td>Return the value of the <code>lang</code> property.</td>
</tr>
<tr>
<td><code>getOnblur()</code></td>
<td>Return the value of the <code>onblur</code> property.</td>
</tr>
<tr>
<td><code>getOnchange()</code></td>
<td>Return the value of the <code>onchange</code> property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>Return the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>Return the value of the <code>onfocus</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>String getOnkeydown()</td>
<td>Return the value of the onkeydown property.</td>
</tr>
<tr>
<td>String getOnmousedown()</td>
<td>Return the value of the onmousedown property.</td>
</tr>
<tr>
<td>String getOnmousemove()</td>
<td>Return the value of the onmousemove property.</td>
</tr>
<tr>
<td>String getOnmouseout()</td>
<td>Return the value of the onmouseout property.</td>
</tr>
<tr>
<td>String getOnmouseover()</td>
<td>Return the value of the onmouseover property.</td>
</tr>
<tr>
<td>String getOnmouseup()</td>
<td>Return the value of the onmouseup property.</td>
</tr>
<tr>
<td>String getOnselect()</td>
<td>Return the value of the onselect property.</td>
</tr>
<tr>
<td>String getStyle()</td>
<td>Return the value of the style property.</td>
</tr>
<tr>
<td>String getStyleClass()</td>
<td>Return the value of the styleClass property.</td>
</tr>
<tr>
<td>String getTabIndex()</td>
<td>Return the value of the tabindex property.</td>
</tr>
<tr>
<td>String getTitle()</td>
<td>Return the value of the title property.</td>
</tr>
<tr>
<td>boolean isDisabled()</td>
<td>Return the value of the disabled property.</td>
</tr>
<tr>
<td>boolean isReadonly()</td>
<td>Return the value of the readonly property.</td>
</tr>
<tr>
<td>void restoreState(FacesContext _context, Object _state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object saveState(FacesContext _context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void setAccesskey(String accesskey)</td>
<td>Set the value of the accesskey property.</td>
</tr>
<tr>
<td>void setDir(String dir)</td>
<td>Set the value of the dir property.</td>
</tr>
</tbody>
</table>
void setDisabled(boolean disabled)
        Set the value of the disabled property.

void setDisabledClass(String disabledClass)
        Set the value of the disabledClass property.

void setEnabledClass(String enabledClass)
        Set the value of the enabledClass property.

void setLabel(String label)
        Set the value of the label property.

void setLang(String lang)
        Set the value of the lang property.

void setOnblur(String onblur)
        Set the value of the onblur property.

void setOnchange(String onchange)
        Set the value of the onchange property.

void setonclick(String onclick)
        Set the value of the onclick property.

void setOndblclick(String ondblclick)
        Set the value of the ondblclick property.

void setOnfocus(String onfocus)
        Set the value of the onfocus property.

void setOnkeydown(String onkeydown)
        Set the value of the onkeydown property.

void setOnkeypress(String onkeypress)
        Set the value of the onkeypress property.

void setOnkeyup(String onkeyup)
        Set the value of the onkeyup property.

void setOnmousedown(String onmousedown)
        Set the value of the onmousedown property.

void setOnmousemove(String onmousemove)
        Set the value of the onmousemove property.

void setOnmouseout(String onmouseout)
        Set the value of the onmouseout property.

void setOnmouseover(String onmouseover)
        Set the value of the onmouseover property.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>void setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>void setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>void setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
<tr>
<td><code>void setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>void setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>void setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UISelectOne`:
- `getFamily`, `validateValue`

Methods inherited from class `javax.faces.component.UlInput`:
- `addValidator`, `addValueChangeListener`, `compareValues`, `decode`, `getConvertedValue`, `getConverterMessage`, `getRequiredMessage`, `getSubmittedValue`, `getValidator`, `getValidatorMessage`, `getValidators`, `getValueChangeListener`, `getValueChangeListeners`, `isImmediate`, `isLocalValueSet`, `isRequired`, `isValid`, `processDecodes`, `processUpdates`, `processValidators`, `removeValidator`, `removeValueChangeListener`, `resetValue`, `setConverterMessage`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValid`, `setValidator`, `setValidatorMessage`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`

Methods inherited from class `javax.faces.component.UIOutput`:
- `getConverter`, `getLocalValue`, `getValue`, `setConverter`

Methods inherited from class `javax.faces.component.UIComponentBase`:
- `addFacesListener`, `broadcast`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `isImmediate`, `isValid`, `setImmediate`, `setLocalValueSet`, `setRequired`, `setRequiredMessage`, `setSubmittedValue`, `setValid`, `setValidator`, `setValidatorMessage`, `setValue`, `setValueChangeListener`, `updateModel`, `validate`
getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.faces.component.ValueHolder
getConverter, getLocalValue, getValue, setConverter

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

Constructor Detail

HtmlSelectOneMenu

public HtmlSelectOneMenu()
Method Detail

getAccesskey

public String getAccesskey()

Return the value of the accesskey property.

Contents: Access key that, when pressed, transfers focus to this element.

setAccesskey

public void setAccesskey(String accesskey)

Set the value of the accesskey property.

getDir

public String getDir()

Return the value of the dir property.

Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

setDir

public void setDir(String dir)

Set the value of the dir property.

isDisabled
public boolean isEnabled()

    Return the value of the disabled property.

    Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

---

setDisabled

public void setDisabled(boolean disabled)

    Set the value of the disabled property.

---

getDisabledClass

public String getDisabledClass()

    Return the value of the disabledClass property.

    Contents: CSS style class to apply to the rendered label on disabled options.

---

setDisabledClass

public void setDisabledClass(String disabledClass)

    Set the value of the disabledClass property.

---

getEnabledClass

public String getEnabledClass()

    Return the value of the enabledClass property.
Contents: CSS style class to apply to the rendered label on enabled options.

---

**setEnabledClass**

public void `setEnabledClass(String enabledClass)`

Set the value of the `enabledClass` property.

---

**getLabel**

public `String getLabel()`

Return the value of the `label` property.

Contents: A localized user presentable name for this component.

---

**setLabel**

public void `setLabel(String label)`

Set the value of the `label` property.

---

**getLang**

public `String getLang()`

Return the value of the `lang` property.

Contents: Code describing the language used in the generated markup for this component.

---

**setLang**

public void `setLang(String lang)`
Set the value of the `lang` property.

---

**getOnblur**

```java
public String getOnblur()
```

Return the value of the `onblur` property.

Contents: Javascript code executed when this element loses focus.

---

**setOnblur**

```java
public void setOnblur(String onblur)
```

Set the value of the `onblur` property.

---

**getOnchange**

```java
public String getOnchange()
```

Return the value of the `onchange` property.

Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

---

**setOnchange**

```java
public void setOnchange(String onchange)
```

Set the value of the `onchange` property.

---

**getOnclick**

```java
public String getOnclick()
```
Return the value of the `onclick` property.

Contents: Javascript code executed when a pointer button is clicked over this element.

---

**setOnclick**

```java
public void setOnclick(String onclick)
```

Set the value of the `onclick` property.

---

**getOndblclick**

```java
public String getOndblclick()
```

Return the value of the `ondblclick` property.

Contents: Javascript code executed when a pointer button is double clicked over this element.

---

**setOndblclick**

```java
public void setOndblclick(String ondblclick)
```

Set the value of the `ondblclick` property.

---

**getOnfocus**

```java
public String getOnfocus()
```

Return the value of the `onfocus` property.

Contents: Javascript code executed when this element receives focus.
**setOnfocus**

public void setOnfocus(String onfocus)

Set the value of the onfocus property.

---

**getOnkeydown**

public String getOnkeydown()

Return the value of the onkeydown property.

Contents: Javascript code executed when a key is pressed down over this element.

---

**setOnkeydown**

public void setOnkeydown(String onkeydown)

Set the value of the onkeydown property.

---

**getOnkeypress**

public String getOnkeypress()

Return the value of the onkeypress property.

Contents: Javascript code executed when a key is pressed and released over this element.

---

**setOnkeypress**

public void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.
**getOnkeyup**

```java
public String getOnkeyup()
```

Return the value of the onkeyup property.

Contents: Javascript code executed when a key is released over this element.

**setOnkeyup**

```java
public void setOnkeyup(String onkeyup)
```

Set the value of the onkeyup property.

**getOnmousedown**

```java
public String getOnmousedown()
```

Return the value of the onmousedown property.

Contents: Javascript code executed when a pointer button is pressed down over this element.

**setOnmousedown**

```java
public void setOnmousedown(String onmousedown)
```

Set the value of the onmousedown property.

**getOnmousemove**

```java
public String getOnmousemove()
```

Return the value of the onmousemove property.
Contents: Javascript code executed when a pointer button is moved within this element.

---

**setOnmousemove**

```java
public void setOnmousemove(String onmousemove)

Set the value of the onmousemove property.
```

---

**getOnmouseout**

```java
public String getOnmouseout()

Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.
```

---

**setOnmouseout**

```java
public void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.
```

---

**getOnmouseover**

```java
public String getOnmouseover()

Return the value of the onmouseover property.

Contents: Javascript code executed when a pointer button is moved onto this element.
```

---

**setOnmouseover**
public void setOnmouseover(String onmouseover)

Set the value of the onmouseover property.

---

getOnmouseup

public String getOnmouseup()

Return the value of the onmouseup property.

Contents: Javascript code executed when a pointer button is released over this element.

---

setOnmouseup

public void setOnmouseup(String onmouseup)

Set the value of the onmouseup property.

---

getOnselect

public String getOnselect()

Return the value of the onselect property.

Contents: Javascript code executed when text within this element is selected by the user.

---

setOnselect

public void setOnselect(String onselect)

Set the value of the onselect property.

---

isReadonly
public boolean isReadonly()

    Return the value of the readonly property.

    Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".

---

setReadonly

public void setReadonly(boolean readonly)

    Set the value of the readonly property.

---

getStyle

public String getStyle()

    Return the value of the style property.

    Contents: CSS style(s) to be applied when this component is rendered.

---

setStyle

public void setStyle(String style)

    Set the value of the style property.

---

getStyleClass

public String getStyleClass()

    Return the value of the styleClass property.
Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

---

**setStyleClass**

```java
public void setStyleClass(String styleClass)
```

Set the value of the styleClass property.

---

**getTabindex**

```java
public String getTabindex()
```

Return the value of the tabindex property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

---

**setTabindex**

```java
public void setTabindex(String tabindex)
```

Set the value of the tabindex property.

---

**getTitle**

```java
public String getTitle()
```

Return the value of the title property.

Contents: Advisory title information about markup elements generated for this component.
setTitle

public void setTitle(String title)

Set the value of the title property.

saveState

public Object saveState(FacesContext _context)

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**
(saveState in interface StateHolder)

**Overrides:**
(saveState in class UIInput)

restoreState

public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreStatejavax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
    restoreState in interface StateHolder

Overrides:
    restoreState in class UIInput
javax.faces.component.html  Class HtmlSelectOneRadio

java.lang.Object
  ▼ javax.faces.component.UIComponent
      ▼ javax.faces.component.UIComponentBase
          ▼ javax.faces.component.UIOutput
              ▼ javax.faces.component.UIInput
                  ▼ javax.faces.component.UISelectOne
                      ▼ javax.faces.component.html.HtmlSelectOneRadio

All Implemented Interfaces:
  EditableValueHolder, StateHolder, ValueHolder

public class HtmlSelectOneRadio
  extends UISelectOne

Extends: UIComponent > UIComponentBase > UIOutput > UIInput > UISelectOne

radio HTML input

renderertype "javax.faces.Radio" setRendererType()

Represents a single-selection component that is rendered as a set of HTML input elements of type radio.

By default, the renderertype property must be set to "javax.faces.Radio". This value can be changed by calling the setRendererType() method.

---

Field Summary

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
<th>The standard component type for this component.</th>
</tr>
</thead>
</table>
### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>HtmlSelectOneRadio()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getAccesskey()</td>
<td>Return the value of the accesskey property.</td>
</tr>
<tr>
<td>int getBorder()</td>
<td>Return the value of the border property.</td>
</tr>
<tr>
<td>String getDir()</td>
<td>Return the value of the dir property.</td>
</tr>
<tr>
<td>String getDisabledClass()</td>
<td>Return the value of the disabledClass property.</td>
</tr>
<tr>
<td>String getEnabledClass()</td>
<td>Return the value of the enabledClass property.</td>
</tr>
<tr>
<td>String getLabel()</td>
<td>Return the value of the label property.</td>
</tr>
<tr>
<td>String getLang()</td>
<td>Return the value of the lang property.</td>
</tr>
<tr>
<td>String getLayout()</td>
<td>Return the value of the layout property.</td>
</tr>
<tr>
<td>String getOnblur()</td>
<td>Return the value of the onblur property.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>getOnchange()</code></td>
<td>Return the value of the <code>onchange</code> property.</td>
</tr>
<tr>
<td><code>getOnclick()</code></td>
<td>Return the value of the <code>onclick</code> property.</td>
</tr>
<tr>
<td><code>getOndblclick()</code></td>
<td>Return the value of the <code>ondblclick</code> property.</td>
</tr>
<tr>
<td><code>getOnfocus()</code></td>
<td>Return the value of the <code>onfocus</code> property.</td>
</tr>
<tr>
<td><code>getOnkeydown()</code></td>
<td>Return the value of the <code>onkeydown</code> property.</td>
</tr>
<tr>
<td><code>getOnkeypress()</code></td>
<td>Return the value of the <code>onkeypress</code> property.</td>
</tr>
<tr>
<td><code>getOnkeyup()</code></td>
<td>Return the value of the <code>onkeyup</code> property.</td>
</tr>
<tr>
<td><code>getOnmousedown()</code></td>
<td>Return the value of the <code>onmousedown</code> property.</td>
</tr>
<tr>
<td><code>getOnmousemove()</code></td>
<td>Return the value of the <code>onmousemove</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseout()</code></td>
<td>Return the value of the <code>onmouseout</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseover()</code></td>
<td>Return the value of the <code>onmouseover</code> property.</td>
</tr>
<tr>
<td><code>getOnmouseup()</code></td>
<td>Return the value of the <code>onmouseup</code> property.</td>
</tr>
<tr>
<td><code>getOnselect()</code></td>
<td>Return the value of the <code>onselect</code> property.</td>
</tr>
<tr>
<td><code>getStyle()</code></td>
<td>Return the value of the <code>style</code> property.</td>
</tr>
<tr>
<td><code>getStyleClass()</code></td>
<td>Return the value of the <code>styleClass</code> property.</td>
</tr>
<tr>
<td><code>getTabindex()</code></td>
<td>Return the value of the <code>tabindex</code> property.</td>
</tr>
<tr>
<td><code>getTitle()</code></td>
<td>Return the value of the <code>title</code> property.</td>
</tr>
</tbody>
</table>
boolean isDisabled()
    Return the value of the disabled property.

boolean isReadonly()
    Return the value of the readonly property.

void restoreState(FacesContext _context, Object _state)
    Perform any processing required to restore the state from the entries in the state Object.

Object saveState(FacesContext _context)
    Gets the state of the instance as a Serializable Object.

void setAccesskey(String accesskey)
    Set the value of the accesskey property.

void setBorder(int border)
    Set the value of the border property.

void setDir(String dir)
    Set the value of the dir property.

void setDisabled(boolean disabled)
    Set the value of the disabled property.

void setDisabledClass(String disabledClass)
    Set the value of the disabledClass property.

void setEnabledClass(String enabledClass)
    Set the value of the enabledClass property.

void setLabel(String label)
    Set the value of the label property.

void setLang(String lang)
    Set the value of the lang property.

void setLayout(String layout)
    Set the value of the layout property.

void setOnblur(String onblur)
    Set the value of the onblur property.

void setOnchange(String onchange)
    Set the value of the onchange property.

void setOnClick(String onclick)
    Set the value of the onclick property.

void setOndblclick(String ondblclick)
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>setOnfocus(String onfocus)</code></td>
<td>Set the value of the onfocus property.</td>
</tr>
<tr>
<td><code>setOnkeydown(String onkeydown)</code></td>
<td>Set the value of the onkeydown property.</td>
</tr>
<tr>
<td><code>setOnkeypress(String onkeypress)</code></td>
<td>Set the value of the onkeypress property.</td>
</tr>
<tr>
<td><code>setOnkeyup(String onkeyup)</code></td>
<td>Set the value of the onkeyup property.</td>
</tr>
<tr>
<td><code>setOnmousedown(String onmousedown)</code></td>
<td>Set the value of the onmousedown property.</td>
</tr>
<tr>
<td><code>setOnmousemove(String onmousemove)</code></td>
<td>Set the value of the onmousemove property.</td>
</tr>
<tr>
<td><code>setOnmouseout(String onmouseout)</code></td>
<td>Set the value of the onmouseout property.</td>
</tr>
<tr>
<td><code>setOnmouseover(String onmouseover)</code></td>
<td>Set the value of the onmouseover property.</td>
</tr>
<tr>
<td><code>setOnmouseup(String onmouseup)</code></td>
<td>Set the value of the onmouseup property.</td>
</tr>
<tr>
<td><code>setOnselect(String onselect)</code></td>
<td>Set the value of the onselect property.</td>
</tr>
<tr>
<td><code>setReadonly(boolean readonly)</code></td>
<td>Set the value of the readonly property.</td>
</tr>
<tr>
<td><code>setStyle(String style)</code></td>
<td>Set the value of the style property.</td>
</tr>
<tr>
<td><code>setStyleClass(String styleClass)</code></td>
<td>Set the value of the styleClass property.</td>
</tr>
<tr>
<td><code>setTabindex(String tabindex)</code></td>
<td>Set the value of the tabindex property.</td>
</tr>
<tr>
<td><code>setTitle(String title)</code></td>
<td>Set the value of the title property.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.faces.component.UISelectOne`:
- `getFamily`
- `validateValue`
Methods inherited from class `javax.faces.component.UIInput`
- addValidator
- addValueChangeListener
- compareValues
- decode
- getConvertedValue
- getConverterMessage
- getRequiredMessage
- getSubmittedValue
- getValidator
- getValidatorMessage
- getValidators
- getValueChangeListener
- getValueChangeListeners
- isImmediate
- isLocalValueSet
- isRequired
- isValid
- processDecodes
- processUpdates
- processValidators
- removeValidator
- removeValueChangeListener
- resetValue
- setConverterMessage
- setImmediate
- setLocalValueSet
- setRequired
- setRequiredMessage
- setSubmittedValue
- setValid
- setValidator
- setValidatorMessage
- setValue
- setValueChangeListener
- updateModel
- validate

Methods inherited from class `javax.faces.component.UIOutput`
- getConverter
- getLocalValue
- getValue
- setConverter

Methods inherited from class `javax.faces.component.UIComponentBase`
- addFacesListener
- broadcast
- encodeBegin
- encodeChildren
- encodeEnd
- findComponent
- getAttributes
- getChildCount
- getChildren
- getClientId
- getFacesContext
- getFacesListeners
- getFacet
- getFacetCount
- getFacets
- getFacetsAndChildren
- getId
- getParent
- getRenderer
- getRendererType
- getRendersChildren
- getValueBinding
- invokeOnComponent
- isRendered
- isTransient
- processRestoreState
- processSaveState
- queueEvent
- removeFacesListener
- restoreAttachedState
- saveAttachedState
- setId
- setParent
- setRendered
- setRendererType
- setTransient
- setValueBinding

Methods inherited from class `javax.faces.component.UIComponent`
- encodeAll
- getContainerClientId
- getValueExpression
- setValueExpression

Methods inherited from class `java.lang.Object`
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
- wait

Methods inherited from interface `javax.faces.component.ValueHolder`
- getConverter
- getLocalValue
- getValue
- setConverter
**Field Detail**

**COMPONENT_TYPE**

```java
class HtmlSelectOneRadio {
    public static final String COMPONENT_TYPE = "html_select_one_radio";
}
```

The standard component type for this component.

**See Also:**
- [Constant Field Values](#)

**Constructor Detail**

```java
class HtmlSelectOneRadio {
    public HtmlSelectOneRadio() {
    }
}
```

**Method Detail**

```java
public String getAccesskey() {
    return accesskey;
}
```

getAccesskey
public String getAccesskey() {
    Return the value of the accesskey property.
    Contents: Access key that, when pressed, transfers focus to this element.
}

public void setAccesskey(String accesskey) {
    accesskey
}

setAccesskey

public void setAccesskey(String accesskey) {
    Set the value of the accesskey property.
}

public int getBorder() {
    border
}

getBorder

public int getBorder() {
    Return the value of the border property.
    Contents: Width (in pixels) of the border to be drawn around the table containing the options list.
}
public void setBorder(int border)

    border

setBorder

public void setBorder(int border)

    Set the value of the border property.

public String getDir()

    dir

"LTR"left-to-right "RTL"right-to-left

getDir

public String getDir()

    Return the value of the dir property.

    Contents: Direction indication for text that does not inherit directionality. Valid values are "LTR" (left-to-right) and "RTL" (right-to-left).

public void setDir(String dir)

    dir
setDir

public void setDir(String dir)

Set the value of the dir property.

public boolean isDisabled()

disabled

false true disabled="disabled"

isDisabled

public boolean isDisabled()

Return the value of the disabled property.

Contents: Flag indicating that this element must never receive focus or be included in a subsequent submit. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as disabled="disabled".

public void setDisabled(boolean disabled)

disabled

setDisabled

public void setDisabled(boolean disabled)

Set the value of the disabled property.
public String getDisabledClass()

disabledClass

CSS

getDisabledClass

public String getDisabledClass()

Return the value of the disabledClass property.

Contents: CSS style class to apply to the rendered label on disabled options.

public void setDisabledClass(String disabledClass)

disabledClass

setDisabledClass

public void setDisabledClass(String disabledClass)

Set the value of the disabledClass property.

public String getEnabledClass()

enabledClass

CSS
**getEnabledClass**

```java
public String getEnabledClass()
```

Return the value of the `enabledClass` property.

Contents: CSS style class to apply to the rendered label on enabled options.

**setEnabledClass**

```java
public void setEnabledClass(String enabledClass)
```

`enabledClass`

**setLabel**

```java
public String getLabel()
```

`label`

**getLabel**

```java
public String getLabel()
```

Return the value of the `label` property.
Contents: A localized user presentable name for this component.

public void setLabel(String label)

    label

setLabel

public void setLabel(String label)

    Set the value of the label property.

public String getLang()

    lang

gGetLang

public String getLang()

    Return the value of the lang property.

    Contents: Code describing the language used in the generated markup for this component.

public void setLang(String lang)

    lang
setLang

public void setLang(String lang)

Set the value of the lang property.

public String getLayout()

Return the value of the layout property.

Contents: Orientation of the options list to be created. Valid values are "pageDirection" (list is laid out vertically), or "lineDirection" (list is laid out horizontally). If not specified, the default value is "lineDirection".

public void setLayout(String layout)


public void setLayout(String layout)

Set the value of the layout property.

public String getOnblur()

onblur

Javascript

getOnblur

public String getOnblur()

Return the value of the onblur property.

Contents: Javascript code executed when this element loses focus.

public void setOnblur(String onblur)

onblur

setOnblur

public void setOnblur(String onblur)

Set the value of the onblur property.

public String getOnchange()

onchange
getOnchange

public String getOnchange()

    Return the value of the onchange property.

    Contents: Javascript code executed when this element loses focus and its value has been modified since gaining focus.

public void setOnchange(String onchange)

    onchange

setOnchange

public void setOnchange(String onchange)

    Set the value of the onchange property.

public String getOnclick()

    onclick

getOnclick

public String getOnclick()
Return the value of the onclick property.

Contents: Javascript code executed when a pointer button is clicked over this element.

```java
public void setOnclick(String onclick)
 onclick
```

setOnclick

```java
public void setOnclick(String onclick)
    Set the value of the onclick property.
```

```java
public String getOndblclick()
 ondblclick
```

Javascript

getOndblclick

```java
public String getOndblclick()
    Return the value of the ondblclick property.
    Contents: Javascript code executed when a pointer button is double clicked over this element.
```

```java
public void setOndblclick(String ondblclick)
```
**setOndblclick**

```java
public void setOndblclick(String ondblclick)
```

Set the value of the ondblclick property.

---

**public String getOnfocus()**

```java
onfocus
```

**Javascript**

---

**getOnfocus**

```java
public String getOnfocus()
```

Return the value of the onfocus property.

Contents: Javascript code executed when this element receives focus.

---

**public void setOnfocus(String onfocus)**

```java
onfocus
```

---

**setOnfocus**

```java
public void setOnfocus(String onfocus)
```
Set the value of the `onfocus` property.

```
public String getOnkeydown()
```

`onkeydown`

Javascript

**getOnkeydown**

```
public String getOnkeydown()
```

Return the value of the `onkeydown` property.

Contents: Javascript code executed when a key is pressed down over this element.

```
public void setOnkeydown(String onkeydown)
```

`onkeydown`

**setOnkeydown**

```
public void setOnkeydown(String onkeydown)
```

Set the value of the `onkeydown` property.

```
public String getOnkeypress()
```

`onkeypress`
Javascript

getOnkeypress

public String getOnkeypress()

Return the value of the onkeypress property.
Contents: Javascript code executed when a key is pressed and released over this element.

--------------------

public void setOnkeypress(String onkeypress)

onkeypress

setOnkeypress

public void setOnkeypress(String onkeypress)

Set the value of the onkeypress property.
--------------------

public String getOnkeyup()

onkeyup

Javascript

getOnkeyup

public String getOnkeyup()
Return the value of the `onkeyup` property.

Contents: Javascript code executed when a key is released over this element.

```java
public void setOnkeyup(String onkeyup)
  onkeyup
```

`setOnkeyup`

```java
public void setOnkeyup(String onkeyup)
  Set the value of the `onkeyup` property.
```

```java
public String getOnmousedown()
  onmousedown
```

`getOnmousedown`

```java
public String getOnmousedown()
  Return the value of the `onmousedown` property.
  Contents: Javascript code executed when a pointer button is pressed down over this element.
```

```java
public void setOnmousedown(String onmousedown)
```
onmousedown

setOnmousedown

public void setOnmousedown(String onmousedown)

Set the value of the onmousedown property.

public String getOnmousemove()

onmousemove

Javascript

getOnmousemove

public String getOnmousemove()

Return the value of the onmousemove property.

Contents: Javascript code executed when a pointer button is moved within this element.

public void setOnmousemove(String onmousemove)

onmousemove

setOnmousemove

public void setOnmousemove(String onmousemove)
Set the value of the onmousemove property.

public String getOnmouseout()

onmouseout

Javascript

getOnmouseout

public String getOnmouseout()

Return the value of the onmouseout property.

Contents: Javascript code executed when a pointer button is moved away from this element.

public void setOnmouseout(String onmouseout)

onmouseout

setOnmouseout

public void setOnmouseout(String onmouseout)

Set the value of the onmouseout property.

public String getOnmouseover()
Javascript

getOnmouseover

public String getOnmouseover()

    Return the value of the onmouseover property.

    Contents: Javascript code executed when a pointer button is moved onto this element.

-------------------------------------------------------------------------------------------------------------------

public void setOnmouseover(String onmouseover)

    onmouseover

setOnmouseover

public void setOnmouseover(String onmouseover)

    Set the value of the onmouseover property.

-------------------------------------------------------------------------------------------------------------------

public String getOnmouseup()

    onmouseup

Javascript

getOnmouseup

public String getOnmouseup()
Return the value of the `onmouseup` property.

Contents: Javascript code executed when a pointer button is released over this element.

```java
public void setOnmouseup(String onmouseup)

onmouseup
```

`setOnmouseup`

```java
public void setOnmouseup(String onmouseup)

Set the value of the `onmouseup` property.
```

`getOnselect`

```java
public String getOnselect()

onselect
```

`getOnselect`

```java
public String getOnselect()

Return the value of the `onselect` property.

Contents: Javascript code executed when text within this element is selected by the user.
```

```java
public void setOnselect(String onselect)
```
onselect

setOnselect

public void setOnselect(String onselect)

Set the value of the onselect property.

public boolean isReadonly()

readonly

false true readonly="readonly"

isReadonly

public boolean isReadonly()

Return the value of the readonly property.

Contents: Flag indicating that this component will prohibit changes by the user. The element may receive focus unless it has also been disabled. A value of false causes no attribute to be rendered, while a value of true causes the attribute to be rendered as readonly="readonly".

public void setReadonly(boolean readonly)

readonly
**setReadonly**

public void setReadonly(boolean readonly)

Set the value of the readonly property.

---

**public String getStyle()**

style

CSS

**getStyle**

public String getStyle()

Return the value of the style property.

Contents: CSS style(s) to be applied when this component is rendered.

---

**public void setStyle(String style)**

style

**setStyle**

public void setStyle(String style)

Set the value of the style property.
public String getStyleClass()

  styleClass

CSS "class"

getStyleClass

public String getStyleClass()

  Return the value of the styleClass property.

  Contents: Space-separated list of CSS style class(es) to be applied when this element is rendered. This value must be passed through as the "class" attribute on generated markup.

public void setStyleClass(String styleClass)

  styleClass

setStyleClass

public void setStyleClass(String styleClass)

  Set the value of the styleClass property.

public String getTabindex()

  tabindex

Tab 0 32767
**getTabindex**

public String getTabindex()

Return the value of the `tabindex` property.

Contents: Position of this element in the tabbing order for the current document. This value must be an integer between 0 and 32767.

---

**public void setTabindex(String tabindex)**

`tabindex`

**setTabindex**

public void setTabindex(String `tabindex`)

Set the value of the `tabindex` property.

---

**public String getTitle()**

`title`

**getTitle**

public String getTitle()

Return the value of the `title` property.
public void setTitle(String title)

title

setTitle

public void setTitle(String title)

Set the value of the title property.

public Object saveState(FacesContext _context)

saveState

public Object saveState(FacesContext _context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);
component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
    saveState in interface StateHolder

Overrides:
    saveState in class UIInput

public void restoreState(FacesContext _context, Object _state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
    restoreState in interface StateHolder

Overrides:
    restoreState in class UIInput
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface HTTPBinding

extends Binding

Implements: Binding

HTTPBinding XML/HTTP

since JAX-WS 2.0

The HTTPBinding interface is an abstraction for the XML/HTTP binding.

Since:

JAX-WS 2.0

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>HTTP_BINDING</th>
</tr>
</thead>
<tbody>
<tr>
<td>A constant representing the identity of the XML/HTTP binding.</td>
<td></td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from interface javax.xml.ws.Binding
getHandlerChain, setHandlerChain

Field Detail
HTTP_BINDING

static final String HTTP_BINDING

A constant representing the identity of the XML/HTTP binding.

See Also:
Constant Field Values
javax.xml.ws.http  Class HTTPException

java.lang.Object  
  | java.lang.Throwable  
  |   | java.lang.Exception  
  |   |   | java.lang.RuntimeException  
  |   |   | javax.xml.ws.WebServiceException  
  |   |   | javax.xml.ws.ProtocolException  
  |   |   | javax.xml.ws.http.HTTPException  

All Implemented Interfaces:
  Serializable

public class HTTPException
  extends ProtocolException

Extends: Throwable > Exception > RuntimeException > 
WebServiceException > ProtocolException

HTTPException  XML/HTTP

XML/HTTP  HTTP
  since  JAX-WS 2.0

The HTTPException exception represents a XML/HTTP fault.

Since there is no standard format for faults or exceptions in XML/HTTP messaging, only the HTTP status code is captured.

Since:  
  JAX-WS 2.0

See Also:  
  Serialized Form
Constructor Summary

**HttpException**

**(int statusCode)**

Constructor for the HTTPException

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>int getStatusCode()</strong></td>
<td>Gets the HTTP status code.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

Constructor Detail

```java
public HTTPException(int statusCode)
```

**HttpException**

**Parameters:**

- **statusCode**: int for the HTTP status code
public int getStatusCode()
HTTP

    return
HTTP

getStatusCode

public int getStatusCode()

    Gets the HTTP status code.

    Returns:
    HTTP status code
javax.servlet.jsp Interface HttpJspPage

All Superinterfaces:

JspPage, Servlet

public interface HttpJspPage

extends JspPage

Implements: JspPage

HttpJspPage HTTP JSP

JspPage _jspService Java

See also javax.servlet.jsp.JspPage

The HttpJspPage interface describes the interaction that a JSP Page Implementation Class must satisfy when using the HTTP protocol.

The behaviour is identical to that of the JspPage, except for the signature of the _jspService method, which is now expressible in the Java type system and included explicitly in the interface.

See Also:

JspPage

---

Method Summary

<table>
<thead>
<tr>
<th>void _jspService([HttpServletRequest] request, [HttpServletResponse] response)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The _jspService() method corresponds to the body of the JSP page.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.servlet.jsp JspPage
Methods inherited from interface javax.servlet.Servlet
destroy, getServletConfig, getServletInfo, init, service

Method Detail

public void _jspService(HttpServletRequest request, HttpServletResponse response) throws ServletException, java.io.IOException
_jspService()  JSP  JSP  JSP

_jjspService()  service()  JSP
JSP_Engine
    request  JSP
    response  JSP
    Throws  ServletException:  JSP
    Throws  java.io.IOException:

_jjspService

void _jspService(HttpServletRequest request,
        HttpServletResponse response)
    throws  ServletException,
            IOException

The _jspService() method corresponds to the body of the JSP page. This method is defined automatically by the JSP container and should never be defined by the JSP page author.

If a superclass is specified using the extends attribute, that superclass may choose to perform some actions in its service() method before or after calling the _jspService() method. See using the extends attribute in the JSP_Engine chapter of the JSP
specification.

**Parameters:**

- request - Provides client request information to the JSP.
- response - Assists the JSP in sending a response to the client.

**Throws:**

- `ServletException` - Thrown if an error occurred during the processing of the JSP and that the container should take appropriate action to clean up the request.
- `IOException` - Thrown if an error occurred while writing the response for this page.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD |
| FRAME | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
public abstract class HttpServlet

extends GenericServlet
implements Serializable

**Extends:** GenericServlet
**Implements:** java.io.Serializable

Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site. A subclass of HttpServlet must override at least one method, usually one of these:
- `doGet`, if the servlet supports HTTP GET requests
- `doPost`, for HTTP POST requests
- `doPut`, for HTTP PUT requests
- `doDelete`, for HTTP DELETE requests
- `init` and `destroy`, to manage resources that are held for the life of the servlet
- `getServletInfo`, which the servlet uses to provide information about itself

There's almost no reason to override the `service` method. `service` handles standard HTTP requests by dispatching them to the handler methods for each HTTP request type (the `doXXX` methods listed above).

Likewise, there's almost no reason to override the `doOptions` and `doTrace` methods.

Servlets typically run on multithreaded servers, so be aware that a servlet must handle concurrent requests and be careful to synchronize access to shared resources. Shared resources include in-memory data such as instance or class variables and external objects such as files, database connections, and network connections. See the [Java Tutorial on Multithreaded Programming](https://docs.oracle.com/javase/tutorial/essential/concurrency/) for more information on handling multiple threads in a Java program.

**Author:**
Various

**See Also:**
[Serialized Form](https://docs.oracle.com/javase/tutorial/essential/concurrency/)

---

### Constructor Summary

<table>
<thead>
<tr>
<th>HttpServlet()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does nothing, because this is an abstract class.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>protected void <code>doDelete</code>(HttpServletRequest req, HttpServletResponse resp)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Called by the server (via the <code>service</code> method) to allow a</td>
</tr>
</tbody>
</table>
A servlet to handle a DELETE request.

```java
protected void doGet(HttpServletRequest req, HttpServletResponse resp)
```
Called by the server (via the service method) to allow a servlet to handle a GET request.

```java
protected void doHead(HttpServletRequest req, HttpServletResponse resp)
```
Receives an HTTP HEAD request from the protected service method and handles the request.

```java
protected void doOptions(HttpServletRequest req, HttpServletResponse resp)
```
Called by the server (via the service method) to allow a servlet to handle a OPTIONS request.

```java
protected void doPost(HttpServletRequest req, HttpServletResponse resp)
```
Called by the server (via the service method) to allow a servlet to handle a POST request.

```java
protected void doPut(HttpServletRequest req, HttpServletResponse resp)
```
Called by the server (via the service method) to allow a servlet to handle a PUT request.

```java
protected void doTrace(HttpServletRequest req, HttpServletResponse resp)
```
Called by the server (via the service method) to allow a servlet to handle a TRACE request.

```java
protected long getLastModified(HttpServletRequest req)
```
Returns the time the HttpServletRequest object was last modified, in milliseconds since midnight January 1, 1970 GMT.

```java
protected void service(HttpServletRequest req, HttpServletResponse resp)
```
Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class.

```java
void service(ServletRequest req, ServletResponse res)
```
Dispatches client requests to the protected service method.

**Methods inherited from class javax.servlet.GenericServlet**

`destroy`, `getInitParameter`, `getInitParameterNames`, `getServletConfig`, `getServletContext`, `getServletInfo`, `getServletName`, `init`, `init`, `log`, `log`

**Methods inherited from class java.lang.Object**
Constructor Detail

public HttpServlet()

HttpServlet

public HttpServlet()

  Does nothing, because this is an abstract class.

Method Detail

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException

  service servlet GET

  GET HTTP HEAD HEAD GET

    PrintWriter PrintWriter

  servlet HTTP

  Content-Length

  javax.servlet.ServletResponse#setContentLength

  servlet

  HTTP 1.1 Transfer-Encoding Content-Length
GET HTTP

GET

doGet   HTTP "Bad Request"

req     servlet HttpServletRequest
resp    servlet HttpServletResponse

Throws  java.io.IOException: servlet  GET
Throws  ServletException: GET
See also  setContentType

doGet

protected void doGet(HttpServletRequest req,
                       HttpServletResponse resp)
              throws ServletException,
                        IOException

Called by the server (via the service method) to allow a servlet to handle a GET request.

Overriding this method to support a GET request also automatically supports an HTTP HEAD request. A HEAD request is a GET request that returns no body in the response, only the request header fields.

When overriding this method, read the request data, write the response headers, get the response's writer or output stream object, and finally, write the response data. It's best to include content type and encoding. When using a PrintWriter object to return the response, set the content type before accessing the PrintWriter object.

The servlet container must write the headers before committing the response, because in HTTP the headers must be sent before the response body.

Where possible, set the Content-Length header (with the
ServletResponse.setContentLength(int) method), to allow the servlet container to use a persistent connection to return its response to the client, improving performance. The content length is automatically set if the entire response fits inside the response buffer.

When using HTTP 1.1 chunked encoding (which means that the response has a Transfer-Encoding header), do not set the Content-Length header.

The GET method should be safe, that is, without any side effects for which users are held responsible. For example, most form queries have no side effects. If a client request is intended to change stored data, the request should use some other HTTP method.

The GET method should also be idempotent, meaning that it can be safely repeated. Sometimes making a method safe also makes it idempotent. For example, repeating queries is both safe and idempotent, but buying a product online or modifying data is neither safe nor idempotent.

If the request is incorrectly formatted, doGet returns an HTTP "Bad Request" message.

**Parameters:**

req - an HttpServletResponse object that contains the request the client has made of the servlet
resp - an HttpServletRequest object that contains the response the servlet sends to the client

**Throws:**

IOException - if an input or output error is detected when the servlet handles the GET request
ServletException - if the request for the GET could not be handled

**See Also:**

ServletResponse.setContentType(java.lang.String)

---

protected long getLastModified(HttpServletRequest req)
getLastModified

protected long getLastModified(HttpServletRequest req)

Returns the time the HttpServletRequest object was last modified, in milliseconds since midnight January 1, 1970 GMT. If the time is unknown, this method returns a negative number (the default).

Servlets that support HTTP GET requests and can quickly determine their last modification time should override this method. This makes browser and proxy caches work more effectively, reducing the load on server and network resources.

Parameters:
- req - the HttpServletRequest object that is sent to the servlet

Returns:
- a long integer specifying the time the HttpServletRequest object was last modified, in milliseconds since midnight, January 1, 1970 GMT, or -1 if the time is not known

protected void doHead(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException

service HTTP HEAD Content-Type
Content-Length HEAD HTTP HEAD
Content-Length
doHead  HTTP HEAD

HTTP HEAD  doHead  HTTP "Bad Request"

- req  servlet
- resp  servlet

Throws  java.io.IOException:

Throws  ServletException: HEAD

doHead

protected void doHead(HttpServletResponse req, HttpServletRequest resp)
throws ServletException, IOException

 Receives an HTTP HEAD request from the protected service method and handles the request. The client sends a HEAD request when it wants to see only the headers of a response, such as Content-Type or Content-Length. The HTTP HEAD method counts the output bytes in the response to set the Content-Length header accurately.

If you override this method, you can avoid computing the response body and just set the response headers directly to improve performance. Make sure that the doHead method you write is both safe and idempotent (that is, protects itself from being called multiple times for one HTTP HEAD request).

If the HTTP HEAD request is incorrectly formatted, doHead returns an HTTP "Bad Request" message.

Parameters:
- req - the request object that is passed to the servlet
- resp - the response object that the servlet uses to return the headers to the client

Throws:
- IOException - if an input or output error occurs
ServletException - if the request for the HEAD could not be handled

protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException

service servlet POST

Web

PrintWriter PrintWriter

servlet HTTP

Content-Length
javax.servlet.ServletResponse#setContentLength

servlet

HTTP 1.1 Transfer-Encoding Content-Length

POST

HTTP POST doPost HTTP "Bad Request"

req servlet HttpServletRequest
resp servlet HttpServletResponse

Throws java.io.IOException: servlet

Throws ServletException: POST

See also javax.servlet.ServletOutputStream, setContentLength

doPost

protected void doPost(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException

Called by the server (via the service method) to allow a servlet to handle a POST request. The HTTP POST method allows the client to send data of unlimited length to the Web server a single time and is useful when posting information such as credit card numbers.

When overriding this method, read the request data, write the response headers, get the response's writer or output stream object, and finally, write the response data. It's best to include content type and encoding. When using a PrintWriter object to return the response, set the content type before accessing the PrintWriter object.

The servlet container must write the headers before committing the response, because in HTTP the headers must be sent before the response body.

Where possible, set the Content-Length header (with the ServletResponse.setContentLength(int) method), to allow the servlet container to use a persistent connection to return its response to the client, improving performance. The content length is automatically set if the entire response fits inside the response buffer.

When using HTTP 1.1 chunked encoding (which means that the response has a Transfer-Encoding header), do not set the Content-Length header.

This method does not need to be either safe or idempotent. Operations requested through POST can have side effects for which the user can be held accountable, for example, updating stored data or buying items online.

If the HTTP POST request is incorrectly formatted, doPost returns an HTTP "Bad Request" message.

Parameters:

req - an HttpServletResponse object that contains the request the
client has made of the servlet
resp - an HttpServletResponse object that contains the response the servlet sends to the client

Throws:
IOException - if an input or output error is detected when the servlet handles the request
ServletException - if the request for the POST could not be handled

See Also:
ServletOutputStream,
ServletResponse.setContentType(java.lang.String)

protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException

service servlet PUT

Content-LengthContent-TypeContent-Transfer-EncodingContent-EncodingContent-BaseContent-LanguageContent-LocationContent-MD5  Content-Range (HTTP 501 - Not Implemented) HTTP 1.1 RFC 2616

doPut URL

HTTP PUT doGet HTTP "Bad Request"

req servlet HttpServletRequest
resp servlet HttpServletResponse

Throws java.io.IOException: servlet PUT
Throws ServletException: PUT

doPut
protected void doPost(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException

Called by the server (via the service method) to allow a servlet to handle a PUT request. The PUT operation allows a client to place a file on the server and is similar to sending a file by FTP.

When overriding this method, leave intact any content headers sent with the request (including Content-Length, Content-Type, Content-Transfer-Encoding, Content-Encoding, Content-Base, Content-Language, Content-Location, Content-MD5, and Content-Range). If your method cannot handle a content header, it must issue an error message (HTTP 501 - Not Implemented) and discard the request. For more information on HTTP 1.1, see RFC 2616.

This method does not need to be either safe or idempotent. Operations that doPost performs can have side effects for which the user can be held accountable. When using this method, it may be useful to save a copy of the affected URL in temporary storage.

If the HTTP PUT request is incorrectly formatted, doPost returns an HTTP "Bad Request" message.

Parameters:
- req - the HttpServletRequest object that contains the request the client made of the servlet
- resp - the HttpServletResponse object that contains the response the servlet returns to the client

Throws:
- IOException - if an input or output error occurs while the servlet is handling the PUT request
- ServletException - if the request for the PUT cannot be handled

protected void doDelete(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException
HTTP DELETE  
doDelete  HTTP "Bad Request"

protected void doDelete(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException

Called by the server (via the service method) to allow a servlet to handle a DELETE request. The DELETE operation allows a client to remove a document or Web page from the server.

This method does not need to be either safe or idempotent. Operations requested through DELETE can have side effects for which users can be held accountable. When using this method, it may be useful to save a copy of the affected URL in temporary storage.

If the HTTP DELETE request is incorrectly formatted, doDelete returns an HTTP "Bad Request" message.

Parameters:
- req - the HttpServletRequest object that contains the request the client made of the servlet
- resp - the HttpServletResponse object that contains the
response the servlet returns to the client

**Throws:**
- `IOException` - if an input or output error occurs while the servlet is handling the DELETE request
- `ServletException` - if the request for the DELETE cannot be handled

```java
protected void doOptions(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException {
    service servlet OPTIONS
    doGet
    Allow: GET, HEAD, TRACE, OPTIONS
}
```

**doOptions**

```java
protected void doOptions(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    bodyParser
    doPost
    Allow: GET, HEAD, TRACE, OPTIONS
}
```
There's no need to override this method unless the servlet implements new HTTP methods, beyond those implemented by HTTP 1.1.

**Parameters:**
- `req` - the `HttpServletRequest` object that contains the request the client made of the servlet
- `resp` - the `HttpServletResponse` object that contains the response the servlet returns to the client

**Throws:**
- `IOException` - if an input or output error occurs while the servlet is handling the OPTIONS request
- `ServletException` - if the request for the OPTIONS cannot be handled

---

```java
protected void doTrace(HttpServletRequest req, HttpServletResponse resp) throws ServletException, IOException {
    // Called by the server (via the service method) to allow a servlet to handle a TRACE request. A TRACE returns the headers sent with the TRACE request to the client, so that they can be used in debugging. There's no need to override this method.
}
```
Parameters:

req - the HttpServletRequest object that contains the request the client made of the servlet
resp - the HttpServletResponse object that contains the response the servlet returns to the client

Throws:

IOException - if an input or output error occurs while the servlet is handling the TRACE request
ServletException - if the request for the TRACE cannot be handled

protected void service(HttpServletRequest req, HttpServletResponse resp) throws ServletException, java.io.IOException
public service HTTP doXXX
javax.servlet.Servlet#service HTTP

req servlet HttpServletRequest
resp servlet HttpServletResponse

Throws java.io.IOException: HTTP
Throws ServletException: HTTP
See also

service

protected void service(HttpServletRequest req, HttpServletResponse resp)
throws ServletException, IOException

 Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class. This method is an HTTP-specific version of the Servlet.service(javax.servlet.ServletRequest, javax.servlet.ServletResponse) method. There's no need to override this method.
public void service(ServletRequest req, ServletResponse res) throws ServletException, IOException

service

Dispenses client requests to the protected service method. There's no need to override this method.

Specified by: service in interface Servlet
Specified by: service in class GenericServlet
Parameters:
req - the HttpServletRequest object that contains the request the client made of the servlet
res - the HttpServletResponse object that contains the response the servlet returns to the client

Throws:
IOException - if an input or output error occurs while the servlet is handling the HTTP request
ServletException - if the HTTP request cannot be handled

See Also:
Servlet.service(javax.servlet.ServletRequest, javax.servlet.ServletResponse)
Interface HttpServletRequest

All Superinterfaces:

- ServletRequest

All Known Implementing Classes:

- HttpServletRequestWrapper

```java
public interface HttpServletRequest extendsServletRequest

Implements: ServletRequest
Implemented by: HttpServletRequestWrapper
```

Extends the ServletRequest interface to provide request information for HTTP servlets.

The servlet container creates an HttpServletRequest object and passes it as an argument to the servlet's service methods (doGet, doPost, etc).

Author:
Various

---

### Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>BASIC_AUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>String identifier for Basic authentication.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>CLIENT_CERT_AUTH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
String identifier for Client Certificate authentication.

static String DIGEST_AUTH
  String identifier for Digest authentication.

static String FORM_AUTH
  String identifier for Form authentication.

## Method Summary

**getAuthType()**

Returns the name of the authentication scheme used to protect the servlet.

**getContextPath()**

Returns the portion of the request URI that indicates the context of the request.

**getCookies()**

Returns an array containing all of the Cookie objects the client sent with this request.

**getDateHeader(String name)**

Returns the value of the specified request header as a long value that represents a Date object.

**getHeader(String name)**

Returns the value of the specified request header as a String.

**getHeaderNames()**

Returns an enumeration of all the header names this request contains.

**getHeaders(String name)**

Returns all the values of the specified request header as an Enumeration of String objects.

**getIntHeader(String name)**

Returns the value of the specified request header as an int.

**getMethod()**

Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getPathInfo()</code></td>
<td>Returns any extra path information associated with the URL the client sent when it made this request.</td>
</tr>
<tr>
<td><code>getPathTranslated()</code></td>
<td>Returns any extra path information after the servlet name but before the query string, and translates it to a real path.</td>
</tr>
<tr>
<td><code>getQueryString()</code></td>
<td>Returns the query string that is contained in the request URL after the path.</td>
</tr>
<tr>
<td><code>getRemoteUser()</code></td>
<td>Returns the login of the user making this request, if the user has been authenticated, or <code>null</code> if the user has not been authenticated.</td>
</tr>
<tr>
<td><code>getRequestedSessionId()</code></td>
<td>Returns the session ID specified by the client.</td>
</tr>
<tr>
<td><code>getRequestURI()</code></td>
<td>Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.</td>
</tr>
<tr>
<td><code>getRequestURL()</code></td>
<td>Reconstructs the URL the client used to make the request.</td>
</tr>
<tr>
<td><code>getServletPath()</code></td>
<td>Returns the part of this request's URL that calls the servlet.</td>
</tr>
<tr>
<td><code>getSession()</code></td>
<td>Returns the current session associated with this request, or if the request does not have a session, creates one.</td>
</tr>
<tr>
<td><code>getSession(boolean create)</code></td>
<td>Returns the current <code>HttpSession</code> associated with this request or, if there is no current session and <code>create</code> is true, returns a new session.</td>
</tr>
<tr>
<td><code>getUserPrincipal()</code></td>
<td>Returns a <code>java.security.Principal</code> object containing</td>
</tr>
</tbody>
</table>
the name of the current authenticated user.

| boolean isRequestedSessionIdFromCookie() | Checks whether the requested session ID came in as a cookie. |
| boolean isRequestedSessionIdFromUrl()   | Deprecated. As of Version 2.1 of the Java Servlet API, use isRequestedSessionIdFromURL() instead. |
| boolean isRequestedSessionIdFromURL()   | Checks whether the requested session ID came in as part of the request URL. |
| boolean isRequestedSessionIdValid()     | Checks whether the requested session ID is still valid. |
| boolean isUserInRole(String role)       | Returns a boolean indicating whether the authenticated user is included in the specified logical "role". |

Methods inherited from interface javax.servlet.ServletRequest
getAttribute, getAttributeNames, getCharacterEncoding, getContentLength, getContentType, getInputStream, getLocalAddr, getLocale, getLocales, getLocalName, getLocalPort, getParameter, getParameterMap, getParameterNames, getParameterValues, getProtocol, getReader, getRealPath, getRemoteAddr, getRemoteHost, getRemotePort, getRequestDispatcher, getScheme, getServerName, getServerPort, isSecure, removeAttribute, setAttribute, setCharacterEncoding

Field Detail

BASIC_AUTH

static final String BASIC_AUTH

String identifier for Basic authentication. Value "BASIC"

See Also:
Constant Field Values

FORM_AUTH

static final String FORM_AUTH

String identifier for Form authentication. Value "FORM"

See Also:
Constant Field Values

CLIENT_CERT_AUTH

static final String CLIENT_CERT_AUTH

String identifier for Client Certificate authentication. Value "CLIENT_CERT"

See Also:
Constant Field Values

DIGEST_AUTH

static final String DIGEST_AUTH

String identifier for Digest authentication. Value "DIGEST"

See Also:
Constant Field Values
public String getAuthType()

    servlet servlet basicform client certificate
digest servlet

    CGI AUTH_TYPE
    return BASIC_AUTHFORM_AUTHCLIENT_CERT_AUTH
    DIGEST_AUTH ==

getAuthType

String getAuthType()

    Returns the name of the authentication scheme used to protect the
servlet. All servlet containers support basic, form and client
certificate authentication, and may additionally support digest
authentication. If the servlet is not authenticated null is returned.

    Same as the value of the CGI variable AUTH_TYPE.

    Returns:
    one of the static members BASIC_AUTH, FORM_AUTH,
CLIENT_CERT_AUTH, DIGEST_AUTH (suitable for ==
comparison) or the container-specific string indicating the
authentication scheme, or null if the request was not
authenticated.

public Cookie[] getCookies()

    Cookie cookie

    return null
getCookies

Cookie[] getCookies()

Returns an array containing all of the Cookie objects the client sent with this request. This method returns null if no cookies were sent.

Returns:
an array of all the Cookies included with this request, or null if the request has no cookies

public long getDateHeader(String name)

Date  long  If-Modified-Since

1970 1 1

-1 IllegalArgumentException

name String
return long 1970 1 1 -1

Throws IllegalArgumentException:

getDateHeader

long getDateHeader(String name)

Returns the value of the specified request header as a long value that represents a Date object. Use this method with headers that contain dates, such as If-Modified-Since.

The date is returned as the number of milliseconds since January 1, 1970 GMT. The header name is case insensitive.

If the request did not have a header of the specified name, this method returns -1. If the header can't be converted to a date, the method throws an IllegalArgumentException.
Parameters:
   name - a String specifying the name of the header

Returns:
   a long value representing the date specified in the header expressed as the number of milliseconds since January 1, 1970 GMT, or -1 if the named header was not included with the request

Throws:
   IllegalArgumentException - If the header value can't be converted to a date

public String getHeader(String name)

Parameters:
   name - a String specifying the header name

Returns:
   a String containing the value of the requested header, or null if the request does not have a header of that name

getHeader

Returns the value of the specified request header as a String. If the request did not include a header of the specified name, this method returns null. If there are multiple headers with the same name, this method returns the first head in the request. The header name is case insensitive. You can use this method with any request header.

public java.utilEnumeration\<E\> getHeaders(String name)
Accept-Language

Enumeration

name String
return Enumeration null

getHeaders

Enumeration getHeaders(String name)

Returns all the values of the specified request header as an Enumeration of String objects.

Some headers, such as Accept-Language can be sent by clients as several headers each with a different value rather than sending the header as a comma separated list.

If the request did not include any headers of the specified name, this method returns an empty Enumeration. The header name is case insensitive. You can use this method with any request header.

Parameters:
name - a String specifying the header name

Returns:
an Enumeration containing the values of the requested header. If the request does not have any headers of that name return an empty enumeration. If the container does not allow access to header information, return null

public java.util Enumeration<E> getHeaderNames()

servlet servlet null
return servlet servlet
getHeaderNames

`Enumeration getHeaderNames()`

Returns an enumeration of all the header names this request contains. If the request has no headers, this method returns an empty enumeration.

Some servlet containers do not allow servlets to access headers using this method, in which case this method returns `null`.

**Returns:**

an enumeration of all the header names sent with this request; if the request has no headers, an empty enumeration; if the servlet container does not allow servlets to use this method, `null`

---

`public int getIntHeader(String name)`

```
int -1
NumberFormatException
```

```
name String
return -1
Throws NumberFormatException: int
```

getIntHeader

`int getIntHeader(String name)`

Returns the value of the specified request header as an `int`. If the request does not have a header of the specified name, this method returns `-1`. If the header cannot be converted to an integer, this method throws a `NumberFormatException`.
The header name is case insensitive.

Parameters:
- name - a String specifying the name of a request header

Returns:
- an integer expressing the value of the request header or -1 if the request doesn't have a header of this name

Throws:
- NumberFormatException - If the header value can't be converted to an int

---

```java
public String getMethod()

HTTP GET POST PUT CGI
REQUEST_METHOD

return String
```

getMethod

```java
String getMethod()

Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT. Same as the value of the CGI variable REQUEST_METHOD.

Returns:
- a String specifying the name of the method with which this request was made
```

---

```java
public String getPathInfo()

URL servlet "/"

null

CGI PATH_INFO

return
```
**getPathInfo**

```java
String getPathInfo()
```

Returns any extra path information associated with the URL the client sent when it made this request. The extra path information follows the servlet path but precedes the query string and will start with a "/" character.

This method returns `null` if there was no extra path information.

Same as the value of the CGI variable PATH_INFO.

**Returns:**
- a `String`, decoded by the web container, specifying extra path information that comes after the servlet path but before the query string in the request URL; or `null` if the URL does not have any extra path information.

---

**public String getPathTranslated()**

```java
servlet CGI PATH_TRANSLATED
```

**getPathTranslated**

```java
String getPathTranslated()
```

Returns any extra path information after the servlet name but before the query string, and translates it to a real path. Same as the value
of the CGI variable PATH_TRANSLATED.

If the URL does not have any extra path information, this method returns null or the servlet container cannot translate the virtual path to a real path for any reason (such as when the web application is executed from an archive). The web container does not decode this string.

Returns:
- a String specifying the real path, or null if the URL does not have any extra path information

```java
public String getContextPath()
```

URI URI "/" "/" servlet

```java
return URI String
```

See also `getContextPath()`

`getContextPath()`

```java
String getContextPath()
```

Returns the portion of the request URI that indicates the context of the request. The context path always comes first in a request URI. The path starts with a "/" character but does not end with a "/" character. For servlets in the default (root) context, this method returns "/". The container does not decode this string.

It is possible that a servlet container may match a context by more than one context path. In such cases this method will return the actual context path used by the request and it may differ from the path returned by the `ServletContext.getContextPath()` method. The
context path returned by `ServletContext.getContextPath()` should be considered as the prime or preferred context path of the application.

**Returns:**
a String specifying the portion of the request URI that indicates the context of the request

**See Also:**
`ServletContext.getContextPath()`

```
public String getQueryString()
URL URL

    return String URL null

getQueryString

String getQueryString()

Returns the query string that is contained in the request URL after the path. This method returns null if the URL does not have a query string. Same as the value of the CGI variable QUERY_STRING.

**Returns:**
a String containing the query string or null if the URL contains no query string. The value is not decoded by the container.

```

```
public String getRemoteUser()

    return String null

getRemoteUser

String getRemoteUser()
```
Returns the login of the user making this request, if the user has been authenticated, or null if the user has not been authenticated. Whether the user name is sent with each subsequent request depends on the browser and type of authentication. Same as the value of the CGI variable REMOTE_USER.

**Returns:**

- a String specifying the login of the user making this request, or null if the user login is not known

---

```java
public boolean isUserInRole(String role)
```

**Parameters:**

- role - a String specifying the name of the role

**Returns:**

- a boolean indicating whether the authenticated user is included in the specified logical "role". Roles and role membership can be defined using deployment descriptors. If the user has not been authenticated, the method returns false.

---

```java
public java.security.Principal getUserPrincipal()
```

**Returns:**

- java.security.Principal
**getUserPrincipal**

```java
Principal getUserPrincipal()
```

Returns a `java.security.Principal` object containing the name of the current authenticated user. If the user has not been authenticated, the method returns `null`.

**Returns:**

- a `java.security.Principal` containing the name of the user making this request; `null` if the user has not been authenticated

---

**public String getRequestedSessionId()**

```java
String getRequestedSessionId()
```

Returns the session ID specified by the client. This may not be the same as the ID of the current valid session for this request. If the client did not specify a session ID, this method returns `null`.

**Returns:**

- a `String` specifying the session ID, or `null` if the request did not specify a session ID

**See Also:**

- `isRequestedSessionIdValid()`

---

**public String getRequestURI()**

```java
URL getRequestURI()
```

**POST /some/path.html HTTP/1.1 /some/path.html**
GET http://foo.bar/a.html HTTP/1.0 /a.html
HEAD /xyz?a=b HTTP/1.1 /xyz

**URL**

```java
HttpUtils#getRequestURL
```

- `return` `URL`
- `String`

**See also**

`getRequestURL`

---

**getRequestURI**

```java
String getRequestURI()
```

Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request. The web container does not decode this String. For example:

<table>
<thead>
<tr>
<th>First line of HTTP request</th>
<th>Returned Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>POST /some/path.html HTTP/1.1</td>
<td>/some/path.html</td>
</tr>
<tr>
<td>GET <a href="http://foo.bar/a.html">http://foo.bar/a.html</a> HTTP/1.0</td>
<td>/a.html</td>
</tr>
<tr>
<td>HEAD /xyz?a=b HTTP/1.1</td>
<td>/xyz</td>
</tr>
</tbody>
</table>

To reconstruct an URL with a scheme and host, use `HttpUtils.getRequestURL(javax.servlet.http.HttpServletRequest)`.

**Returns:**

- a `String` containing the part of the URL from the protocol name up to the query string

**See Also:**

`HttpUtils.getRequestURL(javax.servlet.http.HttpServletRequest)`

---

**public StringBuffer getRequestURL()**

```java
URL URL
```

```java
javas.servlet.RequestDispatcher#forward URL
```
getRequestURL

StringBuffer getRequestURL()

Reconstructs the URL the client used to make the request. The returned URL contains a protocol, server name, port number, and server path, but it does not include query string parameters.

If this request has been forwarded using RequestDispatcher.forward(javax.servlet.ServletRequest, javax.servlet.ServletResponse), the server path in the reconstructed URL must reflect the path used to obtain the RequestDispatcher, and not the server path specified by the client.

Because this method returns a StringBuffer, not a string, you can modify the URL easily, for example, to append query parameters.

This method is useful for creating redirect messages and for reporting errors.

Returns:

- a StringBuffer object containing the reconstructed URL
returns  

servlet  

String URL  servlet  "/*"  

getServletPath

String getServletPath()

Returns the part of this request's URL that calls the servlet. This path starts with a "/" character and includes either the servlet name or a path to the servlet, but does not include any extra path information or a query string. Same as the value of the CGI variable SCRIPT_NAME.

This method will return an empty string ("") if the servlet used to process this request was matched using the "/*" pattern.

Returns:

a String containing the name or path of the servlet being called, as specified in the request URL, decoded, or an empty string if the servlet used to process the request is matched using the "/*" pattern.


public HttpSession getSession(boolean create)

HttpSession create true

create false HttpSession null

cookie IllegalStateException

create true false null

return HttpSession create false null

See also HttpSession

getSession

getSession
HttpSession getSession(boolean create)

Returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

If create is false and the request has no valid HttpSession, this method returns null.

To make sure the session is properly maintained, you must call this method before the response is committed. If the container is using cookies to maintain session integrity and is asked to create a new session when the response is committed, an IllegalStateException is thrown.

Parameters:
create - true to create a new session for this request if necessary; false to return null if there's no current session

Returns:
the HttpSession associated with this request or null if create is false and the request has no valid session

See Also:
getSession()
public boolean isRequestedSessionIdValid() 

ID

false
return id
See also 
getRequestedSessionId, getSession, javax.servlet.http.HttpSessionContext

isRequestedSessionIdValid

boolean isRequestedSessionIdValid() 

Checks whether the requested session ID is still valid. 

If the client did not specify any session ID, this method returns false.

Returns: 
true if this request has an id for a valid session in the current session context; false otherwise
See Also: 
getRequestedSessionId(), getSession(boolean), HttpSessionContext

public boolean isRequestedSessionIdFromCookie() 

ID  cookie
return ID cookie
See also 
getSession

isRequestedSessionIdFromCookie
boolean isRequestedSessionIdFromCookie()

Checks whether the requested session ID came in as a cookie.

**Returns:**
true if the session ID came in as a cookie; otherwise, false

**See Also:**
getSession(boolean)

---

public boolean isRequestedSessionIdFromURL()

**ID** URL

return ID URL true false

See also getSession

---

isRequestedSessionIdFromURL

boolean isRequestedSessionIdFromURL()

Checks whether the requested session ID came in as part of the request URL.

**Returns:**
true if the session ID came in as part of a URL; otherwise, false

**See Also:**
getSession(boolean)

---

public boolean isRequestedSessionIdFromUrl()

**deprecated**

Java Servlet API 2.1

#isRequestedSessionIdFromURL

---

isRequestedSessionIdFromUrl

boolean isRequestedSessionIdFromUrl()

**Deprecated.** As of Version 2.1 of the Java Servlet API, use
isRequestedSessionIdFromURL() instead.
javax.servlet.http

Class HttpServletRequestWrapper

java.lang.Object
  ↓ javax.servlet.ServletRequestWrapper
     ↓ javax.servlet.http.HttpServletRequestWrapper

All Implemented Interfaces:
HttpServletRequest, ServletRequest

public class HttpServletRequestWrapper

extends ServletRequestWrapper
implements HttpServletRequest

Extends: ServletRequestWrapper
Implements: HttpServletRequest

HttpServletRequest Servlet Wrapper Decorator

since v 2.3
See also javax.servlet.http.HttpServletRequest

Provides a convenient implementation of the HttpServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet. This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped request object.

Since: v 2.3
See Also: HttpServletRequest
Fields inherited from interface javax.servlet.http.HttpServletRequest

BASIC_AUTH, CLIENT_CERT_AUTH, DIGEST_AUTH, FORM_AUTH

Constructor Summary

HttpServletRequestWrapper(HttpServletRequest request)
Constructs a request object wrapping the given request.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getAuthType()</td>
<td>The default behavior of this method is to return getAuthType() on the wrapped request object.</td>
</tr>
<tr>
<td>String getAuthType()</td>
<td>The default behavior of this method is to return getAuthType() on the wrapped request object.</td>
</tr>
<tr>
<td>String getCookie[]</td>
<td>The default behavior of this method is to return getCookie() on the wrapped request object.</td>
</tr>
<tr>
<td>Cookie[] getCookie[]</td>
<td>The default behavior of this method is to return getCookie() on the wrapped request object.</td>
</tr>
<tr>
<td>String getDateHeader(String name)</td>
<td>The default behavior of this method is to return getDateHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>String getDateHeader(String name)</td>
<td>The default behavior of this method is to return getDateHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>String getHeader(String name)</td>
<td>The default behavior of this method is to return getHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>String getHeader(String name)</td>
<td>The default behavior of this method is to return getHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration getHeaderNames()</td>
<td>The default behavior of this method is to return getHeaderNames() on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration getHeaderNames()</td>
<td>The default behavior of this method is to return getHeaderNames() on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration getHeaders(String name)</td>
<td>The default behavior of this method is to return getHeaders(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration getHeaders(String name)</td>
<td>The default behavior of this method is to return getHeaders(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>int getIntHeader(String name)</td>
<td>The default behavior of this method is to return getIntHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>int getIntHeader(String name)</td>
<td>The default behavior of this method is to return getIntHeader(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>String getMethod()</td>
<td>The default behavior of this method is to return getMethod() on the wrapped request object.</td>
</tr>
<tr>
<td>Method Type</td>
<td>Method Name</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>String</td>
<td>getMethod()</td>
</tr>
<tr>
<td>String</td>
<td>getPathInfo()</td>
</tr>
<tr>
<td>String</td>
<td>getPathTranslated()</td>
</tr>
<tr>
<td>String</td>
<td>getQueryString()</td>
</tr>
<tr>
<td>String</td>
<td>getRemoteUser()</td>
</tr>
<tr>
<td>String</td>
<td>getRequestedSessionId()</td>
</tr>
<tr>
<td>String</td>
<td>getRequestURI()</td>
</tr>
<tr>
<td>StringBuffer</td>
<td>getRequestURL()</td>
</tr>
<tr>
<td>String</td>
<td>getServletPath()</td>
</tr>
<tr>
<td>HttpSession</td>
<td>getSession()</td>
</tr>
<tr>
<td>HttpSession</td>
<td>getSession(boolean create)</td>
</tr>
<tr>
<td>Principal</td>
<td>getUserPrincipal()</td>
</tr>
<tr>
<td>boolean isRequestedSessionIdFromCookie()</td>
<td></td>
</tr>
<tr>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>The default behavior of this method is to return</td>
<td></td>
</tr>
<tr>
<td>isRequestedSessionIdFromCookie() on the wrapped request object.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean isRequestedSessionIdFromUrl()</th>
</tr>
</thead>
<tbody>
<tr>
<td>The default behavior of this method is to return</td>
</tr>
<tr>
<td>isRequestedSessionIdFromUrl() on the wrapped request object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean isRequestedSessionIdFromURL()</th>
</tr>
</thead>
<tbody>
<tr>
<td>The default behavior of this method is to return</td>
</tr>
<tr>
<td>isRequestedSessionIdFromURL() on the wrapped request object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean isRequestedSessionIdValid()</th>
</tr>
</thead>
<tbody>
<tr>
<td>The default behavior of this method is to return</td>
</tr>
<tr>
<td>isRequestedSessionIdValid() on the wrapped request object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean isUserInRole(String role)</th>
</tr>
</thead>
<tbody>
<tr>
<td>The default behavior of this method is to return</td>
</tr>
<tr>
<td>isUserInRole(String role) on the wrapped request object.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.servlet.ServletRequestWrapper`:

- `getAttribute`, `getAttributeNames`, `getCharacterEncoding`, `getContentType`, `getInputStream`, `getLocalAddr`, `getLocale`, `getLocales`, `getLocalName`, `getLocalPort`, `getParameter`, `getParameterMap`, `getParameterNames`, `getParameterValues`, `getProtocol`, `getReader`, `getRealPath`, `getRemoteAddr`, `getRemoteHost`, `getRemotePort`, `getRequestDispatcher`, `getScheme`, `getServerName`, `getServerPort`, `isSecure`, `removeAttribute`, `setAttribute`, `setCharacterEncoding`, `setRequest`.

Methods inherited from class `java.lang.Object`:

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`.

Methods inherited from interface `javax.servlet.ServletRequest`:

- `getAttribute`, `getAttributeNames`, `getCharacterEncoding`, `getContentType`, `getInputStream`, `getLocalAddr`, `getLocale`, `getLocales`, `getLocalName`, `getLocalPort`, `getParameter`, `getParameterMap`, `getParameterNames`, `getParameterValues`, `getProtocol`, `getReader`, `getRealPath`, `getRemoteAddr`, `getRemoteHost`. 
Constructor Detail

public HttpServletRequestWrapper(HttpServletRequest request)

Throws IllegalArgumentException: request null

HttpServletRequestWrapper

public HttpServletRequestWrapper(HttpServletRequest request)

Constructs a request object wrapping the given request.

Throws:
IllegalArgumentException - if the request is null

Method Detail

public String getAuthType()
getAuthType()

getAuthType

public String getAuthType()

The default behavior of this method is to return getAuthType() on the wrapped request object.

Specified by:
getAuthType in interface HttpServletRequest

Returns:
one of the static members BASIC_AUTH, FORM_AUTH, CLIENT_CERT_AUTH, DIGEST_AUTH (suitable for == comparison) or the container-specific string indicating the authentication scheme, or null if the request was not authenticated.

gpublic Cookie[] get Cookies()
ggetCookies()

gCookies

gpublic Cookie[] get Cookies()

getCookies

The default behavior of this method is to return getCookies() on the wrapped request object.

Specified by:
    getCookies in interface HttpServletRequest

Returns:
an array of all the Cookies included with this request, or null if the request has no cookies

gpublic long getDateHeader(String name)
ggetDateHeader(String name)

getDateHeader

public long getDateHeader(String name)

getDateHeader

The default behavior of this method is to return getDateHeader(String name) on the wrapped request object.
Specified by: 
**getDateTimeHeader** in interface [HttpServletRequest](https://docs.oracle.com/en/java/javase/11/docs/api/index.html)

Parameters: 
- **name** - a **String** specifying the name of the header

Returns: 
a long value representing the date specified in the header expressed as the number of milliseconds since January 1, 1970 GMT, or -1 if the named header was not included with the request.

---

**public String** **getHeader** (String name)

**getHeader(String name)**

**getHeader**

**public String** **getHeader**(String name)

The default behavior of this method is to return **getHeader(String name)** on the wrapped request object.

Specified by: 
**getHeader** in interface [HttpServletRequest](https://docs.oracle.com/en/java/javase/11/docs/api/index.html)

Parameters: 
- **name** - a **String** specifying the header name

Returns: 
a String containing the value of the requested header, or **null** if the request does not have a header of that name.

---

**public java.util.Enumeration<E>** **getHeaders** (String name)

**getHeaders(String name)**

**getHeaders**

**public Enumeration** **getHeaders**(String name)

---
The default behavior of this method is to return getHeaders(String name) on the wrapped request object.

**Specified by:**
getHeaders in interface HttpServletRequest

**Parameters:**
name - a String specifying the header name

**Returns:**
an Enumeration containing the values of the requested header. If the request does not have any headers of that name return an empty enumeration. If the container does not allow access to header information, return null

```java
public java.util.Enumeration<E> getHeaderNames()

getHeaderNames()
```

**getHeaderNames**

```java
public Enumeration getHeaderNames()

 The default behavior of this method is to return getHeaderNames() on the wrapped request object.

**Specified by:**
getHeaderNames in interface HttpServletRequest

**Returns:**
an enumeration of all the header names sent with this request; if the request has no headers, an empty enumeration; if the servlet container does not allow servlets to use this method, null

```java
public int getIntHeader(String name)

getIntHeader(String name)
```
**getIntHeader**

```java
class HttpRequest {
    public int getIntHeader(String name) {
        // Implementation...
    }
}
```

The default behavior of this method is to return `getIntHeader(String name)` on the wrapped request object.

**Specified by:**
`getIntHeader` in interface `HttpServletRequest`

**Parameters:**
- `name` - a String specifying the name of a request header

**Returns:**
- an integer expressing the value of the request header or -1 if the request doesn't have a header of this name

---

**public String getMethod()**

```java
class HttpRequest {
    public String getMethod() {
        // Implementation...
    }
}
```

The default behavior of this method is to return `getMethod()` on the wrapped request object.

**Specified by:**
`getMethod` in interface `HttpServletRequest`

**Returns:**
- a String specifying the name of the method with which this request was made

---

**public String getPathInfo()**

```java
class HttpRequest {
    public String getPathInfo() {
        // Implementation...
    }
}
```
**getPathInfo**

public String getPathInfo()

The default behavior of this method is to return getPathInfo() on the wrapped request object.

**Specified by:**
getPathInfo in interface HttpServletRequest

**Returns:**
a String, decoded by the web container, specifying extra path information that comes after the servlet path but before the query string in the request URL; or null if the URL does not have any extra path information

---

**public String getPathTranslated()**

**getPathTranslated()**

**getPathTranslated**

public String getPathTranslated()

The default behavior of this method is to return getPathTranslated() on the wrapped request object.

**Specified by:**
getPathTranslated in interface HttpServletRequest

**Returns:**
a String specifying the real path, or null if the URL does not have any extra path information

---

**public String getContextPath()**

**getContextPath()**
**getContextPath**

public String getContextPath()

The default behavior of this method is to return getContextPath() on the wrapped request object.

**Specified by:**
getContextPath in interface HttpServletRequest

**Returns:**
a String specifying the portion of the request URI that indicates the context of the request

**See Also:**
ServletContext.getContextPath()

---

**public String getQueryString()**

getQueryString()

---

**getQueryString**

public String getQueryString()

The default behavior of this method is to return getQueryString() on the wrapped request object.

**Specified by:**
getQueryString in interface HttpServletRequest

**Returns:**
a String containing the query string or null if the URL contains no query string. The value is not decoded by the container.

---

**public String getRemoteUser()**

getRemoteUser()
getRemoteUser

public String getRemoteUser()

    The default behavior of this method is to return getRemoteUser() on the wrapped request object.

    Specified by:
        getRemoteUser in interface HttpServletRequest

    Returns:
        a String specifying the login of the user making this request, or null if the user login is not known

____________________________

g瓤lic boolean isUserInRole(String role)

    The default behavior of this method is to return isUserInRole(String role) on the wrapped request object.

    Specified by:
        isUserInRole in interface HttpServletRequest

    Parameters:
        role - a String specifying the name of the role

    Returns:
        a boolean indicating whether the user making this request belongs to a given role; false if the user has not been authenticated

____________________________

public java.security.Principal getUserPrincipal()

    getUserPrincipal()
**getUserPrincipal**

```java
gpublic Principal getUserPrincipal()
```

The default behavior of this method is to return `getUserPrincipal()` on the wrapped request object.

**Specified by:**
`getUserPrincipal` in interface `HttpServletRequest`

**Returns:**
a `java.security.Principal` containing the name of the user making this request; `null` if the user has not been authenticated

---

**public String getRequestedSessionId()**

```
gpublic String getRequestedSessionId()
```

**getRequestedSessionId**

```java
gpublic String getRequestedSessionId()
```

The default behavior of this method is to return `getRequestedSessionId()` on the wrapped request object.

**Specified by:**
`getRequestedSessionId` in interface `HttpServletRequest`

**Returns:**
a `String` specifying the session ID, or `null` if the request did not specify a session ID

**See Also:**
`HttpServletRequest.isRequestedSessionIdValid()`

---

**public String getRequestURI()**

```
gpublic String getRequestURI()
```

**getRequestURI**

```java
gpublic String getRequestURI()
```

---
getRequestURI

public String getRequestURI()

The default behavior of this method is to return getRequestURI() on the wrapped request object.

Specified by: 
    getRequestURI in interface HttpServletRequest

Returns: 
a String containing the part of the URL from the protocol name up to the query string

See Also: 
    HttpUtils.getRequestURL(javax.servlet.http.HttpServletRequest)

getRequestURL

public StringBuffer getRequestURL()

The default behavior of this method is to return getRequestURL() on the wrapped request object.

Specified by: 
    getRequestURL in interface HttpServletRequest

Returns: 
a StringBuffer object containing the reconstructed URL

getServletPath

public String getServletPath()

getServletPath
public String getServletPath()

    The default behavior of this method is to return getServletPath() on the wrapped request object.

    Specified by:
    getServletPath in interface HttpServletRequest

    Returns:
    a String containing the name or path of the servlet being called, as specified in the request URL, decoded, or an empty string if the servlet used to process the request is matched using the "/\*" pattern.

public HttpSession getSession(boolean create)
getSession(boolean create)

getSession

public HttpSession getSession(boolean create)

    The default behavior of this method is to return getSession(boolean create) on the wrapped request object.

    Specified by:
    getSession in interface HttpServletRequest

    Parameters:
    create - true to create a new session for this request if necessary; false to return null if there's no current session

    Returns:
    the HttpSession associated with this request or null if create is false and the request has no valid session

    See Also:
    HttpServletRequest.getSession()
getSession

public HttpSession getSession()

The default behavior of this method is to return getSession() on the wrapped request object.

Specified by:
    getSession in interface HttpServletRequest

Returns:
    the HttpSession associated with this request

See Also:
    HttpServletRequest.getSession(boolean)

isRequestedSessionIdValid

public boolean isRequestedSessionIdValid()

isRequestedSessionIdValid

public boolean isRequestedSessionIdValid()

The default behavior of this method is to return isRequestedSessionIdValid() on the wrapped request object.

Specified by:
    isRequestedSessionIdValid in interface HttpServletRequest

Returns:
    true if this request has an id for a valid session in the current session context; false otherwise

See Also:
    HttpServletRequest.getRequestedSessionId(), HttpServletRequest.getSession(boolean), HttpSessionContext

public boolean isRequestedSessionIdFromCookie()
isRequestedSessionIdFromCookie

isRequestedSessionIdFromCookie

public boolean isRequestedSessionIdFromCookie()

The default behavior of this method is to return
isRequestedSessionIdFromCookie() on the wrapped request object.

Specified by:
    isRequestedSessionIdFromCookie in interface HttpServletRequest
Returns:
    true if the session ID came in as a cookie; otherwise, false
See Also:
    HttpServletRequest.getSession(boolean)

isRequestedSessionIdFromURL

isRequestedSessionIdFromURL

public boolean isRequestedSessionIdFromURL()

The default behavior of this method is to return
isRequestedSessionIdFromURL() on the wrapped request object.

Specified by:
    isRequestedSessionIdFromURL in interface HttpServletRequest
Returns:
    true if the session ID came in as part of a URL; otherwise, false
See Also:
    HttpServletRequest.getSession(boolean)

isRequestedSessionIdFromUrl

isRequestedSessionIdFromUrl

public boolean isRequestedSessionIdFromUrl()
public boolean isRequestedSessionIdFromUrl()

The default behavior of this method is to return
isRequestedSessionIdFromUrl() on the wrapped request object.

Specified by:
   isRequestedSessionIdFromUrl in interface HttpServletRequest

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.servlet.http  Interface HttpServletResponse

All Superinterfaces:  
   ServletException

All Known Implementing Classes:  
   HttpServletResponseWrapper

public interface HttpServletResponse  
extends ServletException

Implements: ServletException  
Implemented by: HttpServletResponseWrapper

   ServletException  HTTP  HTTP  cookie

servlet  HttpServletResponse  servlet  service  doGet doPost

See also  
  javax.servlet.ServletResponse

Extends the ServletException interface to provide HTTP-specific functionality in sending a response. For example, it has methods to access HTTP headers and cookies.

The servlet container creates an HttpServletResponse object and passes it as an argument to the servlet's service methods (doGet, doPost, etc).

Author:  
   Various

See Also:  
   ServletException
<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC_ACCEPTED</td>
<td>Status code (202) indicating that a request was accepted for processing, but was not completed.</td>
</tr>
<tr>
<td>SC_BAD_GATEWAY</td>
<td>Status code (502) indicating that the HTTP server received an invalid response from a server it consulted when acting as a proxy or gateway.</td>
</tr>
<tr>
<td>SC_BAD_REQUEST</td>
<td>Status code (400) indicating the request sent by the client was syntactically incorrect.</td>
</tr>
<tr>
<td>SC_CONFLICT</td>
<td>Status code (409) indicating that the request could not be completed due to a conflict with the current state of the resource.</td>
</tr>
<tr>
<td>SC_CONTINUE</td>
<td>Status code (100) indicating the client can continue.</td>
</tr>
<tr>
<td>SC_CREATED</td>
<td>Status code (201) indicating the request succeeded and created a new resource on the server.</td>
</tr>
<tr>
<td>SC_EXPECTATION_FAILED</td>
<td>Status code (417) indicating that the server could not meet the expectation given in the Expect request header.</td>
</tr>
<tr>
<td>SC_FORBIDDEN</td>
<td>Status code (403) indicating the server understood the request but refused to fulfill it.</td>
</tr>
<tr>
<td>SC_FOUND</td>
<td>Status code (302) indicating that the resource reside temporarily under a different URI.</td>
</tr>
<tr>
<td>SC_GATEWAY_TIMEOUT</td>
<td>Status code (504) indicating that the server did not receive a timely response from the upstream server while acting as a gateway or proxy.</td>
</tr>
<tr>
<td>SC_GONE</td>
<td>Status code (410) indicating that the resource is no longer available at the server and no forwarding address is known.</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SC_HTTP_VERSION_NOT_SUPPORTED</td>
<td>Status code (505) indicating that the server does not support or refuses to support the HTTP protocol version that was used in the request message.</td>
</tr>
<tr>
<td>SC_INTERNAL_SERVER_ERROR</td>
<td>Status code (500) indicating an error inside the HTTP server which prevented it from fulfilling the request.</td>
</tr>
<tr>
<td>SC_LENGTH_REQUIRED</td>
<td>Status code (411) indicating that the request cannot be handled without a defined Content-Length.</td>
</tr>
<tr>
<td>SC_METHOD_NOT_ALLOWED</td>
<td>Status code (405) indicating that the method specified in the Request-Line is not allowed for the resource identified by the Request-URI.</td>
</tr>
<tr>
<td>SC_MOVED_PERMANENTLY</td>
<td>Status code (301) indicating that the resource has permanently moved to a new location, and that future references should use a new URI with their requests.</td>
</tr>
<tr>
<td>SC_MOVED_TEMPORARILY</td>
<td>Status code (302) indicating that the resource has temporarily moved to another location, but that future references should still use the original URI to access the resource.</td>
</tr>
<tr>
<td>SC_MULTIPLE_CHOICES</td>
<td>Status code (300) indicating that the requested resource corresponds to any one of a set of representations, each with its own specific location.</td>
</tr>
<tr>
<td>SC_NO_CONTENT</td>
<td>Status code (204) indicating that the request succeeded but that there was no new information to return.</td>
</tr>
<tr>
<td>SC_NON_AUTHORITATIVE_INFORMATION</td>
<td>Status code (203) indicating that the meta information presented by the client did not originate from the server.</td>
</tr>
<tr>
<td>SC_NOT_ACCEPTABLE</td>
<td>Status code (406) indicating that the resource identified by the request is only capable of generating response entities.</td>
</tr>
</tbody>
</table>
which have content characteristics not acceptable according to the accept headers sent in the request.

<table>
<thead>
<tr>
<th>Status Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>404</td>
<td>SC_NOT_FOUND</td>
</tr>
<tr>
<td>501</td>
<td>SC_NOT_IMPLEMENTED</td>
</tr>
<tr>
<td>304</td>
<td>SC_NOT_MODIFIED</td>
</tr>
<tr>
<td>200</td>
<td>SC_OK</td>
</tr>
<tr>
<td>206</td>
<td>SC_PARTIAL_CONTENT</td>
</tr>
<tr>
<td>402</td>
<td>SC_PAYMENT_REQUIRED</td>
</tr>
<tr>
<td>412</td>
<td>SC_PRECONDITION_FAILED</td>
</tr>
<tr>
<td>407</td>
<td>SC_PROXY_AUTHENTICATION_REQUIRED</td>
</tr>
<tr>
<td>413</td>
<td>SC_REQUEST_ENTITY_TOO_LARGE</td>
</tr>
<tr>
<td>408</td>
<td>SC_REQUEST_TIMEOUT</td>
</tr>
<tr>
<td></td>
<td>SC_REQUEST_URI_TOO_LONG</td>
</tr>
<tr>
<td>Status code</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
</tr>
<tr>
<td>414</td>
<td>Static int indicating that the server is refusing to service the request because the <code>Request-URI</code> is longer than the server is willing to interpret.</td>
</tr>
<tr>
<td>416</td>
<td>Static int indicating that the server cannot serve the requested byte range.</td>
</tr>
<tr>
<td>205</td>
<td>Static int indicating that the agent <strong>SHOULD</strong> reset the document view which caused the request to be sent.</td>
</tr>
<tr>
<td>303</td>
<td>Static int indicating that the response to the request can be found under a different URI.</td>
</tr>
<tr>
<td>503</td>
<td>Static int indicating that the HTTP server is temporarily overloaded, and unable to handle the request.</td>
</tr>
<tr>
<td>101</td>
<td>Static int indicating the server is switching protocols according to <code>Upgrade</code> header.</td>
</tr>
<tr>
<td>307</td>
<td>Static int indicating that the requested resource resides temporarily under a different URI.</td>
</tr>
<tr>
<td>401</td>
<td>Static int indicating that the request requires <code>HTTP</code> authentication.</td>
</tr>
<tr>
<td>415</td>
<td>Static int indicating that the server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.</td>
</tr>
<tr>
<td>305</td>
<td>Static int indicating that the requested resource <strong>MUST</strong> be accessed through the proxy given by the <code>Location</code> field.</td>
</tr>
</tbody>
</table>

**Method Summary**
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addCookie(Cookie cookie)</code></td>
<td>Adds the specified cookie to the response.</td>
</tr>
<tr>
<td><code>addDateHeader(String name, long date)</code></td>
<td>Adds a response header with the given name and date-value.</td>
</tr>
<tr>
<td><code>addHeader(String name, String value)</code></td>
<td>Adds a response header with the given name and value.</td>
</tr>
<tr>
<td><code>addIntHeader(String name, int value)</code></td>
<td>Adds a response header with the given name and integer value.</td>
</tr>
<tr>
<td><code>containsHeader(String name)</code></td>
<td>Returns a boolean indicating whether the named response header has already been set.</td>
</tr>
<tr>
<td><code>encodeRedirectUrl(String url)</code></td>
<td>Encodes the specified URL for use in the <code>sendRedirect</code> method or, if encoding is not needed, returns the URL unchanged.</td>
</tr>
<tr>
<td><code>encodeRedirectURL(String url)</code></td>
<td>Encodes the specified URL for use in the <code>sendRedirect</code> method or, if encoding is not needed, returns the URL unchanged.</td>
</tr>
<tr>
<td><code>encodeUrl(String url)</code></td>
<td>Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged.</td>
</tr>
<tr>
<td><code>encodeURL(String url)</code></td>
<td>Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged.</td>
</tr>
<tr>
<td><code>sendError(int sc)</code></td>
<td>Sends an error response to the client using the specified status code and clearing the buffer.</td>
</tr>
<tr>
<td><code>sendError(int sc, String msg)</code></td>
<td>Sends an error response to the client using the specified status.</td>
</tr>
<tr>
<td><code>sendRedirect(String location)</code></td>
<td>Sends a temporary redirect response to the client using the specified redirect location URL.</td>
</tr>
</tbody>
</table>
void setDateHeader(String name, long date)
Sets a response header with the given name and date-value.

void setHeader(String name, String value)
Sets a response header with the given name and value.

void setIntHeader(String name, int value)
Sets a response header with the given name and integer value.

void setStatus(int sc)
Sets the status code for this response.

void setStatus(int sc, String sm)
Deprecated. As of version 2.1, due to ambiguous meaning of the message parameter. To set a status code use setStatus(int), to send an error with a description use sendError(int, String). Sets the status code and message for this response.

Methods inherited from interface javax.servlet.ServletResponse
flushBuffer, getBufferSize, getCharacterEncoding, getContentType, getLocale, getOutputStream, getWriter, isCommitted, reset, resetBuffer, setBufferSize, setCharacterEncoding, setContentLength, setContentType, setLocale

Field Detail

SC_CONTINUE

static final int SC_CONTINUE
Status code (100) indicating the client can continue.

See Also:
Constant Field Values
SC_SWITCHING_PROTOCOLS

static final int SC_SWITCHING_PROTOCOLS

    Status code (101) indicating the server is switching protocols according to Upgrade header.

    See Also:
    Constant Field Values

SC_OK

static final int SC_OK

    Status code (200) indicating the request succeeded normally.

    See Also:
    Constant Field Values

SC_CREATED

static final int SC_CREATED

    Status code (201) indicating the request succeeded and created a new resource on the server.

    See Also:
    Constant Field Values
SC_ACCEPTED

static final int SC_ACCEPTED

Status code (202) indicating that a request was accepted for processing, but was not completed.

See Also:
   Constant Field Values

---

SC_NON_AUTHORITATIVE_INFORMATION

static final int SC_NON_AUTHORITATIVE_INFORMATION

Status code (203) indicating that the meta information presented by the client did not originate from the server.

See Also:
   Constant Field Values

---

SC_NO_CONTENT

static final int SC_NO_CONTENT

Status code (204) indicating that the request succeeded but that there was no new information to return.

See Also:
   Constant Field Values

---

SC_RESET_CONTENT
static final int SC_RESET_CONTENT

Status code (205) indicating that the agent SHOULD reset the document view which caused the request to be sent.

See Also:
Constant Field Values

---

SC_PARTIAL_CONTENT

static final int SC_PARTIAL_CONTENT

Status code (206) indicating that the server has fulfilled the partial GET request for the resource.

See Also:
Constant Field Values

---

SC_MULTIPLE_CHOICES

static final int SC_MULTIPLE_CHOICES

Status code (300) indicating that the requested resource corresponds to any one of a set of representations, each with its own specific location.

See Also:
Constant Field Values

---

SC_MOVED_PERMANENTLY
static final int SC_MOVED_PERMANENTLY

Status code (301) indicating that the resource has permanently moved to a new location, and that future references should use a new URI with their requests.

See Also:
Constant Field Values

---

SC_MOVED_TEMPORARILY

static final int SC_MOVED_TEMPORARILY

Status code (302) indicating that the resource has temporarily moved to another location, but that future references should still use the original URI to access the resource. This definition is being retained for backwards compatibility. SC_FOUND is now the preferred definition.

See Also:
Constant Field Values

---

SC_FOUND

static final int SC_FOUND

Status code (302) indicating that the resource reside temporarily under a different URI. Since the redirection might be altered on occasion, the client should continue to use the Request-URI for future requests. (HTTP/1.1) To represent the status code (302), it is recommended to use this variable.

See Also:
Constant Field Values
SC_SEE_OTHER

**static final int SC_SEE_OTHER**

Status code (303) indicating that the response to the request can be found under a different URI.

**See Also:**
- [Constant Field Values](#)

SC_NOT_MODIFIED

**static final int SC_NOT_MODIFIED**

Status code (304) indicating that a conditional GET operation found that the resource was available and not modified.

**See Also:**
- [Constant Field Values](#)

SC_USE_PROXY

**static final int SC_USE_PROXY**

Status code (305) indicating that the requested resource *MUST* be accessed through the proxy given by the *Location* field.

**See Also:**
- [Constant Field Values](#)
SC_TEMPORARY_REDIRECT

static final int SC_TEMPORARY_REDIRECT

Status code (307) indicating that the requested resource resides temporarily under a different URI. The temporary URI SHOULD be given by the Location field in the response.

See Also:
Constant Field Values

SC_BAD_REQUEST

static final int SC_BAD_REQUEST

Status code (400) indicating the request sent by the client was syntactically incorrect.

See Also:
Constant Field Values

SC_UNAUTHORIZED

static final int SC_UNAUTHORIZED

Status code (401) indicating that the request requires HTTP authentication.

See Also:
Constant Field Values
SC_PAYMENT_REQUIRED

static final int SC_PAYMENT_REQUIRED

Status code (402) reserved for future use.

See Also:
Constant Field Values

SC_FORBIDDEN

static final int SC_FORBIDDEN

Status code (403) indicating the server understood the request but refused to fulfill it.

See Also:
Constant Field Values

SC_NOT_FOUND

static final int SC_NOT_FOUND

Status code (404) indicating that the requested resource is not available.

See Also:
Constant Field Values
SC_METHOD_NOT_ALLOWED

static final int SC_METHOD_NOT_ALLOWED

Status code (405) indicating that the method specified in the Request-Line is not allowed for the resource identified by the Request-URI.

See Also:
Constant Field Values

SC_NOT_ACCEPTABLE

static final int SC_NOT_ACCEPTABLE

Status code (406) indicating that the resource identified by the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

See Also:
Constant Field Values

SC_PROXY_AUTHENTICATION_REQUIRED

static final int SC_PROXY_AUTHENTICATION_REQUIRED

Status code (407) indicating that the client MUST first authenticate itself with the proxy.

See Also:
Constant Field Values
SC_REQUEST_TIMEOUT

static final int SC_REQUEST_TIMEOUT

Status code (408) indicating that the client did not produce a request within the time that the server was prepared to wait.

See Also:
Constant Field Values

SC_CONFLICT

static final int SC_CONFLICT

Status code (409) indicating that the request could not be completed due to a conflict with the current state of the resource.

See Also:
Constant Field Values

SC_GONE

static final int SC_GONE

Status code (410) indicating that the resource is no longer available at the server and no forwarding address is known. This condition SHOULD be considered permanent.

See Also:
Constant Field Values
SC_LENGTH_REQUIRED

static final int SC_LENGTH_REQUIRED

Status code (411) indicating that the request cannot be handled without a defined Content-Length.

See Also:
Constant Field Values

SC_PRECONDITION_FAILED

static final int SC_PRECONDITION_FAILED

Status code (412) indicating that the precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

See Also:
Constant Field Values

SC_REQUEST_ENTITY_TOO_LARGE

static final int SC_REQUEST_ENTITY_TOO_LARGE

Status code (413) indicating that the server is refusing to process the request because the request entity is larger than the server is willing or able to process.

See Also:
Constant Field Values
SC_REQUEST_URI_TOO_LONG

static final int SC_REQUEST_URI_TOO_LONG

Status code (414) indicating that the server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

See Also:
Constant Field Values

---

SC_UNSUPPORTED_MEDIA_TYPE

static final int SC_UNSUPPORTED_MEDIA_TYPE

Status code (415) indicating that the server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

See Also:
Constant Field Values

---

SC_REQUESTED_RANGE_NOT_SATISFIABLE

static final int SC_REQUESTED_RANGE_NOT_SATISFIABLE

Status code (416) indicating that the server cannot serve the requested byte range.

See Also:
Constant Field Values
SC_EXPECTATION_FAILED

static final int SC_EXPECTATION_FAILED

Status code (417) indicating that the server could not meet the expectation given in the Expect request header.

See Also:
Constant Field Values

SC_INTERNAL_SERVER_ERROR

static final int SC_INTERNAL_SERVER_ERROR

Status code (500) indicating an error inside the HTTP server which prevented it from fulfilling the request.

See Also:
Constant Field Values

SC_NOT_IMPLEMENTED

static final int SC_NOT_IMPLEMENTED

Status code (501) indicating the HTTP server does not support the functionality needed to fulfill the request.

See Also:
Constant Field Values
SC_BAD_GATEWAY

static final int SC_BAD_GATEWAY

Status code (502) indicating that the HTTP server received an invalid response from a server it consulted when acting as a proxy or gateway.

See Also:
Constant Field Values

SC_SERVICE_UNAVAILABLE

static final int SC_SERVICE_UNAVAILABLE

Status code (503) indicating that the HTTP server is temporarily overloaded, and unable to handle the request.

See Also:
Constant Field Values

SC_GATEWAY_TIMEOUT

static final int SC_GATEWAY_TIMEOUT

Status code (504) indicating that the server did not receive a timely response from the upstream server while acting as a gateway or proxy.

See Also:
Constant Field Values
SC_HTTP_VERSION_NOT_SUPPORTED

static final int SC_HTTP_VERSION_NOT_SUPPORTED

Status code (505) indicating that the server does not support or refuses to support the HTTP protocol version that was used in the request message.

See Also:
Constant Field Values

### Method Detail

**public void addCookie(Cookie cookie)**

```java
public void addCookie(Cookie cookie)
```

Adds the specified cookie to the response. This method can be called multiple times to set more than one cookie.

**Parameters:**
- `cookie` - the Cookie to return to the client

**public boolean containsHeader(String name)**

```java
public boolean containsHeader(String name)
```

**ContainsHeader**

```java
public boolean containsHeader(String name)
```
boolean `containsHeader(String name)`

Returns a boolean indicating whether the named response header has already been set.

**Parameters:**
- `name` - the header name

**Returns:**
- `true` if the named response header has already been set; `false` otherwise

---

`public String encodeURL(String url)`

Encodes the specified URL by including the session ID in it, or, if encoding is not needed, returns the URL unchanged. The implementation of this method includes the logic to determine whether the session ID needs to be encoded in the URL. For example, if the browser supports cookies, or session tracking is turned off, URL encoding is unnecessary.

For robust session tracking, all URLs emitted by a servlet should be run through this method. Otherwise, URL rewriting cannot be used with browsers which do not support cookies.

**Parameters:**
public String encodeRedirectURL(String url)

Encodes the specified URL for use in the `sendRedirect` method or, if encoding is not needed, returns the URL unchanged. The implementation of this method includes the logic to determine whether the session ID needs to be encoded in the URL. Because the rules for making this determination can differ from those used to decide whether to encode a normal link, this method is separated from the `encodeURL` method.

All URLs sent to the `HttpServletResponse.sendRedirect` method should be run through this method. Otherwise, URL rewriting cannot be used with browsers which do not support cookies.

**Parameters:**

- `url` - the url to be encoded.

**Returns:**

- the encoded URL if encoding is needed; the unchanged URL
See Also:

`sendRedirect(java.lang.String), encodeUrl(java.lang.String)`

---

**public String encodeUrl(String url)**

deprecated 2.1 `encodeURL(String url)`

```
url
return
```

**encodeUrl**

`String encodeUrl(String url)`

**Deprecated. As of version 2.1, use encodeURL(String url) instead**

**Parameters:**

`url` - the url to be encoded.

**Returns:**

the encoded URL if encoding is needed; the unchanged URL otherwise.

---

**public String encodeRedirectUrl(String url)**

deprecated 2.1 `encodeRedirectURL(String url)`

```
url
return
```

**encodeRedirectUrl**

`String encodeRedirectUrl(String url)`

**Deprecated. As of version 2.1, use encodeRedirectURL(String url) instead**

**Parameters:**
url - the url to be encoded.

**Returns:**
the encoded URL if encoding is needed; the unchanged URL otherwise.

```java
public void sendError(int sc, String msg) throws java.io.IOException
HTML  "text/html" cookie
msg

IllegalStateException
  sc
  msg
Throws java.io.IOException:
Throws IllegalArgumentException:

sendError

void sendError(int sc,
  String msg)
throws IOException

Sends an error response to the client using the specified status. The server defaults to creating the response to look like an HTML-formatted server error page containing the specified message, setting the content type to "text/html", leaving cookies and other headers unmodified. If an error-page declaration has been made for the web application corresponding to the status code passed in, it will be served back in preference to the suggested msg parameter.

If the response has already been committed, this method throws an IllegalStateException. After using this method, the response should be considered to be committed and should not be written to.

**Parameters:**
public void sendError(int sc) throws java.io.IOException

IllegalStateException

Throws java.io.IOException:

Throws IllegalStateException:

sendError

void sendError(int sc)
throws IOException

Sends an error response to the client using the specified status code and clearing the buffer.

If the response has already been committed, this method throws an IllegalStateException. After using this method, the response should be considered to be committed and should not be written to.

Parameters:
sc - the error status code

Throws:
IOException - If an input or output exception occurs
IllegalStateException - If the response was committed before this method call

public void sendRedirect(String location) throws
java.io.IOException
URL URLservlet URL URL '/'
URI '/' servlet

IllegalStateException

Throws java.io.IOException:
Throwss IllegalStateException: URL URL

sendRedirect

void sendRedirect(String location)
throws IOException

Sends a temporary redirect response to the client using the specified redirect location URL. This method can accept relative URLs; the servlet container must convert the relative URL to an absolute URL before sending the response to the client. If the location is relative without a leading '/' the container interprets it as relative to the current request URI. If the location is relative with a leading '/' the container interprets it as relative to the servlet container root.

If the response has already been committed, this method throws an IllegalStateException. After using this method, the response should be considered to be committed and should not be written to.

Parameters:
location - the redirect location URL

Throws:
IOException - If an input or output exception occurs
IllegalStateException - If the response was committed or if a partial URL is given and cannot be converted into a valid URL

public void setDateHeader(String name, long date)
void setDateHeader(String name, long date)

Sets a response header with the given name and date-value. The date is specified in terms of milliseconds since the epoch. If the header had already been set, the new value overwrites the previous one. The containsHeader method can be used to test for the presence of a header before setting its value.

Parameters:
- name - the name of the header to set
- date - the assigned date value

See Also:
- containsHeader(java.lang.String)
- addDateHeader(java.lang.String, long)

public void addDateHeader(String name, long date)

void addDateHeader(String name, long date)

Adds a response header with the given name and date-value. The date is specified in terms of milliseconds since the epoch. This method allows response headers to have multiple values.
public void setHeader(String name, String value)

Sets a response header with the given name and value. If the header had already been set, the new value overwrites the previous one. The containsHeader method can be used to test for the presence of a header before setting its value.

Parameters:

name - the name of the header
value - the header value If it contains octet string, it should be encoded according to RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt)

See Also:

containsHeader(java.lang.String),
addHeader(java.lang.String, java.lang.String)
addHeader

void addHeader(String name, String value)

Adds a response header with the given name and value. This method allows response headers to have multiple values.

Parameters:
   name - the name of the header
   value - the additional header value If it contains octet string, it should be encoded according to RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt)

See Also:
setHeader(java.lang.String, java.lang.String)

public void setIntHeader(String name, int value)

Sets a response header with the given name and integer value. If the header had already been set, the new value overwrites the previous one. The containsHeader method can be used to test for the presence of a header before setting its value.

Parameters:
   name - the name of the header
public void addIntHeader(String name, int value)

   name
   value
 See also      setIntHeader

addIntHeader

void addIntHeader(String name,
                  int value)

   Adds a response header with the given name and integer value. This method allows response headers to have multiple values.

Parameters:

   name - the name of the header
   value - the assigned integer value
See Also:

   setIntHeader(java.lang.String, int)

public void setStatus(int sc)

SC_OK  SC_MOVED_TEMPORARILY Web

   sendError

Location  cookie

   sc
 See also      sendError
setStatus

void setStatus(int sc)

Sets the status code for this response. This method is used to set the return status code when there is no error (for example, for the status codes SC_OK or SC_MOVED_TEMPORARILY). If there is an error, and the caller wishes to invoke an error-page defined in the web application, the sendError method should be used instead.

The container clears the buffer and sets the Location header, preserving cookies and other headers.

Parameters:

  sc - the status code

See Also:

  sendError(int, java.lang.String)

public void setStatus(int sc, String sm)

deprecated 2.1

Sets the status code and message for this response.

Parameters:

  sc - the status code
  sm - the status message

Deprecated. As of version 2.1, due to ambiguous meaning of the message parameter. To set a status code use setStatus(int), to send an error with a description use sendError(int, String). Sets the status code and message for this response.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet.http Class HttpServletResponseWrapper

java.lang.Object
   ▼ javax.servlet.ServletResponseWrapper
      ▼ javax.servlet.http.HttpServletResponseWrapper

All Implemented Interfaces:
   HttpServletResponse, ServletResponse

public class HttpServletResponseWrapper

extends ServletResponseWrapper
implements HttpServletResponse

Extends: ServletResponseWrapper
Implements: HttpServletResponse

HttpServletResponse Servlet Wrapper Decorator

   since v 2.3

See also javax.servlet.http.HttpServletResponse

Provides a convenient implementation of the HttpServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet. This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped response object.

Since:
   v 2.3

Author:
   Various

See Also:
   HttpServletResponse
### Field Summary

**Fields inherited from interface**
java.servlet.http.**HttpServletResponse**

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
</table>

### Constructor Summary

**HttpServletResponseWrapper**(*HttpServletResponse* response)

Constructs a response adaptor wrapping the given response.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addCookie</strong>(<em>Cookie</em> cookie)</td>
<td>The default behavior of this method is to call addCookie(<em>Cookie</em> cookie) on the wrapped response object.</td>
</tr>
<tr>
<td><strong>addDateHeader</strong>(<em>String</em> name, long date)</td>
<td>The default behavior of this method is to call addDateHeader(<em>String</em> name, long date) on the wrapped response object.</td>
</tr>
<tr>
<td><strong>addHeader</strong>(<em>String</em> name, <em>String</em> value)</td>
<td>The default behavior of this method is to return addHeader(<em>String</em> name, <em>String</em> value) on the wrapped response object.</td>
</tr>
<tr>
<td><strong>addIntHeader</strong>(<em>String</em> name, int value)</td>
<td>The default behavior of this method is to return addIntHeader(<em>String</em> name, int value) on the wrapped response object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>void</code></td>
<td>The default behavior of this method is to call <code>addIntHeader(String name, int value)</code> on the wrapped response object.</td>
</tr>
<tr>
<td>boolean</td>
<td>The default behavior of this method is to call <code>containsHeader(String name)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>encodeRedirectUrl(String url)</code></td>
<td>The default behavior of this method is to return <code>encodeRedirectUrl(String url)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>encodeRedirectURL(String url)</code></td>
<td>The default behavior of this method is to return <code>encodeRedirectURL(String url)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>encodeUrl(String url)</code></td>
<td>The default behavior of this method is to call <code>encodeUrl(String url)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>encodeURL(String url)</code></td>
<td>The default behavior of this method is to call <code>encodeURL(String url)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>sendError(int sc)</code></td>
<td>The default behavior of this method is to call <code>sendError(int sc)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>sendError(int sc, String msg)</code></td>
<td>The default behavior of this method is to call <code>sendError(int sc, String msg)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>sendRedirect(String location)</code></td>
<td>The default behavior of this method is to return <code>sendRedirect(String location)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>setDateHeader(String name, long date)</code></td>
<td>The default behavior of this method is to call <code>setDateHeader(String name, long date)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>setHeader(String name, String value)</code></td>
<td>The default behavior of this method is to return <code>setHeader(String name, String value)</code> on the wrapped response object.</td>
</tr>
<tr>
<td><code>setIntHeader(String name, int value)</code></td>
<td></td>
</tr>
</tbody>
</table>
void

The default behavior of this method is to call setIntHeader(String name, int value) on the wrapped response object.

void

**setStatus**(int sc)

The default behavior of this method is to call setStatus(int sc) on the wrapped response object.

void

**setStatus**(int sc, String sm)

The default behavior of this method is to call setStatus(int sc, String sm) on the wrapped response object.

Methods inherited from class java.servlet.**ServletResponseWrapper**

flushBuffer, getBufferSize, getCharacterEncoding, getContentType, getLocale, getOutputStream, getResponse,.getWriter, isCommitted, reset, resetBufferSize, setBufferSize, setCharacterEncoding, setCurrentLength, setContentType, setLocale, setResponse

Methods inherited from class java.lang.**Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface java.servlet.**ServletRequest**

flushBuffer, getBufferSize, getCharacterEncoding, getContentType, getLocale, getOutputStream, getWriter, isCommitted, reset, resetBuffer, setBufferSize, setCharacterEncoding, setCurrentLength, setContentType, setLocale

**Constructor Detail**

class **public HttpServletServletResponseWrapper**(HttpServletRequest response)

Throws IllegalArgumentException: response null
HttpServletResponseWrapper

public HttpServletResponseWrapper(HttpServletResponse response)

Constructs a response adaptor wrapping the given response.

Throws:

IllegalArgumentException - if the response is null

Method Detail

public void addCookie(Cookie cookie)

addCookie(Cookie cookie)

addCookie

public void addCookie(Cookie cookie)

The default behavior of this method is to call addCookie(Cookie cookie) on the wrapped response object.

Specified by:

addCookie in interface HttpServletResponse

Parameters:

cookie - the Cookie to return to the client

public boolean containsHeader(String name)

containsHeader(String name)

containsHeader

public boolean containsHeader(String name)

The default behavior of this method is to call containsHeader(String
name) on the wrapped response object.

**Specified by:**
- `containsHeader` in interface `HttpServletResponse`

**Parameters:**
- `name` - the header name

**Returns:**
- `true` if the named response header has already been set; `false` otherwise

---

```java
public String encodeURL(String url)
```

**encodeURL**

```java
public String encodeURL(String url)
```

The default behavior of this method is to call `encodeURL(String url)` on the wrapped response object.

**Specified by:**
- `encodeURL` in interface `HttpServletResponse`

**Parameters:**
- `url` - the url to be encoded.

**Returns:**
- the encoded URL if encoding is needed; the unchanged URL otherwise.

---

```java
public String encodeRedirectURL(String url)
```

**encodeRedirectURL**

```java
public String encodeRedirectURL(String url)
```

The default behavior of this method is to call `encodeRedirectURL(String url)` on the wrapped response object.

**Specified by:**
- `encodeRedirectURL` in interface `HttpServletResponse`

**Parameters:**
- `url` - the url to be encoded.

**Returns:**
- the encoded URL if encoding is needed; the unchanged URL otherwise.
The default behavior of this method is to return encodeRedirectURL(String url) on the wrapped response object.

**Specified by:**
   encodeRedirectURL in interface HttpServletResponse

**Parameters:**
url - the url to be encoded.

**Returns:**
the encoded URL if encoding is needed; the unchanged URL otherwise.

**See Also:**
   HttpServletResponse.sendRedirect(java.lang.String),
   HttpServletResponse.encodeUrl(java.lang.String)

---

public String encodeUrl(String url)
encodeUrl(String url)

codeUrl

public String encodeUrl(String url)

The default behavior of this method is to call encodeUrl(String url) on the wrapped response object.

**Specified by:**
   encodeUrl in interface HttpServletResponse

**Parameters:**
url - the url to be encoded.

**Returns:**
the encoded URL if encoding is needed; the unchanged URL otherwise.

---

public String encodeRedirectUrl(String url)
encodeRedirectUrl(String url)
**encodeRedirectUrl**

public String encodeRedirectUrl(String url)

The default behavior of this method is to return encodeRedirectUrl(String url) on the wrapped response object.

**Specified by:**
  
  `encodeRedirectUrl` in interface `HttpServletResponse`

**Parameters:**

  url - the url to be encoded.

**Returns:**

  the encoded URL if encoding is needed; the unchanged URL otherwise.

---

**sendError**

public void sendError(int sc, String msg) throws IOException

**sendError(int sc, String msg)**

**sendError**

public void sendError(int sc, String msg) throws IOException

The default behavior of this method is to call sendError(int sc, String msg) on the wrapped response object.

**Specified by:**

  `sendError` in interface `HttpServletResponse`

**Parameters:**

  sc - the error status code
  
  msg - the descriptive message

**Throws:**

  `IOException` - If an input or output exception occurs
public void sendError(int sc) throws java.io.IOException
sendError(int sc)

sendError

public void sendError(int sc)
    throws IOException

    The default behavior of this method is to call sendError(int sc) on the wrapped response object.

    Specified by:
        sendError in interface HttpServletResponse

    Parameters:
        sc - the error status code

    Throws:
        IOException - If an input or output exception occurs

public void sendRedirect(String location) throws java.io.IOException
sendRedirect(String location)

sendRedirect

public void sendRedirect(String location)
    throws IOException

    The default behavior of this method is to return sendRedirect(String location) on the wrapped response object.

    Specified by:
        sendRedirect in interface HttpServletResponse

    Parameters:
        location - the redirect location URL

    Throws:
        IOException - If an input or output exception occurs
public void setDateHeader(String name, long date)
setDateHeader(String name, long date)

setDateHeader

public void setDateHeader(String name, long date)

The default behavior of this method is to call setDateHeader(String name, long date) on the wrapped response object.

Specified by:
   setDateHeader in interface HttpServletResponse
Parameters:
   name - the name of the header to set
date - the assigned date value
See Also:
   HttpServletResponse.containsHeader(java.lang.String),
   HttpServletResponse.addDateHeader(java.lang.String, long)

public void addDateHeader(String name, long date)
addDateHeader(String name, long date)

addDateHeader

public void addDateHeader(String name, long date)

The default behavior of this method is to call addDateHeader(String name, long date) on the wrapped response object.

Specified by:
   addDateHeader in interface HttpServletResponse
Parameters:
   name - the name of the header to set
date - the additional date value

See Also:
HttpServletResponse.setDateHeader(java.lang.String, long)

public void setHeader(String name, String value)
setHeader(String name, String value)

setHeader

public void setHeader(String name, String value)

The default behavior of this method is to return setHeader(String name, String value) on the wrapped response object.

Specified by:
setHeader in interface HttpServletResponse

Parameters:
name - the name of the header
class value - the header value If it contains octet string, it should be encoded according to RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt)

See Also:
HttpServletResponse.containsHeader(java.lang.String),
HttpServletResponse.addHeader(java.lang.String, java.lang.String)

public void addHeader(String name, String value)
addHeader(String name, String value)

addHeader

public void addHeader(String name, String value)
The default behavior of this method is to return `addHeader(String name, String value)` on the wrapped response object.

**Specified by:**
- `addHeader` in interface `HttpServletResponse`

**Parameters:**
- `name` - the name of the header
- `value` - the additional header value If it contains octet string, it should be encoded according to RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt)

**See Also:**
- `HttpServletResponse.setHeader(java.lang.String, java.lang.String)`

---

```java
public void setIntHeader(String name, int value)
```

`setIntHeader` method

```java
public void setIntHeader(String name, int value)
```

The default behavior of this method is to call `setIntHeader(String name, int value)` on the wrapped response object.

**Specified by:**
- `setIntHeader` in interface `HttpServletResponse`

**Parameters:**
- `name` - the name of the header
- `value` - the assigned integer value

**See Also:**
- `HttpServletResponse.containsHeader(java.lang.String)`, `HttpServletResponse.addIntHeader(java.lang.String, int)`

---

```java
public void addIntHeader(String name, int value)
```

`addIntHeader` method

```java
public void addIntHeader(String name, int value)
```

---
addIntHeader

public void addIntHeader(String name, int value)

The default behavior of this method is to call addIntHeader(String name, int value) on the wrapped response object.

Specified by:
   addIntHeader in interface HttpServletResponse

Parameters:
   name - the name of the header
   value - the assigned integer value

See Also:
   HttpServletResponse.setIntHeader(java.lang.String, int)

public void setStatus(int sc)

setStatus(int sc)

setStatus

public void setStatus(int sc)

The default behavior of this method is to call setStatus(int sc) on the wrapped response object.

Specified by:
   setStatus in interface HttpServletResponse

Parameters:
   sc - the status code

See Also:
   HttpServletResponse.sendError(int, java.lang.String)

public void setStatus(int sc, String sm)
setStatus(int sc, String sm)

The default behavior of this method is to call setStatus(int sc, String sm) on the wrapped response object.

Parameters:
- sc - the status code
- sm - the status message

Specified by:
- setStatus in interface HttpServletResponse

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature
<table>
<thead>
<tr>
<th>Summary</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>
**javax.servlet.http**  
**Interface HttpSession**

```java
public interface HttpSession
```

Web

```java
Web
```

```java
servlet HTTP HTTP cookie URL
```

```java
servlet
```

```java
HttpSessionBindingListener
```

```java
VM
```

```java
HttpSessionActivationListener
```

```java
servlet cookieisNew true
```

```java
true
```

```java
Web (ServletContext)
```

```java
See javax.servlet.http.HttpSessionBindingListener,
```

```java
also javax.servlet.http.HttpSessionContext
```

Provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.

The servlet container uses this interface to create a session between an HTTP client and an HTTP server. The session persists for a specified time period, across more than one connection or page request from the user. A session usually corresponds to one user, who may visit a site many times. The server can maintain a session in many ways such as using cookies or rewriting URLs.

This interface allows servlets to
• View and manipulate information about a session, such as the session identifier, creation time, and last accessed time
• Bind objects to sessions, allowing user information to persist across multiple user connections

When an application stores an object in or removes an object from a session, the session checks whether the object implements HttpSessionBindingListener. If it does, the servlet notifies the object that it has been bound to or unbound from the session. Notifications are sent after the binding methods complete. For session that are invalidated or expire, notifications are sent after the session has been invalidated or expired.

When container migrates a session between VMs in a distributed container setting, all session attributes implementing the HttpSessionActivationListener interface are notified.

A servlet should be able to handle cases in which the client does not choose to join a session, such as when cookies are intentionally turned off. Until the client joins the session, isNew returns true. If the client chooses not to join the session, getSession will return a different session on each request, and isNew will always return true.

Session information is scoped only to the current web application (ServletContext), so information stored in one context will not be directly visible in another.

Author:
  Various
See Also:
  HttpSessionBindingListener, HttpSessionContext

---

**Method Summary**

```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAttribute</code></td>
<td>Returns the object bound with the specified name in this session, or null if no object is bound under the name.</td>
</tr>
</tbody>
</table>
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAttributeNames()</code></td>
<td>Returns an <code>Enumeration</code> of <code>String</code> objects containing the names of all the objects bound to this session.</td>
</tr>
<tr>
<td><code>getCreationTime()</code></td>
<td>Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.</td>
</tr>
<tr>
<td><code>getId()</code></td>
<td>Returns a string containing the unique identifier assigned to this session.</td>
</tr>
<tr>
<td><code>getLastAccessedTime()</code></td>
<td>Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container received the request.</td>
</tr>
<tr>
<td><code>getMaxInactiveInterval()</code></td>
<td>Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.</td>
</tr>
<tr>
<td><code>getServletContext()</code></td>
<td>Returns the <code>ServletContext</code> to which this session belongs.</td>
</tr>
<tr>
<td><code>getSessionContext()</code></td>
<td><strong>Deprecated.</strong> As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.</td>
</tr>
</tbody>
</table>
| `getValue(String name)`                    | **Deprecated.** As of Version 2.2, this method is replaced by `getAttribute(java.lang.String)`.
| `getValueNames()`                          | **Deprecated.** As of Version 2.2, this method is replaced by `getAttributeNames()` |
| `invalidate()`                             | Invalidates this session then unbinds any objects bound to it.              |
### Method Detail

#### public long getCreationTime()

1970 1 1

<table>
<thead>
<tr>
<th>return</th>
<th>long 1970 1 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>IllegalStateException:</td>
</tr>
</tbody>
</table>

**getCreationTime**

long getCreationTime()

Returns the time when this session was created, measured in
milliseconds since midnight January 1, 1970 GMT.

**Returns:**

A `long` specifying when this session was created, expressed in milliseconds since 1/1/1970 GMT

**Throws:**

`IllegalStateException` - if this method is called on an invalidated session

---

```java
public String getId()
```

**getld**

```java
String getId()
```

Returns a string containing the unique identifier assigned to this session. The identifier is assigned by the servlet container and is implementation dependent.

**Returns:**

A string specifying the identifier assigned to this session

---

```java
public long getLastAccessedTime()
```

**getLastAccessedTime**

```java
long getLastAccessedTime()
```

Returns: A `long` specifying when this session was last accessed, expressed in milliseconds since 1/1/1970 GMT

**Throws**

`IllegalStateException`
long **getLastAccessedTime()**

Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container received the request.

Actions that your application takes, such as getting or setting a value associated with the session, do not affect the access time.

**Returns:**
- a `long` representing the last time the client sent a request associated with this session, expressed in milliseconds since 1/1/1970 GMT

**Throws:**
- `IllegalStateException` - if this method is called on an invalidated session

---

```java
public **ServletContext** **getServletContext()**

**ServletContext**

```return``` Web **ServletContext**

since 2.3
```

**getServletContext**

**ServletContext** **getServletContext()**

Returns the ServletContext to which this session belongs.

**Returns:**
- The ServletContext object for the web application

**Since:**
- 2.3

---

```java
public **void** **setMaxInactiveInterval**(int interval)
```

**setMaxInactiveInterval(int interval)**
setMaxInactiveInterval

void setMaxInactiveInterval(int interval)

Specifies the time, in seconds, between client requests before the servlet container will invalidate this session. A negative time indicates the session should never timeout.

Parameters:
  interval - An integer specifying the number of seconds

getMaxInactiveInterval

public int getMaxInactiveInterval()

Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses. After this interval, the servlet container will invalidate the session. The maximum time interval can be set with the setMaxInactiveInterval method. A negative time indicates the session should never timeout.

Returns:
  an integer specifying the number of seconds this session remains open between client requests

See Also:
  setMaxInactiveInterval(int)
public HttpSessionContext getSessionContext()

Deprecated. As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.

public Object getAttribute(String name)

name
return

Throws

IllegalStateException: 

getAttribute

Object getAttribute(String name)

Returns the object bound with the specified name in this session, or null if no object is bound under the name.

Parameters:

name - a string specifying the name of the object

Returns:

the object with the specified name

Throws:

IllegalStateException - if this method is called on an invalidated session
public Object getValue(String name)

Deprecated. As of Version 2.2, this method is replaced by
getAttribute(java.lang.String).

Parameters:
   name - a string specifying the name of the object

Returns:
   the object with the specified name

Throws:
   IllegalAccessException - if this method is called on an
   invalidated session

public java.util.Enumeration&lt;E&gt; getAttributeNames()

return String Enumeration

Throws IllegalAccessException:

getAttributeNames

Enumeration getAttributeNames()

Returns an Enumeration of String objects containing the names of all
the objects bound to this session.

Returns:
   an Enumeration of String objects specifying the names of all the
objects bound to this session

Throws:
IllegalStateException - if this method is called on an invalidated session

public String[] getValueNames()

return

Throws
IllegalStateException:

getValueNames

String[] getValueNames()

Deprecated. As of Version 2.2, this method is replaced by getAttributeNames()

Returns:
an array of String objects specifying the names of all the objects bound to this session

Throws:
IllegalStateException - if this method is called on an invalidated session

public void setAttribute(String name, Object value)

setTextAttribute

setAttribute

String

setAttribute

IllegalStateException

setAttribute

IllegalStateException - if this method is called on an invalidated session
setAttribute

void `setAttribute(String name, Object value)`

Binds an object to this session, using the name specified. If an object of the same name is already bound to the session, the object is replaced.

After this method executes, and if the new object implements `HttpSessionBindingListener`, the container calls `HttpSessionBindingListener.valueBound`. The container then notifies any `HttpSessionAttributeListener`s in the web application.

If an object was already bound to this session of this name that implements `HttpSessionBindingListener`, its `HttpSessionBindingListener.valueUnbound` method is called.

If the value passed in is null, this has the same effect as calling `removeAttribute()`.

**Parameters:**

- `name` - the name to which the object is bound; cannot be null
- `value` - the object to be bound

**Throws:**

- `IllegalStateException` - if this method is called on an invalidated session

public void `putValue(String name, Object value)`

deprecated 2.2

`#setAttribute`
**putValue**

```java
void putValue(String name, Object value)
```

**Deprecated. As of Version 2.2, this method is replaced by setAttribute(java.lang.String, java.lang.Object)**

**Parameters:**
- `name` - the name to which the object is bound; cannot be null
- `value` - the object to be bound; cannot be null

**Throws:**
- `IllegalStateException` - if this method is called on an invalidated session

---

**public void removeAttribute(String name)**

```java
HttpSessionBindingListener
HttpSessionBindingListener.valueUnbound Web
HttpSessionAttributeListener
name
```

**Throws**
- `IllegalStateException`: null

---

**removeAttribute**

```java
void removeAttribute(String name)
```

Removes the object bound with the specified name from this session. If the session does not have an object bound with the specified name, this method does nothing.
After this method executes, and if the object implements HttpSessionBindingListener, the container calls HttpSessionBindingListener.valueUnbound. The container then notifies any HttpSessionAttributeListeners in the web application.

**Parameters:**
- name - the name of the object to remove from this session

**Throws:**
- `IllegalStateException` - if this method is called on an invalidated session

---

```java
public void removeValue(String name)
```

Deprecated: As of Version 2.2, this method is replaced by `removeAttribute(java.lang.String)`

**Parameters:**
- name - the name of the object to remove from this session

**Throws:**
- `IllegalStateException` - if this method is called on an invalidated session

---

```java
public void invalidate()
```

**Throws:**
- `IllegalStateException`
void invalidate()

Invalidates this session then unbinds any objects bound to it.

**Throws:**
- `IllegalStateException` - if this method is called on an already invalidated session

---

**public boolean isNew()**

public boolean isNew()

return true

Throws

IllegalStateException:

---

**isNew**

boolean isNew()

returns true if the client does not yet know about the session or if the client chooses not to join the session. For example, if the server used only cookie-based sessions, and the client had disabled the use of cookies, then a session would be new on each request.

**Returns:**
- true if the server has created a session, but the client has not yet joined

**Throws:**
- `IllegalStateException` - if this method is called on an already invalidated session

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
PS:
Interface

HttpSessionActivationListener

All Superinterfaces:
  EventListener

public interface HttpSessionActivationListener
extends EventListener

Implements: java.util.EventListener

VM HttpSessionActivationListener

   since 2.3

Objects that are bound to a session may listen to container events notifying them that sessions will be passivated and that session will be activated. A container that migrates session between VMs or persists sessions is required to notify all attributes bound to sessions implementing HttpSessionActivationListener.

Since:
   2.3

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void sessionDidActivate(HttpSessionEvent se)</td>
</tr>
<tr>
<td>Notification that the session has just been activated.</td>
</tr>
<tr>
<td>void sessionWillPassivate(HttpSessionEvent se)</td>
</tr>
<tr>
<td>Notification that the session is about to be passivated.</td>
</tr>
</tbody>
</table>

Method Detail
public void sessionWillPassivate( HttpSessionEvent se )

sessionWillPassivate

void sessionWillPassivate( HttpSessionEvent se )

  Notification that the session is about to be passivated.

_____________________________________________________________________

public void sessionDidActivate( HttpSessionEvent se )

sessionDidActivate

void sessionDidActivate( HttpSessionEvent se )

  Notification that the session has just been activated.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.servlet.http Interface HttpSessionAttributeListener

All Superinterfaces:
   EventListener

public interface HttpSessionAttributeListener
extends EventListener

Implements: java.util.EventListener

Web
   since v 2.3

This listener interface can be implemented in order to get notifications of changes to the attribute lists of sessions within this web application.

Since:
   v 2.3

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void attributeAdded(HttpSessionBindingEvent se)</td>
</tr>
<tr>
<td>Notification that an attribute has been added to a session.</td>
</tr>
<tr>
<td>void attributeRemoved(HttpSessionBindingEvent se)</td>
</tr>
<tr>
<td>Notification that an attribute has been removed from a session.</td>
</tr>
<tr>
<td>void attributeReplaced(HttpSessionBindingEvent se)</td>
</tr>
<tr>
<td>Notification that an attribute has been replaced in a session.</td>
</tr>
</tbody>
</table>

Method Detail
public void attributeAdded(HttpSessionBindingEvent se)

attributeAdded

void attributeAdded(HttpSessionBindingEvent se)

    Notification that an attribute has been added to a session. Called after the attribute is added.

public void attributeRemoved(HttpSessionBindingEvent se)

attributeRemoved

void attributeRemoved(HttpSessionBindingEvent se)

    Notification that an attribute has been removed from a session. Called after the attribute is removed.

public void attributeReplaced(HttpSessionBindingEvent se)

attributeReplaced

void attributeReplaced(HttpSessionBindingEvent se)

    Notification that an attribute has been replaced in a session. Called after the attribute is replaced.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.servlet.http

Class HttpSessionBindingEvent

java.lang.Object
  ▼ java.util.EventObject
    ▼ javax.servlet.http.HttpSessionEvent
        ▼ javax.servlet.http.HttpSessionBindingEvent

All Implemented Interfaces:
  Serializable

public class HttpSessionBindingEvent
  extends HttpSessionEvent

Extends: java.util.EventObject > HttpSessionEvent

HttpSession.setAttribute    HttpSession.removeAttribute

See also
javax.servlet.http.HttpSession,
javax.servlet.http.HttpSessionBindingListener,
javax.servlet.http.HttpSessionAttributeListener

Events of this type are either sent to an object that implements
HttpSessionBindingListener when it is bound or unbound from a session,
or to a HttpSessionAttributeListener that has been configured in the
deployment descriptor when any attribute is bound, unbound or replaced
in a session.

The session binds the object by a call to HttpSession.setAttribute and
unbinds the object by a call to HttpSession.removeAttribute.

Author:
  Various

See Also:
  HttpSession, HttpSessionBindingListener,
### Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class java.util.EventObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

### Constructor Summary

#### HttpSessionBindingEvent

<table>
<thead>
<tr>
<th>HttpSessionBindingEvent(session, String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs an event that notifies an object that it has been bound to or unbound from a session.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HttpSessionBindingEvent(session, String name, Object value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs an event that notifies an object that it has been bound to or unbound from a session.</td>
</tr>
</tbody>
</table>

### Method Summary

#### String

<table>
<thead>
<tr>
<th>getName()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the name with which the attribute is bound to or unbound from the session.</td>
</tr>
</tbody>
</table>

#### HttpSession

<table>
<thead>
<tr>
<th>getSession()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the session that changed.</td>
</tr>
</tbody>
</table>

#### Object

<table>
<thead>
<tr>
<th>getValue()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the value of the attribute that has been added, removed or replaced.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.util.EventObject

getSource, toString

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
Constructor Detail

public HttpSessionBindingEvent(HttpSession session, String name)

session
ame
See also
getName, getSession

HttpSessionBindingEvent

public HttpSessionBindingEvent(HttpSession session, String name)

Constructs an event that notifies an object that it has been bound to or unbound from a session. To receive the event, the object must implement HttpSessionBindingListener.

Parameters:
  session - the session to which the object is bound or unbound
  name - the name with which the object is bound or unbound

See Also:
  getName(), getSession()

public HttpSessionBindingEvent(HttpSession session, String name, Object value)

session
name
See also
getName, getSession
HttpSessionBindingEvent

public HttpSessionBindingEvent(HttpSession session, String name, Object value)

Constructs an event that notifies an object that it has been bound to or unbound from a session. To receive the event, the object must implement HttpSessionBindingListener.

Parameters:
   session - the session to which the object is bound or unbound
   name - the name with which the object is bound or unbound

See Also:
   getName(), getSession()

Method Detail

public HttpSession getSession()

getSession

public HttpSession getSession()

Return the session that changed.

Overrides:
   getSession in class HttpSessionEvent

public String getName()

return
getName

public String getName()

    Returns the name with which the attribute is bound to or unbound from the session.

    Returns:
    a string specifying the name with which the object is bound to or unbound from the session

public Object getValue()

    since 2.3

getValue

public Object getValue()

    Returns the value of the attribute that has been added, removed or replaced. If the attribute was added (or bound), this is the value of the attribute. If the attribute was removed (or unbound), this is the value of the removed attribute. If the attribute was replaced, this is the old value of the attribute.

    Since:
    2.3
PS:
javax.servlet.http

**Interface HttpSessionBindingListener**

**All Superinterfaces:**

EventListener

---

```java
public interface HttpSessionBindingListener
extends EventListener

Implements: java.util.EventListener
```

**See also**

javax.servlet.http.HttpSession,
javax.servlet.http.HttpSessionBindingEvent

---

Causes an object to be notified when it is bound to or unbound from a session. The object is notified by an HttpSessionBindingEvent object. This may be as a result of a servlet programmer explicitly unbinding an attribute from a session, due to a session being invalidated, or due to a session timing out.

**Author:**

Various

**See Also:**

 HttpSession, HttpSessionBindingEvent

---

<table>
<thead>
<tr>
<th><strong>Method Summary</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>valueBound</strong></td>
<td>(HttpSessionBindingEvent event)</td>
</tr>
<tr>
<td>void</td>
<td>Notifies the object that it is being bound to a session and identifies the session.</td>
</tr>
<tr>
<td><strong>valueUnbound</strong></td>
<td>(HttpSessionBindingEvent event)</td>
</tr>
<tr>
<td>void</td>
<td>Notifies the object that it is being unbound from a session and identifies the session.</td>
</tr>
</tbody>
</table>
public void valueBound(HttpSessionBindingEvent event)

   event
See also

valueBound

void valueBound(HttpSessionBindingEvent event)

   Notifies the object that it is being bound to a session and identifies the session.

Parameters:
   event - the event that identifies the session

See Also:
   valueUnbound(javax.servlet.http.HttpSessionBindingEvent)

public void valueUnbound(HttpSessionBindingEvent event)

   event
See also

valueUnbound

void valueUnbound(HttpSessionBindingEvent event)

   Notifies the object that it is being unbound from a session and identifies the session.
Parameters:
  event - the event that identifies the session

See Also:
  `valueBound(javax.servlet.http.HttpSessionBindingEvent)`

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
### Interface HttpSessionContext

**Deprecated.** As of Java(tm) Servlet API 2.1 for security reasons, with no replacement. This interface will be removed in a future version of this API.

**deprecated**

As of Java(tm) Servlet API 2.1 for security reasons, with no replacement. This interface will be removed in a future version of this API.

**See also**

[javax.servlet.http.HttpSession],
[javax.servlet.http.HttpSessionBindingEvent],
[javax.servlet.http.HttpSessionBindingListener]

---

**public interface HttpSessionContext**

**Author:** Various

**See Also:**

[HttpSession], [ HttpSessionBindingEvent], [ HttpSessionBindingListener]

---

### Method Summary

<table>
<thead>
<tr>
<th>Enumeration</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>getIds()</strong></td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.1 with no replacement. This method must return an empty Enumeration and will be removed in a future version of this API.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>HttpSession</th>
<th>Method (String sessionId)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>getSession (String sessionId)</strong></td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.1 with no replacement. This method must return null and will be removed in a future version of this API.</td>
</tr>
</tbody>
</table>
public HttpSession getSession(String sessionId)

deprecated  Java Servlet API 2.1  null API

getSession

HttpSession getSession(String sessionId)

Deprecated. As of Java Servlet API 2.1 with no replacement. This method must return null and will be removed in a future version of this API.

public java.util.Enumeration<E> getIds()

deprecated  Java Servlet API 2.1

getIds

Enumeration getIds()

Deprecated. As of Java Servlet API 2.1 with no replacement. This method must return an empty Enumeration and will be removed in a future version of this API.
PS:
javax.servlet.http

Class HttpSessionEvent

java.lang.Object
   └ java.util.EventObject
       └ javax.servlet.http.HttpSessionEvent

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   HttpSessionBindingEvent

public class HttpSessionEvent
extends EventObject

Extends: java.util.EventObject
Extended by: HttpSessionBindingEvent

Web
   since v 2.3

This is the class representing event notifications for changes to sessions within a web application.

Since:
   v 2.3
See Also:
   Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
source


Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HttpSessionEvent(HttpSession source)</td>
<td>Construct a session event from the given source.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HttpSession getSession()</td>
<td>Return the session that changed.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.util.EventObject

generate, getSource, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

public HttpSessionEvent(HttpSession source)

HttpSessionEvent

public HttpSessionEvent(HttpSession source)

    Construct a session event from the given source.

Method Detail

public HttpSession getSession()
getSession

public HttpSession getSession()

Return the session that changed.
| Overview | Package | Tree | Deprecated | Index | Help | PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
javax.servlet.http Interface HttpSessionListener

All Superinterfaces:
   EventListener

public interface HttpSessionListener
   extends EventListener

Implements: java.util.EventListener

Web		Web
   since v 2.3 en
See also javax.servlet.http.HttpSessionEvent

Implementations of this interface are notified of changes to the list of active sessions in a web application. To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

Since:
   v 2.3
See Also:
   HttpSessionEvent

---

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong> [sessionCreated](HttpSessionEvent se)</td>
</tr>
<tr>
<td>Notification that a session was created.</td>
</tr>
<tr>
<td><strong>void</strong> [sessionDestroyed](HttpSessionEvent se)</td>
</tr>
<tr>
<td>Notification that a session is about to be invalidated.</td>
</tr>
</tbody>
</table>
Method Detail

public void sessionCreated(HttpSessionEvent se)

  se

sessionCreated

void sessionCreated(HttpSessionEvent se)

  Notification that a session was created.

  Parameters:
  se - the notification event

public void sessionDestroyed(HttpSessionEvent se)

  se

sessionDestroyed

void sessionDestroyed(HttpSessionEvent se)

  Notification that a session is about to be invalidated.

  Parameters:
  se - the notification event
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet.http  Class HttpUtils

java.lang.Object
├── javax.servlet.http.HttpUtils

**Deprecated.** As of Java(tm) Servlet API 2.3. These methods were only useful with the default encoding and have been moved to the request interfaces.

**deprecated**    Java(tm) Servlet API 2.3

```java
public class HttpUtils
extends Object
```

### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HttpUtils()</code></td>
<td><strong>Deprecated.</strong> Constructs an empty <code>HttpUtils</code> object.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>StringBuffer getRequestURL(HttpServletRequest req)</code></td>
<td><strong>Deprecated.</strong> Reconstructs the URL the client used to make the request, using information in the <code>HttpServletRequest</code> object.</td>
</tr>
<tr>
<td><code>Hashtable parsePostData(int len, ServletInputStream in)</code></td>
<td><strong>Deprecated.</strong> Parses data from an HTML form that the client sends to the server using the HTTP POST method and the <code>application/x-www-form-urlencoded</code> MIME type.</td>
</tr>
<tr>
<td><code>Hashtable parseQueryString(String s)</code></td>
<td><strong>Deprecated.</strong> Parses a query string passed from the client to the server and builds a <code>HashTable</code> object</td>
</tr>
</tbody>
</table>
with key-value pairs.

Methods inherited from class java.lang.Object:
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public HttpUtils()
    HttpUtils

HttpUtils

public HttpUtils()
    Deprecated.
    Constructs an empty HttpUtils object.

Method Detail

public static java.util.Hashtable<K, V> parseQueryString(String s)
    -        HashTable
          GE

+ %xx ASCII

S
return
Throws
IllegalArgumentException:
parseQueryString

public static Hashtable parseQueryString(String s)

Deprecated.
 Parses a query string passed from the client to the server and builds
a HashTable object with key-value pairs. The query string should be
in the form of a string packaged by the GET or POST method, that
is, it should have key-value pairs in the form key=value, with each
pair separated from the next by a & character.

A key can appear more than once in the query string with different
values. However, the key appears only once in the hashtable, with its
value being an array of strings containing the multiple values sent by
the query string.

The keys and values in the hashtable are stored in their decoded
form, so any + characters are converted to spaces, and characters
sent in hexadecimal notation (like %xx) are converted to ASCII
characters.

Parameters:
   s - a string containing the query to be parsed

Returns:
   a HashTable object built from the parsed key-value pairs

Throws:
   IllegalArgumentException - if the query string is invalid

parsePostData

public static java.util.Hashtable<K, V> parsePostData(int len,
            ServletInputStream in)

HTML  HTTP POST  application/x-www-urlencoded MIME

POST - POST  POST
parsePostData

public static Hashtable parsePostData(int len, ServletInputStream in)

Deprecated.

Parses data from an HTML form that the client sends to the server using the HTTP POST method and the application/x-www-form-urlencoded MIME type.

The data sent by the POST method contains key-value pairs. A key can appear more than once in the POST data with different values. However, the key appears only once in the hashtable, with its value being an array of strings containing the multiple values sent by the POST method.

The keys and values in the hashtable are stored in their decoded form, so any + characters are converted to spaces, and characters sent in hexadecimal notation (like %xx) are converted to ASCII characters.

Parameters:
  - len - an integer specifying the length, in characters, of the ServletInputStream object that is also passed to this method
  - in - the ServletInputStream object that contains the data sent from the client

Returns:
  - a HashTable object built from the parsed key-value pairs

Throws:
  - IllegalArgumentException - if the data sent by the POST method is invalid
public static StringBuffer getRequestURL(HttpServletRequest req)

HttpServletRequest URL URL

StringBuffer URL

getRequestURL

public static StringBuffer getRequestURL(HttpServletRequest req)

Deprecated.
Reconstructs the URL the client used to make the request, using information in the HttpServletRequest object. The returned URL contains a protocol, server name, port number, and server path, but it does not include query string parameters.

Because this method returns a StringBuffer, not a string, you can modify the URL easily, for example, to append query parameters.

This method is useful for creating redirect messages and for reporting errors.

Parameters:
req - a HttpServletRequest object containing the client's request

Returns:
a StringBuffer object containing the reconstructed URL
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type Id

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Id

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@Id
public Long getId() { return id; }

since Java Persistence 1.0

Specifies the primary key property or field of an entity.

Example:

@Id
public Long getId() { return id; }

Since: Java Persistence 1.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
@javax.persistence.Annotation Type IdClass

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface IdClass

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

@IdClass(com.acme.EmployeePK.class)
@Entity
public class Employee {
  @Id String empName;
  @Id Date birthDay;
  ...
}

**since** Java Persistence 1.0

Specifies a composite primary key class that is mapped to multiple fields or properties of the entity.

The names of the fields or properties in the primary key class and the primary key fields or properties of the entity must correspond and their types must be the same.

Example:

@IdClass(com.acme.EmployeePK.class)
@Entity
public class Employee {
  @Id String empName;
Abstract public Class<T> value()

value

Public abstract Class value

Primary key class
javax.jms Class IllegalStateException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
       ↓ javax.jms.JMSException
          ↓ javax.jms.IllegalStateException

All Implemented Interfaces:
   Serializable

public class IllegalStateException
extends JMSException

Extends: Throwable > Exception > JMSException

version April 9, 2002

This exception is thrown when a method is invoked at an illegal or inappropriate time or if the provider is not in an appropriate state for the requested operation. For example, this exception must be thrown if Session.commit is called on a non-transacted session. This exception is also called when a domain inappropriate method is called, such as calling TopicSession.CreateQueueBrowser.

Version: April 9, 2002

Author: Rahul Sharma, Kate Stout

See Also:
   Serialized Form
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method Details</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IllegalStateException(String reason)</code></td>
<td>Constructs an <code>IllegalStateException</code> with the specified reason.</td>
</tr>
<tr>
<td><code>IllegalStateException(String reason, String errorCode)</code></td>
<td>Constructs an <code>IllegalStateException</code> with the specified reason and error code.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class javax.jms.JMSException</th>
<th>Methods inherited from class java.lang.Throwable</th>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getConfiguredException</code>, <code>getLinkedException</code>, <code>setLinkedException</code></td>
<td><code>fillInStackTrace</code>, <code>getCause</code>, <code>getMessage</code>, <code>getLocalizedMessage</code>, <code>getStackTrace</code>, <code>initCause</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>setStackTrace</code>, <code>toString</code></td>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
</tr>
</tbody>
</table>

## Constructor Detail

public `IllegalStateException(String reason, String errorCode)`

```java
IllegalStateException reason
errorCode

IllegalStateException
```

IllegalStateException
public IllegalStateException(String reason, String errorCode)

Constructs an IllegalStateException with the specified reason and error code.

Parameters:
reason - a description of the exception
errorCode - a string specifying the vendor-specific error code

public IllegalStateException(String reason)

IllegalStateException

public IllegalStateException(String reason)

Constructs an IllegalStateException with the specified reason. The error code defaults to null.

Parameters:
reason - a description of the exception

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi  Class IllegalStateException

java.lang.Object  
  ▼ java.lang.Throwable  
    ▼ java.lang.Exception  
      ▼ javax.resource.ResourceException  
        ▼ javax.resource.spi.IllegalStateException

All Implemented Interfaces:
  Serializable

public class IllegalStateException
extends ResourceException

Extends: Throwable > Exception > ResourceException

version 1.0

An IllegalStateException is thrown from a method if the callee (resource adapter or application server for system contracts) is in an illegal or inappropriate state for the method invocation.

Version: 1.0
Author: Rahul Sharma, Ram Jeyaraman
See Also: Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>IllegalStateException()</td>
</tr>
<tr>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td>IllegalStateException(String message)</td>
</tr>
</tbody>
</table>
Constructs a new instance with the specified detail message.

**IllegalStateException**(String message, String errorCode)

- Constructs a new throwable with the specified detail message and an error code.

**IllegalStateException**(String message, Throwable cause)

- Constructs a new throwable with the specified detail message and cause.

**IllegalStateException**(Throwable cause)

- Constructs a new throwable with the specified cause.

### Method Summary

Methods inherited from class javax.resource.ResourceException

- [getErrorCode](#)
- [getLinkedException](#)
- [getMessage](#)
- [setErrorCode](#)
- [setLinkedException](#)

Methods inherited from class java.lang Throwable

- [fillInStackTrace](#)
- [getCause](#)
- [getLocalizedMessage](#)
- [getStackTrace](#)
- [initCause](#)
- [printStackTrace](#)
- [printStackTrace](#)
- [printStackTrace](#)
- [setStackTrace](#)
- [toString](#)

Methods inherited from class java.lang Object

- [clone](#)
- [equals](#)
- [finalize](#)
- [getClass](#)
- [hashCode](#)
- [notify](#)
- [notifyAll](#)
- [wait](#)
- [wait](#)
- [wait](#)

### Constructor Detail

public IllegalStateException()

null

**IllegalStateException**

public IllegalStateException()
Constructs a new instance with null as its detail message.

```java
public IllegalStateException(String message)

message
```

**IllegalStateException**

```java
public IllegalStateException(String message)

message
```

Constructs a new instance with the specified detail message.

**Parameters:**

- `message`: the detail message.

```java
public IllegalStateException(Throwable cause)

cause
```

```java
cause
```

**IllegalStateException**

```java
public IllegalStateException(Throwable cause)

cause
```

Constructs a new throwable with the specified cause.

**Parameters:**

- `cause`: a chained exception of type `Throwable`.

```java
public IllegalStateException(String message, Throwable cause)

message
```

```java
cause
```
IllegalStateException

public IllegalStateException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public IllegalStateException(String message, String errorCode)
throwable

message
errorCode

IllegalStateException

public IllegalStateException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail  **Class IllegalWriteException**

```java
java.lang.Object
   └ java.lang.Throwable
      └ java.lang.Exception
         └ javax.mail.MessagingException
            └ javax.mail.IllegalWriteException
```

All Implemented Interfaces:
   Serializable

---

public class **IllegalWriteException**

extends **MessagingException**

**Extends:** Throwable > Exception > MessagingException

Messaging

The exception thrown when a write is attempted on a read-only attribute of any Messaging object.

**Author:**
   John Mani

**See Also:**
  Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IllegalWriteException()</strong></td>
<td>Constructs a IllegalWriteException with no detail message.</td>
</tr>
<tr>
<td><strong>IllegalWriteException(String s)</strong></td>
<td>Constructs a IllegalWriteException with the specified detail message.</td>
</tr>
</tbody>
</table>
Method Summary

Methods inherited from class javax.mail.

getCause, getNextException, setNextException, toString

Methods inherited from class java.lang.

fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace

Methods inherited from class java.lang.

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public IllegalWriteException()

IllegalWriteException

IllegalWriteException

public IllegalWriteException()

Constructs a IllegalWriteException with no detail message.

public IllegalWriteException(String s)

IllegalWriteException

s

IllegalWriteException
public IllegalWriteException(String s)

Constructs a IllegalWriteException with the specified detail message.

**Parameters:**
- s - the detail message
public class ImplicitObjectELResolver
extends ELResolver

Defines variable resolution behavior for the EL implicit objects defined in the JSP specification.

The following variables are resolved by this ELResolver, as per the JSP specification:

- `pageContext` - the `PageContext` object.
- `pageScope` - a Map that maps page-scoped attribute names to their values.
- `requestScope` - a Map that maps request-scoped attribute names to their values.
- `sessionScope` - a Map that maps session-scoped attribute names to their values.
- `applicationScope` - a Map that maps application-scoped attribute names to their values.
- `param` - a Map that maps parameter names to a single String parameter value (obtained by calling `ServletRequest.getParameter(String name)`).
- `paramValues` - a Map that maps parameter names to a String[] of all values for that parameter (obtained by calling `ServletRequest.getParameterValues(String name)`).
- `header` - a Map that maps header names to a single String header value (obtained by calling `HttpServletRequest.getHeader(String name)`).
- `headerValues` - a Map that maps header names to a String[] of all values for that header (obtained by calling `HttpServletRequest.getHeaders(String)`).
- `cookie` - a Map that maps cookie names to a single `Cookie` object.
Cookies are retrieved according to the semantics of HttpServletRequest.getCookies(). If the same name is shared by multiple cookies, an implementation must use the first one encountered in the array of Cookie objects returned by the get_cookies() method. However, users of the cookie implicit object must be aware that the ordering of cookies is currently unspecified in the servlet specification.

- initParam - a Map that maps context initialization parameter names to their String parameter value (obtained by calling ServletContext.getInitParameter(String name)).

Since:
- JSP 2.1

See Also:
- ELResolver

### Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.el.ELResolver</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESOLVABLE_AT_DESIGN_TIME, TYPE</td>
</tr>
</tbody>
</table>

### Constructor Summary

| ImplicitObjectELResolver() |

### Method Summary

<table>
<thead>
<tr>
<th>Class&lt;String&gt; getCommonPropertyType(ELContext context, Object base)</th>
<th>If the base object is null, returns String.class.</th>
</tr>
</thead>
<tbody>
<tr>
<td>getFeatureDescriptors(ELContext context, Object base)</td>
<td>If the base object is null, and the property matches the name of a JSP implicit object,</td>
</tr>
</tbody>
</table>
returns an `Iterator` containing `FeatureDescriptor` objects with information about each JSP implicit object resolved by this resolver.

### `getType` Method (Class `object`)

```java
public ELContext getType(ELContext context, Object base, Object property)
```

- If the base object is `null`, and the property matches the name of a JSP implicit object, returns `null` to indicate that no types are ever accepted to `setValue()`.

### `getValue` Method (Object `object`)

```java
public Object getValue(ELContext context, Object base, Object property)
```

- If the base object is `null`, and the property matches the name of a JSP implicit object, returns the implicit object.

### `isReadOnly` Method (Boolean `object`)

```java
public boolean isReadOnly(ELContext context, Object base, Object property)
```

- If the base object is `null`, and the property matches the name of a JSP implicit object, returns `true` to indicate that implicit objects cannot be overwritten.

### `setValue` Method (Void `object`)

```java
public void setValue(ELContext context, Object base, Object property, Object val)
```

- If the base object is `null`, and the property matches the name of a JSP implicit object, throws `PropertyNotWritableException` to indicate that implicit objects cannot be overwritten.

---

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public ImplicitObjectELResolver()

**Method Detail**

**getValue**

public Object getValue(ELContext context, Object base, Object property)

If the base object is `null`, and the property matches the name of a JSP implicit object, returns the implicit object.

The `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver before returning if an implicit object is matched. If this property is not `true` after this method is called, the caller should ignore the return value.

**Specified by:**
getValue in class ELResolver

**Parameters:**
- context - The context of this evaluation.
- base - Only `null` is handled by this resolver. Other values will result in an immediate return.
- property - The name of the implicit object to resolve.

**Returns:**
- If the `propertyResolved` property of `ELContext` was set to `true`, then the implicit object; otherwise undefined.

**Throws:**
- NullPointerException - if context is `null`
- ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

**getType**

public Class getType(ELContext context, Object base,
If the base object is `null`, and the property matches the name of a JSP implicit object, returns `null` to indicate that no types are ever accepted to `setValue()`.

The `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver before returning if an implicit object is matched. If this property is not `true` after this method is called, the caller should ignore the return value.

Specified by:
`getType` in class `ELResolver`

Parameters:
- `context` - The context of this evaluation.
- `base` - Only `null` is handled by this resolver. Other values will result in an immediate return.
- `property` - The name of the implicit object to resolve.

Returns:
If the `propertyResolved` property of `ELContext` was set to `true`, then `null`; otherwise undefined.

Throws:
- `NullPointerException` - if `context` is `null`
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

**setValue**

```java
public void setValue(ELContext context,
                     Object base,
                     Object property,
                     Object val)
```

If the base object is `null`, and the property matches the name of a JSP implicit object, throws `PropertyNotWritableException` to indicate that implicit objects cannot be overwritten.

The `propertyResolved` property of the `ELContext` object must be set to
true by this resolver before returning if an implicit object is matched. If this property is not true after this method is called, the caller should ignore the return value.

**Specified by:**
- `setValue` in class `ELResolver`

**Parameters:**
- `context` - The context of this evaluation.
- `base` - Only null is handled by this resolver. Other values will result in an immediate return.
- `property` - The name of the implicit object.
- `val` - The value to be associated with the implicit object.

**Throws:**
- `NullPointerException` - if context is null.
- `PropertyNotWritableException` - always thrown, if the implicit object name is recognized by this resolver.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

**isReadOnly**

```java
public boolean isReadOnly(ELContext context,
                         Object base,
                         Object property)
```

If the base object is null, and the property matches the name of a JSP implicit object, returns `true` to indicate that implicit objects cannot be overwritten.

The `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver before returning if an implicit object is matched. If this property is not `true` after this method is called, the caller should ignore the return value.

**Specified by:**
- `isReadOnly` in class `ELResolver`

**Parameters:**
- `context` - The context of this evaluation.
base - Only null is handled by this resolver. Other values will result in an immediate return.

property - The name of the implicit object.

**Returns:**

If the `propertyResolved` property of `ELContext` was set to `true`, then `true`; otherwise undefined.

**Throws:**

- `NullPointerException` - if `context` is `null`.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

**getFeatureDescriptors**

```java
public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext context, Object base)
```

If the base object is `null`, and the property matches the name of a JSP implicit object, returns an `Iterator` containing `FeatureDescriptor` objects with information about each JSP implicit object resolved by this resolver. Otherwise, returns `null`.

The `Iterator` returned must contain one instance of `FeatureDescriptor` for each of the EL implicit objects defined by the JSP spec. Each info object contains information about a single implicit object, and is initialized as follows:

- **displayName** - The name of the implicit object.
- **name** - Same as `displayName` property.
- **shortDescription** - A suitable description for the implicit object. Will vary by implementation.
- **expert** - `false`
- **hidden** - `false`
- **preferred** - `true`

In addition, the following named attributes must be set in the returned `FeatureDescriptor`s:

- **ELResolver.TYPE** - The runtime type of the implicit object.
- **ELResolver.RESOLVABLE_AT_DESIGN_TIME** - `true`.
getFeatureDescriptors

Parameters:
- context - The context of this evaluation.
- base - Only null is handled by this resolver. Other values will result in a null return value.

Returns:
An Iterator containing one FeatureDescriptor object for each implicit object, or null if base is not null.

See Also:
FeatureDescriptor

getCommonPropertyType

public void getCommonPropertyType(ELContext context, Object base)

If the base object is null, returns String.class. Otherwise, returns null.

Specified by:
getCommonPropertyType in class ELResolver

Parameters:
- context - The context of this evaluation.
- base - Only null is handled by this resolver. Other values will result in a null return value.

Returns:
null if base is not null; otherwise String.class.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAME NO FRAMES DETAIL: FIELD | CONSTR | METHOD
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.resource.cci Interface IndexedRecord

All Superinterfaces:
   Cloneable, Collection, Iterable, List, Record, Serializable

public interface IndexedRecord
extends Record, List, Serializable

Implements: Record, java.util.List<E>, java.io.Serializable

IndexedRecord java.util.List Collection
   since 0.8

IndexedRecord represents an ordered collection of record elements based on the java.util.List interface. This interface allows a client to access elements by their integer index (position in the list) and search for elements in the List.

Since:
   0.8
Author:
   Rahul Sharma

Method Summary

Methods inherited from interface javax.resource.cci.Record
   clone, equals, getRecordName, getRecordShortDescription, hashCode,
   setRecordName, setRecordShortDescription

Methods inherited from interface java.util.List
   add, add, addAll, addAll, clear, contains, containsAll, equals,
   get, hashCode, indexOf, isEmpty, iterator, lastIndexOf,
   listIterator, listIterator, remove, remove, removeAll, retainAll,
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type Inheritance

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
public @interface Inheritance

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

@Target(value=TYPE)  
@Retention(value=RUNTIME)

@Entity  
@Inheritance(strategy=JOINED)  
public class Customer { ... }

@Entity  
public class ValuedCustomer extends Customer { ... }

**since**  
Java Persistence 1.0  

Defines the inheritance strategy to be used for an entity class hierarchy. It is specified on the entity class that is the root of the entity class hierarchy.

Example:

@Entity  
@Inheritance(strategy=JOINED)  
public class Customer { ... }

@Entity  
public class ValuedCustomer extends Customer { ... }

**Since:**  
Java Persistence 1.0
abstract public InheritanceType strategy()

**strategy**

public abstract InheritanceType strategy

The strategy to be used

**Default:**

SINGLE_TABLE
<table>
<thead>
<tr>
<th>Prev Class</th>
<th>Next Class</th>
<th>Frames</th>
<th>No Frames</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
</tr>
<tr>
<td>SUMMARY: NESTED</td>
<td></td>
<td>ENUM, CONSTANTS</td>
<td>FIELD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>METHOD</td>
<td></td>
</tr>
<tr>
<td>DETAIL: ENUM, CONSTANTS</td>
<td></td>
<td>FIELD</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
**Enum InheritanceType**

**java.lang.Object**
- **java.lang.Enum<InheritanceType>**
- **javax.persistence.InheritanceType**

**All Implemented Interfaces:**
- `Serializable`, `Comparable<InheritanceType>`

```
public enum InheritanceType
extends Enum<InheritanceType>
```

**Extends:** `Enum<E>`

**since** [Java Persistence 1.0](https://en)

Defines inheritance strategy options.

**Since:**
- Java Persistence 1.0

---

**Enum Constant Summary**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JOINED</strong></td>
<td>A strategy in which fields that are specific to a subclass are mapped to a separate table than the fields that are common to the parent class, and a join is performed to instantiate the subclass.</td>
</tr>
<tr>
<td><strong>SINGLE_TABLE</strong></td>
<td>A single table per class hierarchy</td>
</tr>
<tr>
<td><strong>TABLE_PER_CLASS</strong></td>
<td>A table per concrete entity class</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <code>InheritanceType.valueOf(String name)</code></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td>static <code>InheritanceType[] values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Enumeration

clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object

finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

SINGLE_TABLE

public static final `InheritanceType` SINGLE_TABLE

A single table per class hierarchy

---

TABLE_PER_CLASS

public static final `InheritanceType` TABLE_PER_CLASS

A table per concrete entity class

---
public static final InheritanceType JOINED

A strategy in which fields that are specific to a subclass are mapped to a separate table than the fields that are common to the parent class, and a join is performed to instantiate the subclass.

Method Detail

final public static InheritanceType[] values()

values

public static final InheritanceType[] values()

    Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

    for(InheritanceType c : InheritanceType.values())
        System.out.println(c);

    Returns:
    an array containing the constants of this enum type, in the order they're declared

    public static InheritanceType valueOf(String name)

valueOf

public static InheritanceType valueOf(String name)

    Returns the enum constant of this type with the specified name. The
string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**

name - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**

IllegalArgumentException - if this enum type has no constant with the specified name
javax.ejb Annotation Type Init

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface Init

Implements: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Bean Home Local Home create

Designates a method of a session bean that corresponds to the create method of an adapted Home interface or an adapted Local Home interface.

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>The name of the corresponding create method of the adapted Home/LocalHome interface.</td>
</tr>
</tbody>
</table>

abstract public String value()

Home/LocalHome create Home/LocalHome
@Init create create

value

public abstract String value

The name of the corresponding create method of the adapted Home/LocalHome interface. This value is used to disambiguate the case where there are multiple create methods on an adapted
Home/LocalHome interface with the same signature as the annotated @Init method. If no value is specified, the create matching is based on signature only.

Default: 

""

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jws.soap  Annotation Type InitParam

**Deprecated.** As of JSR-181 2.0 with no replacement.

**Implements:** Annotation

**Deprecated.**

```java
@Deprecated
class InitParam
```

An initialization parameter

**Author:**

Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

**Required Element Summary**

| String name | Deprecated. Name of the initialization parameter |
| String value | Deprecated. Value of the initialization parameter |

---

**Element Detail**

abstract public String name()
public abstract String name

   Deprecated.
   Name of the initialization parameter

abstract public String value()

value

public abstract String value

   Deprecated.
   Value of the initialization parameter
public abstract class IntegerComparisonTerm
extends ComparisonTerm

Extends: SearchTerm > ComparisonTerm
Extended by: MessageNumberTerm, SizeTerm

This class implements comparisons for integers.

Author: Bill Shannon, John Mani

See Also: Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>type</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>number</td>
</tr>
</tbody>
</table>

The number.

Fields inherited from class javax.mail.search.ComparisonTerm

comparison, EQ, GE, GT, LE, LT, NE
## Constructor Summary

<table>
<thead>
<tr>
<th>protected</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>IntegerComparisonTerm(int comparison, int number)</code></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(Object obj)</code></td>
</tr>
</tbody>
</table>

Equality comparison.

<table>
<thead>
<tr>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getComparison()</code></td>
</tr>
</tbody>
</table>

Return the type of comparison.

<table>
<thead>
<tr>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getNumber()</code></td>
</tr>
</tbody>
</table>

Return the number to compare with.

<table>
<thead>
<tr>
<th>int</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>hashCode()</code></td>
</tr>
</tbody>
</table>

Compute a hashCode for this object.

<table>
<thead>
<tr>
<th>protected boolean</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>match(int i)</code></td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.mail.search.SearchTerm`

- `match`

### Methods inherited from class `java.lang.Object`

- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
The number.

**Constructor Detail**

`protected IntegerComparisonTerm(int comparison, int number)`

**Method Detail**

`public int getNumber()`

*getNumber*

`public int getNumber()`

Return the number to compare with.

`public int getComparison()`

*getComparison*

`public int getComparison()`

Return the type of comparison.
protected boolean match(int i)

match

protected boolean match(int i)

public boolean equals(Object obj)

equals

public boolean equals(Object obj)

   Equality comparison.

   Overrides:
       equals in class ComparisonTerm

public int hashCode()

hashCode

public int hashCode()

   Compute a hashCode for this object.

   Overrides:
       hashCode in class ComparisonTerm
javax.faces.convert Class IntegerConverter

java.lang.Object
   \- javax.faces.convert.IntegerConverter

All Implemented Interfaces:
   Converter

public class IntegerConverter
extends Object
implements Converter

Implements: Converter

java.lang.Integer int Converter

Converter implementation for java.lang.Integer (and int primitive) values.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String Converter_ID</td>
</tr>
<tr>
<td>static String INTEGER_ID</td>
</tr>
<tr>
<td>static String STRING_ID</td>
</tr>
</tbody>
</table>

Constructor Summary
IntegerConverter()

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAsObject</strong></td>
<td><code>getAsObject(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><strong>getAsString</strong></td>
<td><code>getAsString(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

**CONVERTER_ID**

`public static final String CONVERTER_ID`

The standard converter id for this converter.

**See Also:**

[Constant Field Values](#)
The message identifier of the FacesMessage to be created if the conversion to Integer fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

The message identifier of the FacesMessage to be created if the conversion of the Integer value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values
public IntegerConverter()

IntegerConverter

public IntegerConverter()

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>public Object getAsObject(FacesContext context, UIComponent component, String value)</strong></td>
</tr>
<tr>
<td><strong>Throws</strong></td>
</tr>
<tr>
<td><strong>Throws</strong></td>
</tr>
</tbody>
</table>

getAsObject

**Description copied from interface: Converter**

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

**Specified by:** getAsObject in interface Converter

**Parameters:**
- context - FacesContext for the request being processed
- component - UIComponent with which this model object value is associated
- value - String value to be converted (may be null)

**Returns:**
- null if the value to convert is null, otherwise the result of the
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws converterException - if conversion cannot be successfully performed

Throws NullPointerException - if context or component is null

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by: getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
converterException - if conversion cannot be successfully performed
performed

`NullPointerException` - if context or component is null
javax.xml.rpc.holders  Class IntegerWrapperHolder

java.lang.Object  
   | javax.xml.rpc.holders.IntegerWrapperHolder

All Implemented Interfaces:
   Holder

public final class IntegerWrapperHolder
  extends Object
  implements Holder

Implements: Holder

Field Summary

| Integer | value |

Constructor Summary

IntegerWrapperHolder()  

IntegerWrapperHolder(Integer myint)

Method Summary

Methods inherited from class java.lang.Object: clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
### Field Detail

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>public Integer value</td>
</tr>
</tbody>
</table>

### Constructor Detail

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>public IntegerWrapperHolder()</td>
</tr>
</tbody>
</table>

```java
public IntegerWrapperHolder()
```

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>public IntegerWrapperHolder(Integer myint)</td>
</tr>
</tbody>
</table>

```java
public IntegerWrapperHolder(Integer myint)
```

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://example.com)
PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

| SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |
javax.resource.cci Interface Interaction

public interface Interaction

javax.resource.cci.Interaction EIS Interaction EIS

- Record Record InteractionSpec execute
  InteractionSpec EIS Record
- Record InteractionSpec execute InteractionSpec
  EIS Record

Interaction Connection Connection close Interaction
Interaction Connection

version 0.8
since 0.8
See also java.sql.ResultSet

The javax.resource.cci.Interaction enables a component to execute EIS functions. An Interaction instance supports the following ways of interacting with an EIS instance:

- execute method that takes an input Record, output Record and an InteractionSpec. This method executes the EIS function represented by the InteractionSpec and updates the output Record
- execute method that takes an input Record and an InteractionSpec. This method implementation executes the EIS function represented by the InteractionSpec and produces the output Record as a return value.

An Interaction instance is created from a Connection and is required to maintain its association with the Connection instance. The close method releases all resources maintained by the resource adapter for the Interaction. The close of an Interaction instance should not close the associated Connection instance.
Since: 0.8
Version: 0.8
Author: Rahul Sharma
See Also: ResultSet

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>clearWarnings()</td>
<td>Clears all the warning reported by this Interaction instance.</td>
</tr>
<tr>
<td>void</td>
<td>close()</td>
<td>Closes the current Interaction and release all the resources held for this instance by the resource adapter.</td>
</tr>
<tr>
<td>Record</td>
<td>execute(InteractionSpec ispec, Record input)</td>
<td>Executes an interaction represented by the InteractionSpec.</td>
</tr>
<tr>
<td>boolean</td>
<td>execute(InteractionSpec ispec, Record input, Record output)</td>
<td>Executes an interaction represented by the InteractionSpec.</td>
</tr>
<tr>
<td>Connection</td>
<td>getConnection()</td>
<td>Gets the Connection associated with the Interaction.</td>
</tr>
<tr>
<td>ResourceWarning</td>
<td>getWarnings()</td>
<td>Gets the first ResourceWarning from the chain of warnings associated with this Interaction instance.</td>
</tr>
</tbody>
</table>

### Method Detail

public void close() throws ResourceException
Interaction Interaction  Connection
**Interaction**

Throws [ResourceException](#): Interaction Interaction close

---

**close**

```java
void close() throws ResourceException
```

Closes the current Interaction and release all the resources held for this instance by the resource adapter. The close of an Interaction instance does not close the associated Connection instance. It is recommended that Interaction instances be closed explicitly to free any held resources.

**Throws:**

- [ResourceException](#) - Failed to close the Interaction instance. Invoking close on an already closed Interaction should also throw this exception.

---

**public Connection getConnection()**

**Returns:**

- Connection instance associated with the Interaction

---

**public boolean execute(InteractionSpec ispec, Record**
input, Record output) throws ResourceException

InteractionSpec Record Record

ispec EIS / InteractionSpec
input Record
output Record
return EIS Record true false

Throws:
- EIS
- InteractionSpec
- Record
- Interaction

throws NotSupportedException:

execute

boolean execute(InteractionSpec ispec,
                    Record input,
                    Record output)
    throws ResourceException

Executes an interaction represented by the InteractionSpec. This
form of invocation takes an input Record and updates the output
Record.

Parameters:
- ispec - InteractionSpec representing a target EIS data/function
  module
- input - Input Record
- output - Output Record

Returns:
- true if execution of the EIS function has been successful and
  output Record has been updated; false otherwise

Throws:
- ResourceException - Exception if execute operation fails.
Examples of error cases are:
- Resource adapter internal, EIS-specific or communication
error
• Invalid specification of an InteractionSpec, input or output record structure
• Errors in use of input or output Record
• Invalid connection associated with this Interaction

NotSupportedException - Operation not supported

public Record execute(InteractionSpec ispec, Record input) throws ResourceException

InteractionSpec Interaction Record

Record

execute

Record execute(InteractionSpec ispec, Record input)

throws ResourceException

Executes an interaction represented by the InteractionSpec. This form of invocation takes an input Record and returns an output Record if the execution of the Interaction has been successfull.

Parameters:
    ispec - InteractionSpec representing a target EIS data/function module
    input - Input Record
Returns:
output Record if execution of the EIS function has been successful; null otherwise

Throws:
ResourceException - Exception if execute operation fails.

Examples of error cases are:
- Resource adapter internal, EIS-specific or communication error
- Invalid specification of an InteractionSpec or input record structure
- Errors in use of input Record or creation of an output Record
- Invalid connection associated with this Interaction

NotSupportedException - Operation not supported

---

public ResourceWarning getWarnings() throws ResourceException

Interaction ResourceWarning

return ResourceWarning

Throws ResourceException: Interaction ResourceWarning

---

getWarnings

ResourceWarning getWarnings() throws ResourceException

Gets the first ResourceWarning from the chain of warnings associated with this Interaction instance.

Returns:
ResourceWarning at top of the warning chain

Throws:
ResourceException - Failed to get ResourceWarnings associated with Interaction
public void clearWarnings() throws ResourceException
Interaction Interaction getWarnings
null

Throws ResourceException: Interaction ResourceWarning

clearWarnings

void clearWarnings()
throws ResourceException

Clears all the warning reported by this Interaction instance. After a call to this method, the method getWarnings will return null until a new warning is reported for this Interaction.

Throws:
ResourceException - Failed to clear ResourceWarnings associated with Interaction
javax.resource.cci  Interface InteractionSpec

All Superinterfaces:
  Serializable

public interface InteractionSpec
extends Serializable

Implements: java.io.Serializable

InteractionSpec  EIS Interaction  InteractionSpec  EIS

CCI  InteractionSpec  EIS InteractionSpec

InteractionSpec  Java Bean

  • FunctionNameEIS
  • InteractionVerbEIS  SYNC_SENDSYNC_SEND_RECEIVE
  • ExecutionTimeoutInteraction  EIS

Interaction  ResultSet

  • FetchSize
  • FetchDirection
  • MaxFieldSize
  • ResultSetType
  • ResultSetConcurrency

InteractionSpec  CCI  EIS  CCI

InteractionSpec  JavaBean  InteractionSpec
InteractionSpec  java.io.Serializable

  version  0.8
An InteractionSpec holds properties for driving an Interaction with an EIS instance. An InteractionSpec is used by an Interaction to execute the specified function on an underlying EIS.

The CCI specification defines a set of standard properties for an InteractionSpec. An InteractionSpec implementation is not required to support a standard property if that property does not apply to its underlying EIS.

The InteractionSpec implementation class must provide getter and setter methods for each of its supported properties. The getter and setter methods convention should be based on the Java Beans design pattern.

The standard properties are as follows:

- FunctionName: name of an EIS function
- InteractionVerb: mode of interaction with an EIS instance: SYNC_SEND, SYNC_SEND_RECEIVE, SYNC_RECEIVE
- ExecutionTimeout: the number of milliseconds an Interaction will wait for an EIS to execute the specified function

The following standard properties are used to give hints to an Interaction instance about the ResultSet requirements:

- FetchSize
- FetchDirection
- MaxFieldSize
- ResultSetType
- ResultSetConcurrency

A CCI implementation can provide additional properties beyond that described in the InteractionSpec interface. Note that the format and type of the additional properties is specific to an EIS and is outside the scope of the CCI specification.
It is required that the InteractionSpec interface be implemented as a JavaBean for the toolability support. The properties on the InteractionSpec implementation class should be defined through the getter and setter methods pattern. An implementation class for InteractionSpec interface is required to implement the java.io.Serializable interface.

**Since:**
0.8

**Version:**
0.8

**Author:**
Rahul Sharma

**See Also:**
Interaction

### Field Summary

<table>
<thead>
<tr>
<th>static int</th>
<th>SYNC_RECEIVE</th>
<th>The execution of an Interaction results in a synchronous receive of an output Record.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>SYNC_SEND</td>
<td>Interaction Verb type: The execution of an Interaction does only a send to the target EIS instance.</td>
</tr>
<tr>
<td>static int</td>
<td>SYNC_SEND_RECEIVE</td>
<td>Interaction Verb type: The execution of an Interaction sends a request to the EIS instance and receives response synchronously.</td>
</tr>
</tbody>
</table>

### Field Detail

SYNC_SEND
static final int `SYNC_SEND`

Interaction Verb type: The execution of an Interaction does only a send to the target EIS instance. The input record is sent to the EIS instance without any synchronous response in terms of an output Record or ResultSet.

See Also:
Constant Field Values

---------

SYNC_SEND_RECEIVE

static final int `SYNC_SEND_RECEIVE`

Interaction Verb type: The execution of an Interaction sends a request to the EIS instance and receives response synchronously. The input record is sent to the EIS instance with the output received either as Record or CCIResultSet.

See Also:
Constant Field Values

---------

SYNC_RECEIVE

static final int `SYNC_RECEIVE`

The execution of an Interaction results in a synchronous receive of an output Record. An example is: a session bean gets a method invocation and it uses this SEND_RECEIVE form of interaction to retrieve messages that have been delivered to a message queue.

See Also:
Constant Field Values

---------
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.interceptor  Annotation Type Interceptors

@Target(value={TYPE, METHOD})
@Retention(value=RUNTIME)
public @interface Interceptors

**Implements:** Annotation
@Target(value={TYPE, METHOD})
@Retention(value=RUNTIME)

Interceptor

Declares an ordered list of interceptors for a class or method.

---

**Required Element Summary**

<table>
<thead>
<tr>
<th>Class[]</th>
<th>value</th>
</tr>
</thead>
</table>

**Element Detail**

abstract public Class<T>[] value()

value

public abstract Class[] value

---

Submit a bug or feature
javax.xml.registry.infomodel  Interface InternationalString

public interface InternationalString

Locale  String  String  String  InternationalString
LocalizedString  Collection  String  Locale
InternationalString  set/get  InternationalString  String

See also  javax.xml.registry.infomodel.LocalizedString

This interface represents a String that has been internationalized into several Locales. This interface is used as a replacement for the String type whenever a String attribute needs to be I18N capable. An instance of the InternationalString interface composes within it a Collection of LocalizedString instances, where each String is specific to a particular Locale. The InternationalString interface provides set/get methods for adding or getting locale specific String values for the InternationalString instance.

Author:
Farrukh S. Najmi

See Also:
LocalizedString

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addLocalizedString(LocalizedString localizedString)</td>
</tr>
<tr>
<td>Adds a LocalizedString to this object.</td>
</tr>
<tr>
<td>void addLocalizedStrings(Collection localizedStrings)</td>
</tr>
<tr>
<td>Adds many LocalizedStrings to this object.</td>
</tr>
<tr>
<td>LocalizedString getLocalizedString(Locale locale, String charsetName)</td>
</tr>
<tr>
<td>Gets the LocalizedString for the specified Locale and charsetName.</td>
</tr>
<tr>
<td>getLocalizedStrings()</td>
</tr>
</tbody>
</table>


**Collection**

Returns the LocalizedStrings associated with this object.

**String**

- **getValue()**
  
  Gets the String value for the Locale returned by `Locale.getDefault()`.

- **getValue(Locale locale)**
  
  Gets the String value for the specified Locale.

- **removeLocalizedString(LocalizedString localizedString)**
  
  Removes a LocalizedString from this object.

- **removeLocalizedStrings(Collection localizedStrings)**
  
  Removes specified LocalizedStrings to this object.

- **setValue(Locale locale, String value)**
  
  Sets the String value for the specified Locale.

- **setValue(String value)**
  
  Sets the String value for the Locale returned by `Locale.getDefault()`.

---

**Method Detail**

**public String getValue() throws JAXRException**

Locale.getDefault()  Locale  String

0

    return Locale.getDefault()  Locale  String

Throws JAXRException: JAXR

**getValue**

String getValue() throws JAXRException

Gets the String value for the Locale returned by `Locale.getDefault()`.
public String getValue(java.util.Locale locale) throws JAXRException
Locale String null

locale	Locale
return Locale String
Throws JAXRException: JAXR

getValue

String getValue(Locale locale)
throws JAXRException

Gets the String value for the specified Locale. Return null if no locale specific value is found for the specified locale.

Capability Level: 0

Parameters:
locale - The Locale for the desired value

Returns:
the String value for the specified Locale

Throws:
JAXRException - If the JAXR provider encounters an internal error
public void setValue(String value) throws JAXRException
Locale.getDefault()  Locale  String

Sets the String value for the Locale returned by Locale.getDefault().

**Capability Level:** 0

**Parameters:**
value - the String value for the Locale returned by Locale.getDefault()

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

-------------

public void setValue(java.util.Locale locale, String value)
throws JAXRException
Locale  String

locale
value

**Throws**  
JAXRException:  JAXR
setValue

void setValue(Locale locale, String value) throws JAXRException

Sets the String value for the specified Locale.

**Capability Level:** 0

**Parameters:**
- locale - The locale for this value being set
- value - The value being set

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

public void addLocalizedString(LocalizedString localizedString) throws JAXRException

LocalizedString

0

LocalizedString

Throws JAXRException: JAXR

addLocalizedString

void addLocalizedString(LocalizedString localizedString) throws JAXRException

Adds a LocalizedString to this object.

**Capability Level:** 0

**Parameters:**
- localizedString - the LocalizedString being added to this object
Throws:

`JAXRException` - If the JAXR provider encounters an internal error

```
public void addLocalizedStrings(java.util.Collection<E> localizedStrings) throws JAXRException
LocalizedString

@localizedStrings
LocalizedString Collection
@throws JAXRException: JAXR
```

`addLocalizedStrings`

```
void addLocalizedStrings(Collection localizedStrings)
throws JAXRException

Adds many LocalizedStrings to this object.

Capability Level: 0

Parameters:

`localizedStrings` - the Collection of LocalizedStrings being added to this object

Throws:

`JAXRException` - If the JAXR provider encounters an internal error
```

```
public void removeLocalizedString(LocalizedString localizedString) throws JAXRException
LocalizedString

0
```
removeLocalizedString

void removeLocalizedString(LocalizedString localizedString)
throws JAXRException

Removes a LocalizedString from this object.

Capability Level: 0

Parameters:
 localizedString - the LocalizedString being removed from this object

Throws:
 JAXRException - If the JAXR provider encounters an internal error

---

public void removeLocalizedStrings(java.util.Collection<E> localizedStrings) throws JAXRException

Removes specified LocalizedStrings to this object.
public LocalizedString getLocalizedString(java.util.Locale locale, String charsetName) throws JAXRException

Locale charsetName LocalizedString

Throws: JAXRException: JAXR

getLocalizedString

LocalizedString getLocalizedString(Locale locale, String charsetName) throws JAXRException

Gets the LocalizedString for the specified Locale and charsetName.

Capability Level: 0

Parameters:
locale - The locale for the desired LocalizedStrings
charsetName - The character set name for the desired LocalizedStrings

Returns:
the LocalizedString that matches specified locale and character
Throws: `JAXRException` - If the JAXR provider encounters an internal error.

```java
public java.util.Collection<E> getLocalizedStrings() throws JAXRException
LocalizedString

0

return LocalizedString CollectionCollection null
Throws `JAXRException`: JAXR
See also `javax.xml.registry.infomodel.LocalizedString`

getLocalizedStrings

`Collection` getLocalizedStrings() throws `JAXRException`

Returns the LocalizedStrings associated with this object.

**Capability Level: 0**

**Returns:**
Collection of LocalizedString instances. The Collection may be empty but not null.

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error.

**See Also:**
`LocalizedString`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.internet Class InternetAddress

java.lang.Object  
  javax.mail.Address  
  javax.mail.internet.InternetAddress

All Implemented Interfaces: 
Serializable, Cloneable

public class InternetAddress
  extends Address
  implements Cloneable

Extends: Address 
Implements: Cloneable

RFC822 Internet "user@host.domain" "Personal Name"

This class represents an Internet email address using the syntax of RFC822. Typical address syntax is of the form "user@host.domain" or "Personal Name".

Author: 
  Bill Shannon, John Mani
See Also: 
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected String</th>
<th>address</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String</td>
<td>encodedPersonal</td>
</tr>
<tr>
<td></td>
<td>The RFC 2047 encoded version of the personal name.</td>
</tr>
</tbody>
</table>
protected String personal
    The personal name.

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternetAddress()</td>
<td>Default constructor.</td>
</tr>
<tr>
<td>InternetAddress(String address)</td>
<td>Constructor.</td>
</tr>
<tr>
<td>InternetAddress(String address, boolean strict)</td>
<td>Parse the given string and create an InternetAddress.</td>
</tr>
<tr>
<td>InternetAddress(String address, String personal)</td>
<td>Construct an InternetAddress given the address and personal name.</td>
</tr>
<tr>
<td>InternetAddress(String address, String personal, String charset)</td>
<td>Construct an InternetAddress given the address and personal name.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object clone()</td>
<td>Return a copy of this InternetAddress object.</td>
</tr>
<tr>
<td>boolean equals(Object a)</td>
<td>The equality operator.</td>
</tr>
<tr>
<td>String getAddress()</td>
<td>Get the email address.</td>
</tr>
<tr>
<td>InternetAddress[] getGroup(boolean strict)</td>
<td>Return the members of a group address.</td>
</tr>
<tr>
<td>static InternetAddress getLocalAddress(Session session)</td>
<td>Return an InternetAddress object representing the current user.</td>
</tr>
<tr>
<td>String getPersonal()</td>
<td>Get the personal name.</td>
</tr>
<tr>
<td>String getType()</td>
<td>Return the type of this address.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td>Compute a hash code for the address.</td>
</tr>
<tr>
<td><code>isGroup()</code></td>
<td>Indicates whether this address is an RFC 822 group address.</td>
</tr>
<tr>
<td><code>parse(String addresslist)</code></td>
<td>Parse the given comma separated sequence of addresses into InternetAddress objects.</td>
</tr>
<tr>
<td><code>parse(String addresslist, boolean strict)</code></td>
<td>Parse the given sequence of addresses into InternetAddress objects.</td>
</tr>
<tr>
<td><code>parseHeader(String addresslist, boolean strict)</code></td>
<td>Parse the given sequence of addresses into InternetAddress objects.</td>
</tr>
<tr>
<td><code>setAddress(String address)</code></td>
<td>Set the email address.</td>
</tr>
<tr>
<td><code>setPersonal(String name)</code></td>
<td>Set the personal name.</td>
</tr>
<tr>
<td><code>setPersonal(String name, String charset)</code></td>
<td>Set the personal name.</td>
</tr>
<tr>
<td><code>toString()</code></td>
<td>Convert this address into a RFC 822 / RFC 2047 encoded address.</td>
</tr>
<tr>
<td><code>toString(Address[] addresses)</code></td>
<td>Convert the given array of InternetAddress objects into a comma separated sequence of address strings.</td>
</tr>
<tr>
<td><code>toString(Address[] addresses, int used)</code></td>
<td>Convert the given array of InternetAddress objects into a comma separated sequence of address strings.</td>
</tr>
<tr>
<td><code>toUnicodeString()</code></td>
<td>Returns a properly formatted address (RFC 822 syntax) of Unicode characters.</td>
</tr>
</tbody>
</table>
void validate()
    Validate that this address conforms to the syntax rules of RFC 822.

Methods inherited from class java.lang.Object
final, getClass, notify, notifyAll, wait, wait, wait

Field Detail

address
protected String address

personal
protected String personal
    The personal name.

encodedPersonal
protected String encodedPersonal
    The RFC 2047 encoded version of the personal name.
    This field and the personal field track each other, so if a subclass sets one of these fields directly, it should set the other to null, so that it is suitably recomputed.
public InternetAddress()

Default constructor.

public InternetAddress(String address) throws AddressException

Constructor.

Parse the given string and create an InternetAddress. See the parse method for details of the parsing. The address is parsed using "strict" parsing. This constructor does not perform the additional syntax checks that the InternetAddress(String address, boolean strict) constructor does when strict is true. This constructor is equivalent
public InternetAddress(String address, boolean strict)
throws AddressException

Parameters:
address - the address in RFC822 format
strict - enforce RFC822 syntax

Throws:
AddressException - if the parse failed

Since:
JavaMail 1.3

Parse the given string and create an InternetAddress. If strict is false, the detailed syntax of the address isn't checked.

public InternetAddress(String address, String personal)
throws java.io.UnsupportedEncodingException

Parameters:
address - the address in RFC822 format
personal - the personal part of the address

Since:
JavaMail 1.3
InternetAddress

public InternetAddress(String address, String personal)
throws UnsupportedEncodingException

Construct an InternetAddress given the address and personal name. The address is assumed to be a syntactically valid RFC822 address.

Parameters:
address - the address in RFC822 format
personal - the personal name

Throws:
UnsupportedEncodingException

public InternetAddress(String address, String personal, String charset)
throws java.io.UnsupportedEncodingException

Construct an InternetAddress given the address and personal name. The address is assumed to be a syntactically valid RFC822 address.
Parameters:
  address - the address in RFC822 format
  personal - the personal name
  charset - the MIME charset for the name

Throws:
  UnsupportedEncodingException

Method Detail

public Object clone()

InternetAddress

since JavaMail 1.2

clone

public Object clone()

Return a copy of this InternetAddress object.

Overrides:
  clone in class Object

Since:
  JavaMail 1.2

public String getType()

InternetAddress "rfc822"

getype

public String getType()

Return the type of this address. The type of an InternetAddress is "rfc822".
**setAddress**

```java
public void setAddress(String address)
```

*address*

**setAddress**

```java
public void setAddress(String address)
```

Set the email address.

**Parameters:**

- `address` - email address

**setPersonal**

```java
public void setPersonal(String name, String charset)
```

**Throws**


**See also**

- setPersonal(String)

**setPersonal**

```java
public void setPersonal(String name, String charset)
```
Set the personal name. If the name contains non US-ASCII characters, then the name will be encoded using the specified charset as per RFC 2047. If the name contains only US-ASCII characters, no encoding is done and the name is used as is.

Parameters:
- name - personal name
- charset - MIME charset to be used to encode the name as per RFC 2047

Throws:
- UnsupportedEncodingException - if the charset encoding fails.

See Also:
- setPersonal(String)

---

```java
public void setPersonal(String name) throws java.io.UnsupportedEncodingException
```

name

Throws java.io.UnsupportedEncodingException:

See also setPersonal(String name, String charset)

---

```java
public void setPersonal(String name)
```

Set the personal name. If the name contains non US-ASCII characters, then the name will be encoded using the platform's default charset. If the name contains only US-ASCII characters, no encoding is done and the name is used as is.

Parameters:
- name - personal name
Throws:
UnsupportedEncodingException - if the charset encoding fails.

See Also:
setPersonal(String name, String charset)

public String getAddress()

    return

getAddress

public String getAddress()

    Get the email address.

    Returns:
    email address

public String getPersonal()

RFC 2047  Unicode

    return

getPersonal

public String getPersonal()

    Get the personal name. If the name is encoded as per RFC 2047, it is decoded and converted into Unicode. If the decoding or conversion fails, the raw data is returned as is.

    Returns:
    personal name
public String toString()
RFC 822 / RFC 2047  US-ASCII

    return

**toString**

```java
public String toString()
```

Convert this address into a RFC 822 / RFC 2047 encoded address. The resulting string contains only US-ASCII characters, and hence is mail-safe.

**Specified by:**
```
toString in class Address
```

**Returns:**
```
possibly encoded address string
```

---

public String toUnicodeString()
Unicode RFC 822

    return
    since     Unicode
              JavaMail 1.2

**toUnicodeString**

```java
public String toUnicodeString()
```

Returns a properly formatted address (RFC 822 syntax) of Unicode characters.

**Returns:**
```
Unicode address string
```

**Since:**
```
JavaMail 1.2
```
public boolean equals(Object a)

equals

public boolean equals(Object a)

The equality operator.

Specified by:

equals in class Address

Parameters:

a - Address object

public int hashCode()

hashCode

public int hashCode()

Compute a hash code for the address.

Overrides:

hashCode in class Object

public static String toString(Address[] addresses)

InternetAddress US-ASCII

addresses InternetAddress

Throws

ClassCastException,; InternetAddress

return

RuntimeException
public static String toString(Address[] addresses)

Convert the given array of InternetAddress objects into a comma separated sequence of address strings. The resulting string contains only US-ASCII characters, and hence is mail-safe.

Parameters:
addresses - array of InternetAddress objects

Returns:
comma separated string of addresses

Throws:
ClassCastException, - if any address object in the given array is not an InternetAddress object. Note that this is a RuntimeException.

public static String toString(Address[] addresses, int used)

Convert the given array of InternetAddress objects into a comma separated sequence of address strings. The resulting string contains only US-ASCII characters, and hence is mail-safe.

Parameters:
addresses - array of InternetAddress objects
used - index of the last used address

Returns:
comma separated string of addresses

Throws:
ClassCastException, - if any address object in the given array is not an InternetAddress object. Note that this is a RuntimeException.
separated sequence of address strings. The resulting string contains only US-ASCII characters, and hence is mail-safe.

The 'used' parameter specifies the number of character positions already taken up in the field into which the resulting address sequence string is to be inserted. It is used to determine the line-break positions in the resulting address sequence string.

**Parameters:**
- addresses - array of InternetAddress objects
- used - number of character positions already used, in the field into which the address string is to be inserted.

**Returns:**
- comma separated string of addresses

**Throws:**
- ClassCastException, - if any address object in the given array is not an InternetAddress object. Note that this is a RuntimeException.

```java
public static InternetAddress getLocalAddress(Session session)

InternetAddress "mail.from" "mail.user"
"mail.host" "user.name"
InetAddress.getLocalHost null
    session               Session
    return

getLocalAddress

public static InternetAddress getLocalAddress(Session session)

Return an InternetAddress object representing the current user. The entire email address may be specified in the "mail.from" property. If not set, the "mail.user" and "mail.host" properties are tried. If those are not set, the "user.name" property and InetAddress.getLocalHost method are tried. Security exceptions that may occur while
accessing this information are ignored. If it is not possible to determine an email address, null is returned.

**Parameters:**
- session - Session object used for property lookup

**Returns:**
- current user’s email address

---

```java
public static InternetAddress[] parse(String addresslist)
throws AddressException
```

**InternetAddress RFC822**

**addresslist**

`return` - return InternetAddress

**Throws:**
- `AddressException`:

`parse`

---

```java
public static InternetAddress[] parse(String addresslist)
throws AddressException
```

**Parse** the given comma separated sequence of addresses into InternetAddress objects. Addresses must follow RFC822 syntax.

**Parameters:**
- addresslist - comma separated address strings

**Returns:**
- array of InternetAddress objects

**Throws:**
- `AddressException` - if the parse failed

---

```java
public static InternetAddress[] parse(String addresslist, boolean strict)
throws AddressException
```

**InternetAddress**

`strict` - false

`trueRFC822` - stri
Parse the given sequence of addresses into InternetAddress objects. If strict is false, simple email addresses separated by spaces are also allowed. If strict is true, many (but not all) of the RFC822 syntax rules are enforced. In particular, even if strict is true, addresses composed of simple names (with no "@domain" part) are allowed. Such "illegal" addresses are not uncommon in real messages.

Non-strict parsing is typically used when parsing a list of mail addresses entered by a human. Strict parsing is typically used when parsing address headers in mail messages.

**Parameters:**
- addresslist - comma separated address strings
- strict - enforce RFC822 syntax

**Returns:**
- array of InternetAddress objects

**Throws:**
- AddressException - if the parse failed
public static InternetAddress[] parseHeader(String addresslist, boolean strict) throws AddressException

Parse the given sequence of addresses into InternetAddress objects. If strict is false, the full syntax rules for individual addresses are not enforced. If strict is true, many (but not all) of the RFC822 syntax rules are enforced.

To better support the range of "invalid" addresses seen in real messages, this method enforces fewer syntax rules than the parse method when the strict flag is false and enforces more rules when the strict flag is true. If the strict flag is false and the parse is successful in separating out an email address or addresses, the syntax of the addresses themselves is not checked.

Parameters:
  addresslist - comma separated address strings
  strict - enforce RFC822 syntax

Returns:
  array of InternetAddress objects

Throws:
public void validate() throws AddressException
RFC 822

Throws
AddressException:
since
JavaMail 1.3

validate

public void validate()
throws AddressException

Validate that this address conforms to the syntax rules of RFC 822. The current implementation checks many, but not all, syntax rules. Note that even though the syntax of the address may be correct, there's no guarantee that a mailbox of that name exists.

Throws:
AddressException - if the address isn't valid.
Since:
JavaMail 1.3

public boolean isGroup()
RFC 822  RFC 822

return true
since
JavaMail 1.3

isGroup
Indicates whether this address is an RFC 822 group address. Note that a group address is different than the mailing list addresses supported by most mail servers. Group addresses are rarely used; see RFC 822 for details.

**Returns:**
true if this address represents a group

**Since:**
JavaMail 1.3

```java
public InternetAddress[] getGroup(boolean strict) throws AddressException
0 1 null
parseHeader
    return
    internetAddress null
    throws AddressException:
    since JavaMail 1.3
```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
InternetHeaders is a utility class that manages RFC822 style headers. Given an RFC822 format message stream, it reads lines until the blank line that indicates end of header. The input stream is positioned at the start of the body. The lines are stored within the object and can be extracted as either Strings or Header objects.

This class is mostly intended for service providers. MimeMessage and MimeBody use this class for holding their headers.

A note on RFC822 and MIME headers
RFC822 and MIME header fields must contain only US-ASCII characters. If a header contains non US-ASCII characters, it must be encoded as per the rules in RFC 2047. The MimeUtility class provided in this package can be used to to achieve this. Callers of the setHeader, addHeader, and addHeaderLine methods are responsible for enforcing the MIME requirements for the specified headers. In addition, these header fields must be folded (wrapped) before being sent if they exceed the line length limitation for the transport (1000 bytes for SMTP). Received headers may have been folded. The application is responsible for folding and unfolding headers as appropriate.

Author:  
John Mani, Bill Shannon

See Also:  
MimeUtility

### Nested Class Summary

<table>
<thead>
<tr>
<th>protected static class</th>
<th>InternetHeaders.InternetHeader</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An individual internet header.</td>
</tr>
</tbody>
</table>

### Field Summary

<table>
<thead>
<tr>
<th>protected list</th>
<th>headers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The actual list of Headers, including placeholder entries.</td>
</tr>
</tbody>
</table>

### Constructor Summary

<table>
<thead>
<tr>
<th>InternetHeaders()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create an empty InternetHeaders object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>InternetHeaders(InputStream is)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read and parse the given RFC822 message stream till the blank line separating the header from the body.</td>
</tr>
</tbody>
</table>

### Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void addHeader(String name, String value)</strong></td>
<td>Add a header with the specified name and value to the header list.</td>
</tr>
<tr>
<td><strong>void addHeaderLine(String line)</strong></td>
<td>Add an RFC822 header line to the header store.</td>
</tr>
<tr>
<td><strong>Enumeration getAllHeaderLines()</strong></td>
<td>Return all the header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><strong>Enumeration getAllHeaders()</strong></td>
<td>Return all the headers as an Enumeration of Header objects.</td>
</tr>
<tr>
<td><strong>String[] getHeader(String name)</strong></td>
<td>Return all the values for the specified header.</td>
</tr>
<tr>
<td><strong>String getHeader(String name, String delimiter)</strong></td>
<td>Get all the headers for this header name, returned as a single String, with headers separated by the delimiter.</td>
</tr>
<tr>
<td><strong>Enumeration getMatchingHeaderLines(String[] names)</strong></td>
<td>Return all matching header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><strong>Enumeration getMatchingHeaders(String[] names)</strong></td>
<td>Return all matching Header objects.</td>
</tr>
<tr>
<td><strong>Enumeration getNonMatchingHeaderLines(String[] names)</strong></td>
<td>Return all non-matching header lines.</td>
</tr>
<tr>
<td><strong>Enumeration getNonMatchingHeaders(String[] names)</strong></td>
<td>Return all non-matching Header objects.</td>
</tr>
<tr>
<td><strong>void load(InputStream is)</strong></td>
<td>Read and parse the given RFC822 message stream till the blank line separating the header from the body.</td>
</tr>
<tr>
<td><strong>void removeHeader(String name)</strong></td>
<td>Remove all header entries that match the given name.</td>
</tr>
<tr>
<td><strong>void setHeader(String name, String value)</strong></td>
<td>Change the first header line that matches name to have value, adding a new header if no existing header matches.</td>
</tr>
</tbody>
</table>
Field Detail

headers

protected `List` `headers`

The actual list of Headers, including placeholder entries. Placeholder entries are Headers with a null value and are never seen by clients of the InternetHeaders class. Placeholder entries are used to keep track of the preferred order of headers. Headers are never actually removed from the list, they're converted into placeholder entries. New headers are added after existing headers of the same name (or before in the case of `Received` and `Return-Path` headers). If no existing header or placeholder for the header is found, new headers are added after the special placeholder with the name `"."`.

Since:
JavaMail 1.4

Constructor Detail

`public InternetHeaders()`

`InternetHeaders`

`public InternetHeaders()`

Create an empty `InternetHeaders` object. Placeholder entries are
public InternetHeaders(java.io.InputStream is) throws MessagingException

RFC822

BufferedInputStream

\[is\] RFC822

InternetHeaders

public InternetHeaders(InputStream is) throws MessagingException

Read and parse the given RFC822 message stream till the blank line separating the header from the body. The input stream is left positioned at the start of the body. The header lines are stored internally.

For efficiency, wrap a BufferedInputStream around the actual input stream and pass it as the parameter.

No placeholder entries are inserted; the original order of the headers is preserved.

Parameters:
  \textit{is} - RFC822 input stream

Throws:
  MessagingException

Method Detail
public void load(InputStream is) throws MessagingException
RFC822 InternetHeaders

InternetHeaders

load

public void load(InputStream is)
throws MessagingException

Read and parse the given RFC822 message stream till the blank line separating the header from the body. Store the header lines inside this InternetHeaders object. The order of header lines is preserved.

Note that the header lines are added into this InternetHeaders object, so any existing headers in this object will not be affected. Headers are added to the end of the existing list of headers, in order.

Parameters:
    is - RFC822 input stream

Throws:
    MessagingException

public String[] getHeader(String name)
String

getHeader

public String[] getHeader(String name)
Return all the values for the specified header. The values are String objects. Returns null if no headers with the specified name exist.

**Parameters:**
- `name` - header name

**Returns:**
- array of header values, or null if none

```
public String getHeader(String name, String delimiter)
```

Get all the headers for this header name, returned as a single String, with headers separated by the delimiter. If the delimiter is null, only the first header is returned. Returns null if no headers with the specified name exist.

**Parameters:**
- `name` - header name
- `delimiter` - delimiter

**Returns:**
- the value fields for all headers with this name, or null if none

```
public void setHeader(String name, String value)
```

RFC822 US-ASCII
setHeader

public void setHeader(String name, String value)

Change the first header line that matches name to have value, adding a new header if no existing header matches. Remove all matching headers but the first.

Note that RFC822 headers can only contain US-ASCII characters

Parameters:
  name - header name
  value - header value

addHeader

public void addHeader(String name, String value)

Add a header with the specified name and value to the header list.
The current implementation knows about the preferred order of most well-known headers and will insert headers in that order. In addition, it knows that Received headers should be inserted in reverse order (newest before oldest), and that they should appear at the beginning of the headers, preceded only by a possible Return-Path header.

Note that RFC822 headers can only contain US-ASCII characters.

**Parameters:**
- name - header name
- value - header value

```java
public void removeHeader(String name)
```

**removeHeader**

```java
public void removeHeader(String name)
```

Remove all header entries that match the given name

**Parameters:**
- name - header name

```java
public java.util.Enumeration<E> getAllHeaders()
```

**getAllHeaders**

```java
public Enumeration getAllHeaders()
```

Return all the headers as an Enumeration of `Header` objects.
Returns:
Header objects

public java.util.Enumeration<E>
getMatchingHeaders(String[] names)

javax.mail.Header
return Header

getMatchingHeaders

public Enumeration getMatchingHeaders(String[] names)

Return all matching Header objects.

Returns:
matching Header objects

public java.util.Enumeration<E>
getNonMatchingHeaders(String[] names)

javax.mail.Header
return Header

getNonMatchingHeaders

public Enumeration getNonMatchingHeaders(String[] names)

Return all non-matching Header objects.

Returns:
non-matching Header objects

public void addHeaderLine(String line)
addHeaderLine

public void addHeaderLine(String line)

Add an RFC822 header line to the header store. If the line starts with a space or tab (a continuation line), add it to the last header line in the list. Otherwise, append the new header line to the list.

Note that RFC822 headers can only contain US-ASCII characters

Parameters:
    line - raw RFC822 header line

public java.utilEnumeration<String> getAllHeaderLines()

Return all the header lines as an Enumeration of Strings.

getAllHeaderLines

public Enumeration getAllHeaderLines()

getMatchingHeaderLines(String[] names)

String
getMatchingHeaderLines

public Enumeration getMatchingHeaderLines(String[] names)

    Return all matching header lines as an Enumeration of Strings.

getNonMatchingHeaderLines

public java.util Enumeration<E>
getNonMatchingHeaderLines(String[] names)

getNonMatchingHeaderLines

public Enumeration getNonMatchingHeaderLines(String[] names)

    Return all non-matching header lines

Overview  Package  Tree  Deprecated  Index  Help

PREV CLASS  NEXT CLASS
SUMMARY: NESTED  FIELD  CONSTR  METHOD
FRAMES  NO FRAMES
DETAIL: FIELD  CONSTR  METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
An individual internet header. This class is only used by subclasses of InternetHeaders.

An InternetHeader object with a null value is used as a placeholder for headers of that name, to preserve the order of headers. A placeholder InternetHeader object with a name of "::" marks the location in the list of headers where new headers are added by default.

Since:
JavaMail 1.4
Fields inherited from class javax.mail.Header
name, value

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InternetHeaders.InternetHeader(String l)</td>
<td>Constructor that takes a line and splits out the header name.</td>
</tr>
<tr>
<td>InternetHeaders.InternetHeader(String n, String v)</td>
<td>Constructor that takes a header name and value.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getValue()</td>
<td>Return the &quot;value&quot; part of the header line.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.Header
getName

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

class InternetHeaders.InternetHeader

Public Class InternetHeaders.InternetHeader

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>public InternetHeaders.InternetHeader(String l)</td>
<td>Constructor that takes a line and splits out the header name.</td>
</tr>
</tbody>
</table>
public InternetHeaders.InternetHeader(String n, String v)

InternetHeaders.InternetHeader

public InternetHeaders.InternetHeader(String n, String v)

Constructor that takes a header name and value.

Method Detail

public String getValue()
"value"

getValue

public String getValue()

Return the "value" part of the header line.

Overrides:
getValue in class Header

Returns:
value of the header
PS:
javax.xml.rpc.holders Class IntHolder

java.lang.Object
   javax.xml.rpc.holders.IntHolder

All Implemented Interfaces:
   Holder

public final class IntHolder

extends Object
implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>int</th>
<th>value</th>
</tr>
</thead>
</table>

Constructor Summary

IntHolder()

IntHolder(int myint)

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
value

public int value

Constructor Detail

public IntHolder()

IntHolder

public IntHolder()

public IntHolder(int myint)

IntHolder

public IntHolder(int myint)
PS:
javax.jms Class InvalidClientIDException

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
     └ javax.jms.JMSException
      └ javax.jms.InvalidClientIDException

All Implemented Interfaces:
  Serializable

public class InvalidClientIDException

extends JMSException

Extends: Throwable > Exception > JMSException

ID

version 26 August 1998

This exception must be thrown when a client attempts to set a connection's client ID to a value that is rejected by a provider.

Version:
  26 August 1998

Author:
  Rahul Sharma

See Also:
  Serialized Form

Constructor Summary

InvalidClientIDException(String reason)
   Constructs an InvalidClientIDException with the specified
InvalidClientIDException(String reason, String errorCode)

Constructs an InvalidClientIDException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSEException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public InvalidClientIDException(String reason, String errorCode)

InvalidClientIDException
reason
errorCode

InvalidClientIDException

public InvalidClientIDException(String reason, String errorCode)

Constructs an InvalidClientIDException with the specified reason
and error code.

**Parameters:**
- reason - a description of the exception
- errorCode - a string specifying the vendor-specific error code

```java
public InvalidClientIDException(String reason)
```

InvalidClientIDException

```java
public InvalidClientIDException(String reason)
```

Constructs an InvalidClientIDException with the specified reason. The error code defaults to null.

**Parameters:**
- reason - a description of the exception

---

**Overview**

Package

Class

Tree

Deprecated

Index

Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Class InvalidDestinationException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.jms.JMSException
            ↓ javax.jms.InvalidDestinationException

All Implemented Interfaces:
   Serializable

public class InvalidDestinationException
    extends JMSException

Extends: Throwable > Exception > JMSException

version 26 August 1998

This exception must be thrown when a destination either is not understood by a provider or is no longer valid.

Version:
   26 August 1998

Author:
   Rahul Sharma

See Also:
   Serialized Form

Constructor Summary

InvalidDestinationException(String reason)
   Constructs an InvalidDestinationException with the specified
InvalidDestinationException(String reason, String errorCode)

Constructs an InvalidDestinationException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public InvalidDestinationException(String reason, String errorCode)

InvalidDestinationException
reason
errorCode

InvalidDestinationException

public InvalidDestinationException(String reason, String errorCode)

Constructs an InvalidDestinationException with the specified
reason and error code.

**Parameters:**
- `reason` - a description of the exception
- `errorCode` - a string specifying the vendor-specific error code

```java
public InvalidDestinationException(String reason)
InvalidDestinationException null
  reason
```

InvalidDestinationException

public InvalidDestinationException(String reason)

Constructs an InvalidDestinationException with the specified reason. The error code defaults to null.

**Parameters:**
- `reason` - a description of the exception
javax.enterprise.deploy.spi.exceptions Class
InvalidModuleException

java.lang.Object
    └ java.lang.Throwable
        └ java.lang.Exception
            └ javax.enterprise.deploy.spi.exceptions.InvalidModuleException

All Implemented Interfaces:
    Serializable

public class InvalidModuleException
    extends Exception

Extends: Throwable > Exception

J2EE

This exception is to report an invalid J2EE deployment module type.

See Also:
    Serialized Form

Constructor Summary

InvalidModuleException(String s)
Creates an new InvalidModuleException object.

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace,
printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public InvalidModuleException(String s)
InvalidModuleException

s

InvalidModuleException

public InvalidModuleException(String s)

Creates a new InvalidModuleException object.

Parameters:
s - a string indicating what was wrong with the module type.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.resource.spi  Class InvalidPropertyException

java.lang.Object  
  ↓ java.lang.Throwable  
    ↓ java.lang.Exception  
      ↓ javax.resource.ResourceException  
        ↓ javax.resource.spi.InvalidPropertyException

All Implemented Interfaces:  
  Serializable

```
javax.resource.spi.InvalidPropertyException
```

public class InvalidPropertyException

extends RuntimeException

Extends: Throwable > Exception > RuntimeException

version 0.2

This exception is thrown to indicate invalid configuration property settings.

Version: 0.2
Author: Ram Jeyaraman
See Also: Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>InvalidPropertyException()</td>
<td>Create a InvalidPropertyException.</td>
</tr>
<tr>
<td>InvalidPropertyException(String message)</td>
<td>Create a InvalidPropertyException.</td>
</tr>
</tbody>
</table>
**InvalidPropertyException** *(String message, String errorCode)*
Constructs a new throwable with the specified detail message and an error code.

**InvalidPropertyException** *(String message, Throwable cause)*
Constructs a new throwable with the specified detail message and cause.

**InvalidPropertyException** *(Throwable cause)*
Constructs a new throwable with the specified cause.

---

**Method Summary**

<table>
<thead>
<tr>
<th>PropertyDescriptor[]</th>
<th>getInvalidPropertyDescriptors()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the list of invalid properties.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void</th>
<th>setInvalidPropertyDescriptors(PropertyDescriptor[])</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Set a list of invalid properties.</td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.resource.ResourceException**

getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException

**Methods inherited from class java.lang.Throwable**

fillInStackTrace, get Cause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

**Constructor Detail**

```java
public InvalidPropertyException()
InvalidPropertyException```

InvalidPropertyException

public InvalidPropertyException()

Create an InvalidPropertyException.

public InvalidPropertyException(String message)
InvalidPropertyException

message

InvalidPropertyException

public InvalidPropertyException(String message)

Create an InvalidPropertyException.

Parameters:
message - a description of the exception

InvalidPropertyException

public InvalidPropertyException(Throwable cause)
cause throwable

cause

InvalidPropertyException

public InvalidPropertyException(Throwable cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

InvalidPropertyException

public InvalidPropertyException(String message,
InvalidPropertyException

public InvalidPropertyException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public InvalidPropertyException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
public void setInvalidPropertyDescriptors(PropertyDescriptor[] invalidProperties)

Set a list of invalid properties.

public java.beans.PropertyDescriptor[] getInvalidPropertyDescriptors()

Get the list of invalid properties.
PS:

javax.xml.registry Class InvalidRequestException

java.lang.Object
   ↓ java.lang.Throwable
       ↓ java.lang.Exception
           ↓ javax.xml.registry.JAXRException
               ↓ javax.xml.registry.InvalidRequestException

All Implemented Interfaces:
   Serializable, JAXRResponse

public class InvalidRequestException
   extends JAXRException

Extends: Throwable > Exception > JAXRException

JAXR API

This exception is thrown when a JAXR client attempts to invoke a method that is not valid for some reason.

Author:
   Farrukh S. Najmi

See Also:
  Serialized Form

Field Summary

| Fields inherited from class javax.xml.registry.JAXRException |
| cause |

| Fields inherited from interface javax.xml.registry.JAXRResponse |
| STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING |
**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InvalidRequestException()</code></td>
<td>Constructs a JAXRException object with no reason or embedded Throwable.</td>
</tr>
<tr>
<td><code>InvalidRequestException(String reason)</code></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><code>InvalidRequestException(String reason, Throwable cause)</code></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.</td>
</tr>
<tr>
<td><code>InvalidRequestException(Throwable cause)</code></td>
<td>Constructs a JAXRException object initialized with the given Throwable object.</td>
</tr>
</tbody>
</table>

**Method Summary**

Methods inherited from class `javax.xml.registry.JAXRException`:
- `getCause`, `getMessage`, `getRequestId`, `getStatus`, `initCause`, `isAvailable`

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public InvalidRequestException()
        Throwable JAXRException

InvalidRequestException

public InvalidRequestException()

    Constructs a JAXRException object with no reason or embedded Throwable.

public InvalidRequestException(String reason)

    JAXRException String reason

InvalidRequestException

public InvalidRequestException(String reason)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

    Parameters:
        reason - a description of what caused the exception

public InvalidRequestException(String reason, Throwable cause)

    JAXRException String Throwable
    Throwable reason
    cause JAXRException Throwable
public InvalidRequestException(String reason, Throwable cause)

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
reason - a description of what caused the exception
cause - a Throwable object that is to be embedded in this JAXRException object

public InvalidRequestException(Throwable cause)

Throwable JAXRException
cause Exception Throwable

InvalidRequestException

public InvalidRequestException(Throwable cause)

Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
cause - the Throwable that caused this Exception

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.jms Class InvalidSelectorException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.jms.JMSException
            ↓ javax.jms.InvalidSelectorException

All Implemented Interfaces:
   Serializable

public class InvalidSelectorException
extends JMSException

Extends: Throwable > Exception > JMSException

JMS

version 26 August 1998

This exception must be thrown when a JMS client attempts to give a provider a message selector with invalid syntax.

Version:
   26 August 1998

Author:
   Rahul Sharma

See Also:
   Serialized Form

Constructor Summary

InvalidSelectorException(String reason)
   Constructs an InvalidSelectorException with the specified
InvalidSelectorException(String reason, String errorCode)

Constructs an InvalidSelectorException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public InvalidSelectorException(String reason, String errorCode)

InvalidSelectorException

    reason

    errorCode

InvalidSelectorException

public InvalidSelectorException(String reason, String errorCode)

    Constructs an InvalidSelectorException with the specified reason
Parameters:
    reason - a description of the exception
errorCode - a string specifying the vendor-specific error code

public InvalidSelectorException(String reason)

InvalidSelectorException

public InvalidSelectorException(String reason)

Constructs an InvalidSelectorException with the specified reason. The error code defaults to null.

Parameters:
    reason - a description of the exception
javax.transaction  Class InvalidTransactionException

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
      └ java.io.IOException
        └ java.rmi.RemoteException
          └ javax.transaction.InvalidTransactionException

All Implemented Interfaces:
  Serializable

public class InvalidTransactionException
  extends RemoteException

  Extends: Throwable > Exception > java.io.IOException >
  java.rmi.RemoteException

This exception indicates that the request carried an invalid transaction context. For example, this exception could be raised if an error occurred when trying to register a resource.

See Also:
  Serialized Form

Field Summary

- Fields inherited from class java.rmi.RemoteException
detail

Constructor Summary
InvalidTransactionException

<table>
<thead>
<tr>
<th>Constructor Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public InvalidTransactionException()</td>
</tr>
</tbody>
</table>

InvalidTransactionException

Methods inherited from class java.rmi.RemoteException
getCause, getMessage

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
public InvalidTransactionException(String msg)
javax.interceptor  Interface InvocationContext

public interface InvocationContext

AroundInvoke  Interceptor

Context information passed to AroundInvoke and Interceptor-class lifecycle callback methods.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| Map<String, Object>       | **getContextData()**  
|                           | Returns the context data associated with this invocation or lifecycle callback. |
| Method                    | **getMethod()**  
|                           | Returns the method of the bean class for which the interceptor was invoked. |
| Object[]                  | **getParameters()**  
|                           | Returns the parameters that will be used to invoke the business method. |
| Object                    | **getTarget()**  
|                           | Returns the target instance. |
| Object                    | **proceed()**  
|                           | Proceed to the next entry in the interceptor chain. |
| void                      | **setParameters(Object[] params)**  
|                           | Sets the parameters that will be used to invoke the business method. |

Method Detail
public Object getTarget()

getTarget

Object getTarget()

Returns the target instance.

public Method getMethod()
Interceptor Bean AroundInvoke Bean
null

getMethod

Method getMethod()

Returns the method of the bean class for which the interceptor was invoked. For AroundInvoke methods, this is the business method on the bean class. For lifecycle callback methods, returns null.

public Object[] getParameters()
setParameters getParameters()

Throws IllegalStateException:

getParameters

Object[] getParameters()

Returns the parameters that will be used to invoke the business method. If setParameters has been called, getParameters() returns the values to which the parameters have been set.
Throws:

- **IllegalStateException** - if invoked within a lifecycle callback method.

---

public void setParameters(Object[] params)

**Throws**

- IllegalStateException
- IllegalArgumentException

### setParameters

Sets the parameters that will be used to invoke the business method.

**Throws:**

- **IllegalStateException** - if invoked within a lifecycle callback method.
- **IllegalArgumentException** - if the parameter types do not match the types for the business method, or the number of parameters supplied does not equal the number of parameters for the business method.

---

public java.util.Map<K, V> getContextData()

**Map**

### getContextData

Returns the context data associated with this invocation or lifecycle callback. If there is no context data, an empty Map object will be returned.
public Object proceed() throws Exception
Interceptor proceed  void proceed
null

proceed

**Object**  **proceed()**

throws  **Exception**

Proceed to the next entry in the interceptor chain. The proceed method returns the result of the next method invoked. If the method returns void, proceed returns null.

**Throws:**  **Exception**
javax.servlet.jsp.tagext  Interface IterationTag

All Superinterfaces:
   JspTag, Tag

All Known Subinterfaces:
   BodyTag

All Known Implementing Classes:
   AttributeTag, BodyTagSupport, ConverterELTag, ConverterTag, FacetTag, TagSupport, UICOMPONENTBODYTag, UICOMPONENTCLASSTAGBase, UICOMPONENTELTag, UICOMPONENTTag, ValidatorELTag, ValidatorTag

public interface IterationTag
   extends Tag

Implements: Tag
Implemented by: BodyTag, TagSupport

IterationTag Tag
   doStartTag() doEndTag() Tag IterationTag
      doAfterBody()

   doAfterBody() doAfterBody()
   IterationTag.EVAL_BODY_AGAIN doAfterBody()
   Tag.SKIP_BODY doEndTag()

   Tag
   doAfterBody()

   doStartTag()BODY doAfterBody() TryCatchFinally
The `IterationTag` interface extends `Tag` by defining one additional method that controls the reevaluation of its body.

A tag handler that implements `IterationTag` is treated as one that implements `Tag` regarding the `doStartTag()` and `doEndTag()` methods. `IterationTag` provides a new method: `doAfterBody()`.

The `doAfterBody()` method is invoked after every body evaluation to control whether the body will be reevaluated or not. If `doAfterBody()` returns `IterationTag.EVAL_BODY_AGAIN`, then the body will be reevaluated. If `doAfterBody()` returns `Tag.SKIP_BODY`, then the body will be skipped and `doEndTag()` will be evaluated instead.

**Properties** There are no new properties in addition to those in `Tag`.

**Methods** There is one new methods: `doAfterBody()`.

**Lifecycle**

Lifecycle details are described by the transition diagram below. Exceptions that are thrown during the computation of `doStartTag()`, `BODY` and `doAfterBody()` interrupt the execution sequence and are propagated up the stack, unless the tag handler implements the `TryCatchFinally` interface; see that interface for details.
Empty and Non-Empty Action

If the TagLibraryDescriptor file indicates that the action must always have an empty element body, by a `<body-content>` entry of "empty", then the `doStartTag()` method must return `SKIP_BODY`.

Note that which methods are invoked after the `doStartTag()` depends on both the return value and on if the custom action element is empty or not in the JSP page, not on how it's declared in the TLD.

If `SKIP_BODY` is returned the body is not evaluated, and then `doEndTag()` is invoked.

If `EVAL_BODY_INCLUDE` is returned, and the custom action element is not empty, the body is evaluated and "passed through" to the current out, then `doAfterBody()` is invoked and, after zero or more iterations, `doEndTag()` is invoked.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static int</th>
<th>EVAL_BODY_AGAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Request the reevaluation of some body.</td>
</tr>
</tbody>
</table>

**Fields inherited from interface** `javax.servlet.jsp.tagext.Tag`

`EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

**Method Summary**

<table>
<thead>
<tr>
<th>int</th>
<th><code>doAfterBody()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Process body (re)evaluation.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface** `javax.servlet.jsp.tagext.Tag`

`doEndTag`, `doStartTag`, `getParent`, `release`, `setPageContext`, `setParent`
**Field Detail**

EVAL_BODY_AGAIN

static final int EVAL_BODY_AGAIN

Request the reevaluation of some body. Returned from doAfterBody. For compatibility with JSP 1.1, the value is carefully selected to be the same as the, now deprecated, BodyTag.EVAL_BODY_TAG,

See Also:
Constant Field Values

**Method Detail**

```java
public int doAfterBody() throws JspException
```

BodyEvaluation JSP

doAfterBody EVAL_BODY_AGAIN
doAfterBody doAfterBody SKIP_BODY
doEndTag

BodyTag doAfterBody SKIP_BODY doEndTag
pageContext popBody out

doAfterBody() JSP AT_BEGIN NESTED
TagExtraInfo TLD

return

Throws JspException:
doAfterBody

int doAfterBody() throws JspException

Process body (re)evaluation. This method is invoked by the JSP Page implementation object after every evaluation of the body into the BodyEvaluation object. The method is not invoked if there is no body evaluation.

If doAfterBody returns EVAL_BODY_AGAIN, a new evaluation of the body will happen (followed by another invocation of doAfterBody). If doAfterBody returns SKIP_BODY, no more body evaluations will occur, and the doEndTag method will be invoked.

If this tag handler implements BodyTag and doAfterBody returns SKIP_BODY, the value of out will be restored using the popBody method in pageContext prior to invoking doEndTag.

The method re-invocations may be lead to different actions because there might have been some changes to shared state, or because of external computation.

The JSP container will resynchronize the values of any AT_BEGIN and NESTED variables (defined by the associated TagExtraInfo or TLD) after the invocation of doAfterBody().

Returns:
whether additional evaluations of the body are desired

Throws:
JspException - if an error occurred while processing this tag
PS:
public interface **J2eeApplicationObject**

extends **DeployableObject**

**Implements:** **DeployableObject**

J2eeApplicationObject is an interface that represents a J2EE application (EAR); it maintains a DeployableObject for each module in the archive.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void addXPathListener(ModuleType type, String xpath, XpathListener xpl)</strong></td>
<td>Register a listener for changes in XPath that are related to this deployableObject.</td>
</tr>
<tr>
<td><strong>DDBean[] getChildBean(ModuleType type, String xpath)</strong></td>
<td>Return a list of DDBean based upon an XPath; all deployment descriptors of the specified type are searched.</td>
</tr>
<tr>
<td><strong>DeployableObject getDeployableObject(String uri)</strong></td>
<td>Return the DeployableObject of the specified URI designator.</td>
</tr>
<tr>
<td><strong>DeployableObject[] getDeployableObjects()</strong></td>
<td>Return the all DeployableObjects in this application.</td>
</tr>
<tr>
<td><strong>DeployableObject[] getDeployableObjects(ModuleType type)</strong></td>
<td>Return the all DeployableObject of the specified type.</td>
</tr>
</tbody>
</table>
getModuleUris

Return the list of URIs for all modules in the application.

getModuleUris(ModuleType type)

Return the list of URIs of the designated module type.

getText(ModuleType type, String xpath)

Return the text value from the XPath; search only the deployment descriptors of the specified type.

removeXPathListener(ModuleType type, String xpath, XpathListener xpl)

Unregister the listener for an XPath.

Methods inherited from interface javax.enterprise.deploy.model.DeployableObject
entries, getChildBean, getClassFromScope, getDDBeanRoot, getDDBeanRoot, getEntry, getModuleDTDVersion, getText, getType

Method Detail

public DeployableObject getDeployableObject(String uri)

URI DeployableObject

uri

return URI J2EE DeployableObject ‘null'

getDeployableObject

DeployableObject getDeployableObject(String uri)

Return the DeployableObject of the specified URI designator.

Parameters:

uri - Describes where to get the module from.
Returns:
the DeployableObject describing the j2ee module at this uri or 'null' if there is not match.

public DeployableObject[] getDeployableObjects(ModuleType type)
DeployableObject
    type
    return URI J2EE DeployableObject 'null'

defgetDeployableObjects
DeployableObject[] getDeployableObjects(ModuleType type)

    return URI J2EE DeployableObject 'null'

defgetDeployableObjects
    return URI J2EE DeployableObject 'null'

defgetDeployableObjects

    return URI J2EE DeployableObject 'null'
public String[] getModuleUris(ModuleType type)

URI

type
return

URI 'null'

getModuleUris

String[] getModuleUris(ModuleType type)

Return the list of URIs of the designated module type.

Parameters:
  type - The type of module to return.

Returns:
  the URIs of the contained modules or 'null' if there are no matches.

public String[] getModuleUris()

URI

return

URI 'null'

getModuleUris

String[] getModuleUris()

Return the list of URIs for all modules in the application.

Returns:
  the URIs of the contained modules or 'null' if there are no matches.
public **DDBean**[] getChildBean(**ModuleType** type, String xpath)

**XPath DDBean**

* type
* xpath  
* return DDBeans 'null'

**getChildBean**

**DDBean**[] **getChildBean**(**ModuleType** type, **String** xpath)

Return a list of DDBean based upon an XPath; all deployment descriptors of the specified type are searched.

**Parameters:**
* type - The type of deployment descriptor to query.
* xpath - An XPath string referring to a location in the deployment descriptor

**Returns:**
The list of DDBeans or 'null' of there are no matches.

---

public **String**[] getText(**ModuleType** type, String xpath)

**XPath**

* type
* xpath  
* return XPath 'null'

**getText**

**String**[] **getText**(**ModuleType** type, **String** xpath)

Return the text value from the XPath; search only the deployment descriptors of the specified type.
Parameters:
| type   | The type of deployment descriptor to query. |
| xpath  | An xpath string referring to a location in the deployment descriptor |

Returns:
The text values of this xpath or 'null' if there are no matches.

---

public void addXpathListener(ModuleType type, String xpath, XpathListener xpl)

addXpathListener

Register a listener for changes in XPath that are related to this deployableObject.

Parameters:
| type   | The type of deployment descriptor to query. |
| xpath  | The xpath to listen for. |
| xpl    | The listener. |

---

public void removeXpathListener(ModuleType type, String xpath, XpathListener xpl)

XPath

removeXpathListener

void removeXpathListener(ModuleType type, String xpath, XpathListener xpl)

Remove the listener for changes in XPath that are related to this deployableObject.
removeXpathListener

void removeXpathListener(ModuleType type, String xpath, XpathListener xpl)

Unregister the listener for an XPath.

Parameters:
- type - The type of deployment descriptor to query.
- xpath - The XPath to listen for
- xpl - The listener

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface JavaMailStats

JavaMail

Specifies the statistics provided by a JavaMail resource.

Method Summary

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getSentMailCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The number of mail messages sent.</td>
</tr>
</tbody>
</table>

Method Detail

public CountStatistic getSentMailCount()

gSentMailCount

CountStatistic getSentMailCount()

The number of mail messages sent.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind  **Class JAXBContext**

**java.lang.Object**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>javax.xml.bind.JAXBContext</td>
</tr>
</tbody>
</table>

```
public abstract class JAXBContext extends Object
```

**JAXBContext**  JAXB API  JAXB  XML/Java

```java
newInstance
```

- `JAXBContext.newInstance("com.acme.foo:com.acme.bar")`
  - JAXBContext  Java  java  JAXB/java
  - JAXB  JLS 3 7.4.1""
- `JAXBContext.newInstance(com.acme.foo.Foo.class)`
  - JAXBContext

```java
public static JAXBContext createContext(String contextPath, ClassLoader classLoader, Map properties)
```

**JAXB 1.0 java / JAXB JAXB**

```
jaxb.properties
javax.xml.bind.context.factory createContext
```

**marshal unmarshal**

```
DatatypeConverter.setDatatypeConverter api
```

**Unmarshaller**  XML  Java unmarshal  XML

unmarshal  xsi:type
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo:com.acme.bar" );
Unmarshaller u = jc.createUnmarshaller();
FooObject fooObj = (FooObject)u.unmarshal( new File( "foo.xml" ) );
BarObject barObj = (BarObject)u.unmarshal( new File( "bar.xml" ) );
BazObject bazObj = (BazObject)u.unmarshal( new File( "baz.xml" ) );

Java XML JAXB / JAXB

ObjectFactory contextPath java java
com.acme.foo PurchaseOrder

com.acme.foo.PurchaseOrder po =
com.acme.foo.ObjectFactory.createPurchaseOrder();

mutator

ObjectFactory 4.2" Java "

ObjectFactory newInstance( javaContentInterface )

Marshaller Java XML unmarshal java
java.io.OutputStream java.io.Writer XML
ContentHandler SAX2 DOM XML

XML

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );

// unmarshal from foo.xml
Unmarshaller u = jc.createUnmarshaller();
FooObject fooObj = (FooObject)u.unmarshal( new File( "foo.xml" ) );

// marshal to System.out
Marshaller m = jc.createMarshaller();
m.marshal( fooObj, System.out );
The JAXBContext class provides the client's entry point to the JAXB API. It provides an abstraction for managing the XML/Java binding information necessary to implement the JAXB binding framework operations: unmarshal, marshal and validate.

A client application normally obtains new instances of this class using one of these two styles for newInstance methods, although there are other specialized forms of the method available:

- **JAXBContext.newInstance("com.acme.foo:com.acme.bar")**
  The JAXBContext instance is initialized from a list of colon separated Java package names. Each java package contains JAXB mapped classes, schema-derived classes and/or user annotated classes. Additionally, the java package may contain JAXB package annotations that must be processed. (see JLS 3rd Edition, Section 7.4.1. Package Annotations).

- **JAXBContext.newInstance(com.acme.foo.Foo.class)**
  The JAXBContext instance is initialized with class(es) passed as parameter(s) and classes that are statically reachable from these class(es). See newInstance(Class...) for details.

**SPEC REQUIREMENT:** the provider must supply an implementation
class containing the following method signatures:

```java
public static JAXBContext createContext(String contextPath, ClassLoader classLoader, Map properties)
public static JAXBContext createContext(Class[] classes, Map properties) throws JAXBException
```

The following JAXB 1.0 requirement is only required for schema to java interface/implementation binding. It does not apply to JAXB annotated classes. JAXB Providers must generate a `jaxb.properties` file in each package containing schema derived classes. The property file must contain a property named `javax.xml.bind.context.factory` whose value is the name of the class that implements the `createContext` APIs.

The class supplied by the provider does not have to be assignable to `javax.xml.bind.JAXBContext`, it simply has to provide a class that implements the `createContext` APIs.

In addition, the provider must call the `DatatypeConverter.setDatatypeConverter` api prior to any client invocations of the marshal and unmarshal methods. This is necessary to configure the datatype converter that will be used during these operations.

**Unmarshalling**

The `Unmarshaller` class provides the client application the ability to convert XML data into a tree of Java content objects. The unmarshal method allows for any global XML element declared in the schema to be unmarshalled as the root of an instance document. Additionally, the unmarshal method allows for an unrecognized root element that has an xsi:type attribute's value that references a type definition declared in the schema to be unmarshalled as the root of an instance document. The `JAXBContext` object allows the merging of global elements and type definitions across a set of schemas (listed in the `contextPath`). Since each schema in the schema set can belong to distinct namespaces, the unification of schemas to an unmarshalling context should be namespace independent. This means that a client application is able to unmarshal XML documents that are instances of any of the schemas listed in the `contextPath`. 
For example:

```java
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo:com.acme.bar" );
Unmarshaller u = jc.createUnmarshaller();
FooObject fooObj = (FooObject)u.unmarshal( new File( "foo.xml" ) );
BarObject barObj = (BarObject)u.unmarshal( new File( "bar.xml" ) );
BazObject bazObj = (BazObject)u.unmarshal( new File( "baz.xml" ) );
```

The client application may also generate Java content trees explicitly rather than unmarshalling existing XML data. For all JAXB-annotated value classes, an application can create content using constructors. For schema-derived interface/implementation classes and for the creation of elements that are not bound to a JAXB-annotated class, an application needs to have access and knowledge about each of the schema derived ObjectFactory classes that exist in each of java packages contained in the contextPath. For each schema derived java class, there is a static factory method that produces objects of that type. For example, assume that after compiling a schema, you have a package `com.acme.foo` that contains a schema derived interface named `PurchaseOrder`. In order to create objects of that type, the client application would use the factory method like this:

```java
com.acme.foo.PurchaseOrder po =
    com.acme.foo.ObjectFactory.createPurchaseOrder();
```

Once the client application has an instance of the schema derived object, it can use the mutator methods to set content on it.

For more information on the generated ObjectFactory classes, see Section 4.2 Java Package of the specification.

**SPEC REQUIREMENT:** the provider must generate a class in each package that contains all of the necessary object factory methods for that package named ObjectFactory as well as the static newInstance( javaContentInterface ) method

**Marshalling**

The `Marshaller` class provides the client application the ability to convert a Java content tree back into XML data. There is no
difference between marshalling a content tree that is created manually using the factory methods and marshalling a content tree that is the result an unmarshal operation. Clients can marshal a java content tree back to XML data to a java.io.OutputStream or a java.io.Writer. The marshalling process can alternatively produce SAX2 event streams to a registered ContentHandler or produce a DOM Node object. Client applications have control over the output encoding as well as whether or not to marshal the XML data as a complete document or as a fragment.

Here is a simple example that unmarshals an XML document and then marshals it back out:

```java
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" )

// unmarshal from foo.xml
Unmarshaller u = jc.createUnmarshaller();
FooObject fooObj = (FooObject) u.unmarshal( new File( "foo.xml" ) )

// marshal to System.out
Marshaller m = jc.createMarshaller();
m.marshal( fooObj, System.out );
```

**Validation**

Validation has been changed significantly since JAXB 1.0. The `Validator` class has been deprecated and made optional. This means that you are advised not to use this class and, in fact, it may not even be available depending on your JAXB provider. JAXB 1.0 client applications that rely on `Validator` will still work properly when deployed with the JAXB 1.0 runtime system. In JAXB 2.0, the `Unmarshaller` has included convenince methods that expose the JAXP 1.3 `javax.xml.validation` framework. Please refer to the `Unmarshaller.setSchema(javax.xml.validation.Schema)` API for more information.

**JAXB Runtime Binding Framework Compatibility**

The following JAXB 1.0 restriction only applies to binding schema to interfaces/implementation classes. Since this binding does not require a common runtime system, a JAXB client application must
not attempt to mix runtime objects (JAXBContext, Marshaller, etc.) from different providers. This does not mean that the client application isn't portable, it simply means that a client has to use a runtime system provided by the same provider that was used to compile the schema.

Since:
   JAXB1.0
Version:
   $Revision: 1.24 $ $Date: 2006/03/08 17:05:01 $
Author:
   • Ryan Shoemaker, Sun Microsystems, Inc.
   • Kohsuke Kawaguchi, Sun Microsystems, Inc.
   • Joe Fialli, Sun Microsystems, Inc.
See Also:
   Marshaller, Unmarshaller, S 7.4.1.1 "Package Annotations" in Java Language Specification, 3rd Edition

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>JAXB_CONTEXT_FACTORY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name of the property that contains the name of the class capable of creating new JAXBContext objects.</td>
</tr>
</tbody>
</table>

Constructor Summary

| protected JAXBContext () | |

Method Summary

<table>
<thead>
<tr>
<th>Binder&lt;Node&gt;</th>
<th>createBinder()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates a Binder for W3C DOM.</td>
<td></td>
</tr>
</tbody>
</table>

createBinder(Class<T> domType)
   Creates a Binder object that can be used for associative/in-place unmarshalling/marshalling. JAXBIntrospector createJAXBIntrospector()
Creates a JAXBIntrospector object that can be used to introspect JAXB objects. abstract Marshaller createMarshaller()

Create a Marshaller object that can be used to convert a java content tree into XML data. abstract Unmarshaller createUnmarshaller()

Create an Unmarshaller object that can be used to convert XML data into a java content tree. abstract Validator createValidator()

Deprecated. since JAXB2.0 void generateSchema(SchemaOutputResolver outputResolver)

Generates the schema documents for this context. static JAXBContext newInstance(Class... classesToBeBound)

Obtain a new instance of a JAXBContext class. static JAXBContext newInstance(Class[] classesToBeBound, Map<String,?> properties)

Obtain a new instance of a JAXBContext class. static JAXBContext newInstance(String contextPath)

Obtain a new instance of a JAXBContext class. static JAXBContext newInstance(String contextPath, ClassLoader classLoader)

Obtain a new instance of a JAXBContext class. static JAXBContext newInstance(String contextPath, ClassLoader classLoader, Map<String,?> properties)

Obtain a new instance of a JAXBContext class.

### Methods inherited from class java.lang.Object

| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |

---

### Field Detail

#### JAXB_CONTEXT_FACTORY

public static final String JAXB_CONTEXT_FACTORY

The name of the property that contains the name of the class capable of creating new JAXBContext objects.

See Also:

[Constant Field Values](#)
Constructor Detail

protected JAXBContext()

JAXBContext

protected JAXBContext()

Method Detail

public static JAXBContext newInstance(String contextPath) throws JAXBException

    JAXBContext
    newInstance
    Thread.setContextClassLoader()

api

newInstance

JAXBException: JAXBContext

    1. ObjectFactory.class  jaxb.index
    2. contextPath
    3. contextPath
    4. contextPath

newInstance

public static JAXBContext newInstance(String contextPath) throws JAXBException

    Obtain a new instance of a JAXBContext class.

    This is a convenience method for the newInstance method. It uses
the context class loader of the current thread. To specify the use of a
different class loader, either set it via the
Thread.setContextClassLoader() api or use the newInstance method.

Throws:

JAXBException - if an error was encountered while creating the
JAXBContext such as

1. failure to locate either ObjectFactory.class or jaxb.index in
   the packages
2. an ambiguity among global elements contained in the
   contextPath
3. failure to locate a value for the context factory provider
   property
4. mixing schema derived packages from different providers
   on the same contextPath

public static JAXBContext newInstance(String
contextPath, ClassLoader classLoader) throws
JAXBException

JAXBContext

(';', :) java / JAXB
ObjectFactory.class JAXBContext JAXB
jaxb.index java JAXB java
JAXB JLS 3 7.4.1 “”

countexPath

1. ObjectFactory.class
2. jaxb.index

jaxb.index

'#$' (0x23) UTF-8 JAXBContext
newInstance(Class...) 

jaxb.index

- ".class"
- jaxb.index
- jaxb.index

JAXB 1.0 java /
valueClass="false", JAXB jaxb.properties
javax.xml.bind.context.factory

JAXB contextPath XML JAXBException

JAXB /
contextPath / java JAXB java
classLoader
return JAXBContext

JAXBException: JAXBContext

1. ObjectFactory.class jaxb.index

Throws
2. contextPath
3.
4. contextPath

newInstance

public static JAXBContext newInstance(String contextPath, ClassLoader classLoader)
throws JAXBException

Obtain a new instance of a JAXBContext class.

The client application must supply a context path which is a list of colon (':', \u003A) separated java package names that contain
schema-derived classes and/or fully qualified JAXB-annotated classes. Schema-derived code is registered with the JAXBContext by the ObjectFactory.class generated per package. Alternatively than being listed in the context path, programmer annotated JAXB mapped classes can be listed in a jaxb.index resource file, format described below. Note that a java package can contain both schema-derived classes and user annotated JAXB classes. Additionally, the java package may contain JAXB package annotations that must be processed. (see JLS 3rd Edition, Section 7.4.1. "Package Annotations").

Every package listed on the contextPath must meet **one or both** of the following conditions otherwise a JAXBException will be thrown:

1. it must contain ObjectFactory.class
2. it must contain jaxb.index

**Format for jaxb.index**

The file contains a newline-separated list of class names. Space and tab characters, as well as blank lines, are ignored. The comment character is '#' (0x23); on each line all characters following the first comment character are ignored. The file must be encoded in UTF-8. Classes that are reachable, as defined in newInstance(Class...), from the listed classes are also registered with JAXBContext.

Constraints on class name occuring in a jaxb.index file are:

- Must not end with ".class".
- Class names are resolved relative to package containing jaxb.index file. Only classes occuring directly in package containing jaxb.index file are allowed.
- Fully qualified class names are not allowed. A qualified class name,relative to current package, is only allowed to specify a nested or inner class.

To maintain compatibility with JAXB 1.0 schema to java interface/implementation binding, enabled by schema customization, the JAXB provider will ensure that each package on the context path has a jaxb.properties file which contains a value for the
javax.xml.bind.context.factory property and that all values resolve to the same provider. This requirement does not apply to JAXB annotated classes.

If there are any global XML element name collisions across the various packages listed on the contextPath, a JAXBException will be thrown.

Mixing generated interface/implement bindings from multiple JAXB Providers in the same context path may result in a JAXBException being thrown.

**Parameters:**
- `contextPath` - list of java package names that contain schema derived class and/or java to schema (JAXB-annotated) mapped classes
- `classLoader` - This class loader will be used to locate the implementation classes.

**Returns:**
- a new instance of a JAXBContext

**Throws:**
- JAXBException - if an error was encountered while creating the JAXBContext such as
  1. failure to locate either ObjectFactory.class or jaxb.index in the packages
  2. an ambiguity among global elements contained in the contextPath
  3. failure to locate a value for the context factory provider property
  4. mixing schema derived packages from different providers on the same contextPath

```java
newInstance

public static JAXBContext newInstance(String contextPath, ClassLoader classLoader, Map<String,?> properties)
```
Obtain a new instance of a JAXBContext class.

This is mostly the same as `newInstance(String, ClassLoader)`, but this version allows you to pass in provider-specific properties to configure the instanciation of JAXBContext.

The interpretation of properties is up to implementations.

**Parameters:**
- `contextPath` - list of java package names that contain schema derived classes
- `classLoader` - This class loader will be used to locate the implementation classes.
- `properties` - provider-specific properties

**Returns:**
a new instance of a JAXBContext

**Throws:**
- `JAXBException` - if an error was encountered while creating the JAXBContext such as
  1. failure to locate either ObjectFactory.class or jaxb.index in the packages
  2. an ambiguity among global elements contained in the contextPath
  3. failure to locate a value for the context factory provider property
  4. mixing schema derived packages from different providers on the same contextPath

**Since:**
JAXB2.0

```java
public static JAXBContext newInstance(Class<T>[] classesToBeBound) throws JAXBException

JAXBContext
```
newInstance(Foo.class)  JAXBContext  Foo
Bar  Zot  FooBar

class Foo {
    @XmlTransient FooBar c;
    Bar b;
}
class Bar { int x; }
class Zot extends Bar { int y; }
class FooBar { }

JAXBContext java JLS 3 7.4.1

```
newInstance null

JAXBContext
JAXBException

return

JAXB

Throws

1. JAXB
2. JAXB
3. JAXB
4. JAXB

Throws

IllegalArgumentException:

null

newInstance(null);

since

JAXB2.0

newInstance

public static JAXBContext newInstance(Class... classesToBeBound) throws JAXBException

Obtain a new instance of a JAXBContext class.

The client application must supply a list of classes that the new context object needs to recognize. Not only the new context will
recognize all the classes specified, but it will also recognize any
classes that are directly/indirectly referenced statically from the
specified classes. Subclasses of referenced classes nor
@XmlTransient referenced classes are not registered with
JAXBContext. For example, in the following Java code, if you do
newInstance(Foo.class), the newly created JAXBContext will
recognize both Foo and Bar, but not Zot Or FooBar:

class Foo {
    @XmlTransient FooBar c;
    Bar b;
}
class Bar { int x; }
class Zot extends Bar { int y; }
class FooBar { }

Therefore, a typical client application only needs to specify the top-
level classes, but it needs to be careful.

Note that for each java package registered with JAXBContext, when
the optional package annotations exist, they must be processed.
(see JLS 3rd Edition, Section 7.4.1. "Package Annotations").

Parameters:
    classesToBeBound - list of java classes to be recognized by the
new JAXBContext. Can be empty, in which case a JAXBContext
that only knows about spec-defined classes will be returned.

Returns:
A new instance of a JAXBContext. Always non-null valid object.

Throws:
    JAXBException - if an error was encountered while creating the
JAXBContext, such as (but not limited to):
    1. No JAXB implementation was discovered
    2. Classes use JAXB annotations incorrectly
    3. Classes have colliding annotations (i.e., two classes with
the same type name)
    4. The JAXB implementation was unable to locate provider-
specific out-of-band information (such as additional files
generated at the development time.)
    IllegalArgumentException - if the parameter contains null (i.e.,
newInstance

public static JAXBContext newInstance(Class[] classesToBeBound, Map<String,?> properties)
    throws JAXBException

Obtain a new instance of a JAXBContext class.

An overloading of newInstance(Class...) to configure 'properties' for this instantiation of JAXBContext.

The interpretation of properties is implementation specific.

Parameters:
  classesToBeBound - list of java classes to be recognized by the new JAXBContext. Can be empty, in which case a JAXBContext that only knows about spec-defined classes will be returned.

Returns:
  A new instance of a JAXBContext. Always non-null valid object.

Throws:
  JAXBException - if an error was encountered while creating the JAXBContext, such as (but not limited to):
  1. No JAXB implementation was discovered
  2. Classes use JAXB annotations incorrectly
  3. Classes have colliding annotations (i.e., two classes with the same type name)
  4. The JAXB implementation was unable to locate provider-specific out-of-band information (such as additional files generated at the development time.)
  IllegalArgumentException - if the parameter contains null (i.e., newInstance(null);

Since:
  JAXB2.0
abstract public **Unmarshal**ler **createUnm**arshaller() throws **JAXBException**

**XML** return **Unmarshal**ler

**Throws** **JAXBException**: Unmarshaller

**createUnmarshall**er

public abstract **Unmarshal**ler **createUnmarsh**aller() throws **JAXBException**

Create an **Unmarshal**ler object that can be used to convert XML data into a java content tree.

**Returns:**

an **Unmarshal**ler object

**Throws:**

**JAXBException** - if an error was encountered while creating the **Unmarshal**ler object

---

abstract public **Marsh**aller **createMarsh**aller() throws **JAXBException**

**java** return **Marsh**aller

**Throws** **JAXBException**: **Marsh**aller

**createMarsh**aller

public abstract **Marsh**aller **createMarsh**aller() throws **JAXBException**

Create a **Marsh**aller object that can be used to convert a java content tree into XML data.
abstract public **Validator** createValidator() throws **JAXBException**

**Validator** JAXB 2.0

**validator javadoc**

**createValidator**

```java
Validator createValidator() throws JAXBException
```

**Deprecated. since JAXB2.0**

**Validator** has been made optional and deprecated in JAXB 2.0. Please refer to the javadoc for **Validator** for more detail.

Create a **Validator** object that can be used to validate a java content tree against its source schema.

**Returns:**
- a **Validator** object

**Throws:**
- **JAXBException** - if an error was encountered while creating the **Validator** object
createBinder

public <T> Binder<T> createBinder(Class<T> domType)

Creates a Binder object that can be used for associative/in-place unmarshalling/marshalling.

Parameters:
   domType - select the DOM API to use by passing in its DOM Node class.

Returns:
   always a new valid Binder object.

Throws:
   UnsupportedOperationException - if DOM API corresponding to domType is not supported by the implementation.

Since:
   JAXB2.0

createBinder

public ">Binder createBinder()
W3C DOM

return Binder
since Binder
since JAXB2.0

createBinder

public Binder<Node> createBinder()

Creates a Binder for W3C DOM.

Returns:
   always a new valid Binder object.

Since:
   JAXB2.0
public JAXBIntrospector createJAXBIntrospector()

JAXB

return null

JAXBIntrospector

Throws

UnsupportedOperationException: JAXB 1.0

UnsupportedOperationException

since

JAXB2.0

createJAXBIntrospector

public JAXBIntrospector createJAXBIntrospector()

Creates a JAXBIntrospector object that can be used to introspect JAXB objects.

Returns:
always return a non-null valid JAXBIntrospector object.

Throws:

UnsupportedOperationException - Calling this method on JAXB 1.0 implementations will throw an

UnsupportedOperationException.

Since:
JAXB2.0

generateSchema

public void generateSchema(SchemaOutputResolver outputResolver) throws java.io.IOException

outputResolver

Throws

java.io.IOException: SchemaOutputResolver

IOException

Throws

UnsupportedOperationException: JAXB 1.0

UnsupportedOperationException

since

JAXB 2.0

generateSchema
public void generateSchema(SchemaOutputResolver outputResolver)
  throws IOException

Generates the schema documents for this context.

Parameters:
  outputResolver - this object controls the output to which
                    schemas will be sent.

Throws:
  IOException - if SchemaOutputResolver throws an IOException.
  UnsupportedOperation - Calling this method on JAXB 1.0
                        implementations will throw an
                        UnsupportedOperationException.

Since:
  JAXB 2.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.bind Class JAXBElement<T>

java.lang.Object
   - javax.xml.bind.JAXBElement<T>

All Implemented Interfaces:
   Serializable

public class JAXBElement<T>
   extends Object
   implements Serializable

Implements: java.io.Serializable
Inner classes: JAXBElement.GlobalScope

Xml JAXB

Xml xml

   • xml name
   • value
   • declaredType xs:element @type
   • scope
   • boolean nil xsi:nil
declaredType scope xml JAXB

Scope  GlobalScope Java

   value null nil true nil nil true value
   null nil

   since  JAXB 2.0

JAXB representation of an Xml Element.
This class represents information about an Xml Element from both the element declaration within a schema and the element instance value within an xml document with the following properties

- element's xml tag **name**
- **value** represents the element instance's attribute(s) and content model
- element declaration's **declaredType** (xs:element @type attribute)
- **scope** of element declaration
- boolean **nil** property. (element instance's xsi:nil attribute)

The **declaredType** and **scope** property are the JAXB class binding for the xml type definition.

**Scope** is either **JAXBElement.GlobalScope** or the Java class representing the complex type definition containing the schema element declaration.

There is a property constraint that if **value** is null, then **nil** must be true. The converse is not true to enable representing a nil element with attribute(s). If **nil** is true, it is possible that **value** is non-null so it can hold the value of the attributes associated with a nil element.

**Since:**
JAXB 2.0

**Author:**
Kohsuke Kawaguchi, Joe Fialli

**See Also:**
Serialized Form

---

### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th><strong>JAXBElement.GlobalScope</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Designates global scope for an xml element.</td>
</tr>
</tbody>
</table>

### Field Summary

<table>
<thead>
<tr>
<th>protected Class&lt;T&gt;</th>
<th><strong>declaredType</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Java datatype binding for xml element declaration's type.</td>
</tr>
</tbody>
</table>
protected `QName` name
   xml element tag name

protected boolean nil
   true iff the xml element instance has xsi:nil="true".

protected `Class` scope
   Scope of xml element declaration representing this xml element instance.

protected `T` value
   xml element value.

## Constructor Summary

**JAXBElement**(`QName` name, `Class<T>` declaredType, `Class` scope, `T` value)
   Construct an xml element instance.

**JAXBElement**(`QName` name, `Class<T>` declaredType, `T` value)
   Construct an xml element instance.

## Method Summary

**Class<T>**

`getDeclaredType()`
   Returns the Java binding of the xml element declaration's type attribute.

**QName**

`getName()`
   Returns the xml element tag name.

**Class**

`getScope()`
   Returns scope of xml element declaration.

**T**

`getValue()`
   Return the content model and attribute values for this element.

**boolean**

`isGlobalScope()`
   Returns true iff this xml element declaration is global.

**boolean**

`isNil()`
   Returns true iff this element instance content model is nil.

`isTypeSubstituted()`
<table>
<thead>
<tr>
<th>boolean</th>
<th>Returns true iff this xml element instance's value has a different type than xml element declaration's declared type.</th>
</tr>
</thead>
</table>
| void | **setNil**(boolean value)  
Set whether this element has nil content. |
| void | **setValue**(T t)  
Set the content model and attributes of this xml element. |

**Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

**Field Detail**

**name**

- protected final **QName** name
  
  xml element tag name

---

**declaredType**

- protected final **Class**<T> declaredType
  
  Java datatype binding for xml element declaration's type.

---

**scope**

- protected final **Class** scope
Scope of xml element declaration representing this xml element instance. Can be one of the following values: - `JAXBElement.GlobalScope` for global xml element declaration. - local element declaration has a scope set to the Java class representation of complex type definition containing xml element declaration.

value

protected `T` **value**

xml element value. Represents content model and attributes of an xml element instance.

nil

protected boolean **nil**

true iff the xml element instance has xsi:nil="true".

### Constructor Detail

**JAXBElement**

```java
class JAXBElement {
    public JAXBElement(QName name, Class<T> declaredType, Class scope, T value)
```

Construct an xml element instance.

**Parameters:**
name - Java binding of xml element tag name
declaredType - Java binding of xml element declaration's type
scope - Java binding of scope of xml element declaration.
Passing null is the same as passing GlobalScope.class
default - Java instance representing xml element's value.

See Also:
getScope(), isTypeSubstituted()

**JAXBElement**

```java
public JAXBElement(QName name, Class<T> declaredType, T value)
```

Construct an xml element instance. This is just a convenience method for
new JAXBElement(name,declaredType,GlobalScope.class,value)

```
Method Detail
```

```java
public Class<T> getDeclaredType()
```

getDeclaredType

```java
public Class<T> getDeclaredType()
```

Returns the Java binding of the xml element declaration's type attribute.

```java
public javax.xml.namespace.QName getName()
```

getName
getName

public QName getName()

    Returns the xml element tag name.

setValue

public void setValue(T t)

    Set the content model and attributes of this xml element.

    When this property is set to null, isNil() must be true. Details of
    constraint are described at isNil().

    See Also:
    isTypeSubstituted()

public Object getValue()

    null #isNil()

getValue

public T getValue()

    Return the content model and attribute values for this element.

    See isNil() for a description of a property constraint when this value
    is null
public Class<T> getScope()

xml

    return GlobalScope.class

See also isGlobalScope()

getScope

public Class getScope()

    Returns scope of xml element declaration.

    Returns: GlobalScope.class if this element is of global scope.

    See Also: isGlobalScope()

public boolean isNil()

    nil true

    #getValue() null true true #getValue()

    null nil xml

isNil

public boolean isNil()

    Returns true iff this element instance content model is nil.

    This property always returns true when getValue() is null. Note that
the converse is not true, when this property is true, getValue() can
contain a non-null value for attribute(s). It is valid for a nil xml
element to have attribute(s).
public void setNil(boolean value)

nil

See also

setNil

public void setNil(boolean value)

Set whether this element has nil content.

See Also:

isNil()

-----------

public boolean isGlobalScope()

xml true

isGlobalScope

public boolean isGlobalScope()

Returns true iff this xml element declaration is global.

-----------

public boolean isTypeSubstituted()

xml xml true

isTypeSubstituted

public boolean isTypeSubstituted()

Returns true iff this xml element instance's value has a different type than xml element declaration's declared type.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind  **Class JAXBEElement.GlobalScope**

java.lang.Object  
   javax.xml.bind.JAXBEElement.GlobalScope

Enclosing class:  
   JAXBEElement<T>

public static final class JAXBEElement.GlobalScope extends Object

Contained within: "JAXBEElement"

xml

Designates global scope for an xml element.

Constructor Summary

| JAXBEElement.GlobalScope() |

Method Summary

| Methods inherited from class java.lang.Object  |
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |

Constructor Detail

public JAXBEElement.GlobalScope()
JAXBElemetn.GlobalScope

public JAXBElemetn.GlobalScope();

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind  **Class JAXBException**

**java.lang.Object**
   ↳ **java.lang.Throwable**
      ↳ **java.lang.Exception**
         ↳ **javax.xml.bind.JAXBException**

**All Implemented Interfaces:**
   **Serializable**

**Direct Known Subclasses:**
   **MarshalException, PropertyException, UnmarshalException, ValidationException**

---

public class **JAXBException**

extends **Exception**

**Extends:** Throwable > Exception
**Extended by:** MarshalException, PropertyException, UnmarshalException, ValidationException

JAXB

**version** $Revision: 1.1 $ $Date: 2004/12/14 21:50:39$

**since** JAXB1.0

**See** [javax.xml.bind.JAXBContext](#), [javax.xml.bind.Marshaller](#), [javax.xml.bind.Unmarshaller](#)

**also** [javax.xml.bind.Marshaller](#), [javax.xml.bind.Unmarshaller](#)

This is the root exception class for all JAXB exceptions.

**Since:**
   JAXB1.0

**Version:**
   $Revision: 1.1 $ $Date: 2004/12/14 21:50:39$

**Author:**
   - Ryan Shoemaker, Sun Microsystems, Inc.
See Also:
JAXBContext, Marshaller, Unmarshaller, Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXBException(String message)</td>
<td>Construct a JAXBException with the specified detail message.</td>
</tr>
<tr>
<td>JAXBException(String message, String errorCode)</td>
<td>Construct a JAXBException with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td>JAXBException(String message, String errorCode, Throwable exception)</td>
<td>Construct a JAXBException with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td>JAXBException(String message, Throwable exception)</td>
<td>Construct a JAXBException with the specified detail message and linkedException.</td>
</tr>
<tr>
<td>JAXBException(Throwable exception)</td>
<td>Construct a JAXBException with a linkedException.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable getCause()</td>
<td></td>
</tr>
<tr>
<td>String getErrorCode()</td>
<td>Get the vendor specific error code</td>
</tr>
<tr>
<td>Throwable getLinkedException()</td>
<td>Get the linked exception</td>
</tr>
<tr>
<td>void printStackTrace()</td>
<td>Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to System.err.</td>
</tr>
<tr>
<td>void printStackTrace(PrintStream s)</td>
<td>Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to the</td>
</tr>
</tbody>
</table>
### printStackTrace(PrintWriter s)

Prints this JAXBException and its stack trace (including the stack trace of the(linkedException if it is non-null) to the PrintWriter.

### setLinkedException(Throwable exception)

Add a linked Exception.

### toString()

Returns a short description of this JAXBException.

---

**Methods inherited from class java.lang.Throwable**

- fillInStackTrace
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- setStackTrace

---

**Methods inherited from class java.lang.Object**

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

---

### Constructor Detail

**public JAXBException(String message)**

JAXBException

**errorCode**

**linkedException**

**null**

**message**

---

**JAXBException**

**public JAXBException(String message)**

Construct a JAXBException with the specified detail message. The errorCode and linkedException will default to null.

**Parameters:**

message - a description of the exception
public JAXBException(String message, String errorCode)

errorCode JAXBException linkedException null

message
errorCode

JAXBException

public JAXBException(String message, String errorCode)

Construct a JAXBException with the specified detail message and vendor specific errorCode. The linkedException will default to null.

Parameters:
    message - a description of the exception
    errorCode - a string specifying the vendor specific error code

public JAXBException(Throwable exception)

linkedException JAXBException errorCode null

exception

JAXBException

public JAXBException(Throwable exception)

Construct a JAXBException with a linkedException. The detail message and vendor specific errorCode will default to null.

Parameters:
    exception - the linked exception

public JAXBException(String message, Throwable exception)


public JAXBException(String message, String errorCode, Throwable exception)

Construct a JAXBException with the specified detail message, vendor specific errorCode, and linkedException.

Parameters:
  - message - a description of the exception
  - errorCode - a string specifying the vendor specific error code

public JAXBException(String message, Throwable exception)

Construct a JAXBException with the specified detail message and linkedException. The errorCode will default to null.

Parameters:
  - message - a description of the exception
  - exception - the linked exception
public String getErrorCode()

    return

genericCode

genericCode

public Throwable getLinkedException()

    return null

getLinkedException

public void setLinkedException(Throwable exception)

    exception null
setLinkedException

public void setLinkedException(Throwable exception)

Add a linked Exception.

Parameters:
  exception - the linked Exception (A null value is permitted and indicates that the linked exception does not exist or is unknown).

public String toString()

JAXBException

toString

public String toString()

JAXBException

Returns a short description of this JAXBException.

Overrides:
  toString in class Throwable

public void printStackTrace(java.io.PrintStream s)

JAXBException PrintStream null

linkedException
  s          PrintStream

printStackTrace

public void printStackTrace(PrintStream s)

Prints this JAXBException and its stack trace (including the stack
trace of the linkedException if it is non-null) to the PrintStream.

Overrides:

printStackTrace in class Throwable

Parameters:

s - PrintStream to use for output

public void printStackTrace()
JAXBException System.err null
linkedException

printStackTrace

public void printStackTrace()

Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to System.err.

Overrides:

printStackTrace in class Throwable

public void printStackTrace(java.io.PrintWriter s)
JAXBException PrintWriter null
linkedException
s PrintWriter

printStackTrace

public void printStackTrace(PrintWriter s)

Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to the PrintWriter.

Overrides:
printStackTrace in class Throwable

Parameters:
   s - PrintWriter to use for output

getCause

public Throwable getCause()

Overrides:
   getCause in class Throwable

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature

PS:
javax.xml.bind  **Class JAXBIntrospector**

```java
java.lang.Object
  `javax.xml.bind.JAXBIntrospector`
```

```java
public abstract class JAXBIntrospector extends Object
```

JAXB  JAXB xml

JAXB xml java schema schema java  
JAXB XML  
createJAXBIntrospector()  

```java
since  JAXB2.0  
See also  createJAXBIntrospector()
```

Provide access to JAXB xml binding data for a JAXB object.

Initially, the intent of this class is to just conceptualize how a JAXB application developer can access xml binding information, independent if binding model is java to schema or schema to java. Since accessing the XML element name related to a JAXB element is a highly requested feature, demonstrate access to this binding information. The factory method to get a JAXBIntrospector instance is `JAXBContext.createJAXBIntrospector()`.

**Since:**  
JAXB2.0  
**See Also:**  
`JAXBContext.createJAXBIntrospector()`

---

**Constructor Summary**
JAXBIntrospector

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract QName</td>
<td><code>getElementName(Object jaxbElement)</code></td>
<td>Get xml element qname for jaxbElement.</td>
</tr>
<tr>
<td>static Object</td>
<td><code>getValue(Object jaxbElement)</code></td>
<td>Get the element value of a JAXB element.</td>
</tr>
<tr>
<td>abstract boolean</td>
<td><code>isElement(Object object)</code></td>
<td>Return true iff object represents a JAXB element.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructor Detail

public JAXBIntrospector()

JAXBIntrospector

public JAXBIntrospector()

Method Detail

abstract public boolean isElement(Object object)

object JAXB true

object JAXB
1.(javax.xml.bind.JAXBElement)
2.(object)

See also

getElementName(Object)

isElement

public abstract boolean isElement(Object object)

Return true iff object represents a JAXB element.

Parameter object is a JAXB element for following cases:

1. It is an instance of javax.xml.bind.JAXBElement.
2. The class of object is annotated with @XmlRootElement.

See Also:

g getElementName(Object)

abstract public javax.xml.namespace.QName
getElementName(Object jaxbElement)

jaxbElement xml qname

jaxbElement #isElement(Object) true
return jaxbElement xml qname
JAXB null

getElementName

public abstract QName getElementName(Object jaxbElement)

Get xml element qname for jaxbElement.

Parameters:
jaxbElement - is an object that isElement(Object) returned true.

Returns:
xml element qname associated with jaxbElement; null if jaxbElement is not a JAXB Element.

public static Object getValue(Object jaxbElement)

JAXB

javax.xml.bind.JAXBElemet >@XmlRootElement
Java

jaxbElement #isElement(Object) true
return jaxbElement

getValue

public static Object getValue(Object jaxbElement)

Get the element value of a JAXB element.

Convenience method to abstract whether working with either a javax.xml.bind.JAXBElemet instance or an instance of @XmlRootElement annotated Java class.

Parameters:
jaxbElement - object that #isElement(Object) returns true.

Returns:
The element value of the jaxbElement.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>

SUMMARY:
NESTED | FIELD | CONSTR | METHOD

DETAIL:
FIELD | CONSTR | METHOD
javax.xml.bind.util Class JAXBResult

java.lang.Object  
   ↓ javax.xml.transform.sax.SAXResult  
   ↓ javax.xml.bind.util.JAXBResult

All Implemented Interfaces:
   Result

public class JAXBResult
extends SAXResult

Extends: javax.xml.transform.sax.SAXResult

JAXB JAXP  
   ↓ javax.xml.transform.Result

JAXB Java/XML

JAXB XSLT

   JAXBResult result = new JAXBResult(JAXBContext.newInstance("org.acme.foo"));

   // set up XSLT transformation
   TransformerFactory tf = TransformerFactory.newInstance();
   Transformer t = tf.newTransformer(new StreamSource("test.xsl"));

   // run transformation
   t.transform(new StreamSource("document.xml"),result);

   // obtain the unmarshalled content tree
   Object o = result.getResult();

SAXResult JAXBResult SAXResult

setHandler setLexicalHandler setSystemId
JAXP Result implementation that unmarshals a JAXB object.

This utility class is useful to combine JAXB with other Java/XML technologies.

The following example shows how to use JAXB to unmarshal a document resulting from an XSLT transformation.

```java
JAXBResult result = new JAXBResult(
    JAXBContext.newInstance("org.acme.foo")
);

// set up XSLT transformation
TransformerFactory tf = TransformerFactory.newInstance();
Transformer t = tf.newTransformer(new StreamSource("test.xs"));

// run transformation
    t.transform(new StreamSource("document.xml"),result);

// obtain the unmarshalled content tree
Object o = result.getResult();
```

The fact that JAXBResult derives from SAXResult is an implementation detail. Thus in general applications are strongly discouraged from accessing methods defined on SAXResult.

In particular it shall never attempt to call the setHandler, setLexicalHandler, and setSystemId methods.

**Author:**
Kohsuke Kawaguchi (kohsuke.kawaguchi@sun.com)

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class <code>javax.xml.transform.sax.SAXResult</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>FEATURE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface <code>javax.xml.transform.Result</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>PI_DISABLE_OUTPUT_ESCAPING, PI_ENABLE_OUTPUT_ESCAPING</td>
</tr>
</tbody>
</table>
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JAXBResult</strong>(JAXBContext context)</td>
<td>Creates a new instance that uses the specified JAXBContext to unmarshal.</td>
</tr>
<tr>
<td><strong>JAXBResult</strong>(Unmarshaller _unmarshaller)</td>
<td>Creates a new instance that uses the specified Unmarshaller to unmarshal an object.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getResult</strong>()</td>
<td>Gets the unmarshalled object created by the transformation.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.xml.transform.sax.SAXResult

- getHandler, getLexicalHandler, getSystemId, setHandler, setLexicalHandler, setSystemId

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

```java
public JAXBResult(JAXBContext context) throws JAXBException
```

**JAXBContext**

```java
context Unmarshaller JAXBContext null
```

**Throws**

- JAXBException: JAXBResult null

JAXBResult
public JAXBResult(JAXBContext context) throws JAXBException

Creates a new instance that uses the specified JAXBContext to unmarshal.

**Parameters:**
- context - The JAXBContext that will be used to create the necessary Unmarshaller. This parameter must not be null.

**Throws:**
- JAXBException - if an error is encountered while creating the JAXBResult or if the context parameter is null.

public JAXBResult(Unmarshaller _unmarshaller) throws JAXBException

Unmarshal

**JAXBResult Unmarshaller Unmarshaller Unmarshaller**

Unmarshal

**Unmarshaller JAXBContext**

_unmarshaller    Unmarshaller null

**Throws**
- JAXBException: JAXBResult Unmarshaller null

**JAXBResult**

public JAXBResult(Unmarshaller _unmarshaller) throws JAXBException

Creates a new instance that uses the specified Unmarshaller to unmarshal an object.

This JAXBResult object will use the specified Unmarshaller instance. It is the caller's responsibility not to use the same Unmarshaller for other purposes while it is being used by this object.
The primary purpose of this method is to allow the client to configure Unmarshaller. Unless you know what you are doing, it's easier and safer to pass a JAXBContext.

**Parameters:**

_unmarshaller - the unmarshaller. This parameter must not be null.

**Throws:**

[JAXBException](#) - if an error is encountered while creating the JAXBResult or the Unmarshaller parameter is null.

**Method Detail**

```java
public Object getResult() throws JAXBException
```

```java
return null

Throws

IllegalStateException: Always return a non-null object.

Throws

JAXBException: SAXException
```

**getResult**

```java
public Object getResult() throws JAXBException
```

Gets the unmarshalled object created by the transformation.

**Returns:**

Always return a non-null object.

**Throws:**

[IllegalStateException](#) - if this method is called before an object is unmarshalled.

[JAXBException](#) - if there is any unmarshalling error. Note that the implementation is allowed to throw SAXException during the parsing when it finds an error.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.util Class JAXBSource

java.lang.Object
  ↓ javax.xml.transform.sax.SAXSource
   ↓ javax.xml.bind.util.JAXBSource

All Implemented Interfaces:
  Source

public class JAXBSource
extends SAXSource

Extends: javax.xml.transform.sax.SAXSource

JAXB JAXP

JAXB Java/XML

JAXB XSLT

MyObject o = // get JAXB content tree

// jaxbContext is a JAXBContext object from which 'o' is created.
JAXBSource source = new JAXBSource( jaxbContext, o );

// set up XSLT transformation
TransformerFactory tf = TransformerFactory.newInstance();
Transformer t = tf.newTransformer(new StreamSource("test.xsl"));

// run transformation
t.transform(source,new StreamResult(System.out));

SAXSource JAXBSource SAXSource
setXMLReader setInputSource getXMLReader XMLReader
getInputSource InputSource XMLReader

getInputSource InputSource getXMLReader XMLReader
JAXP Source implementation that marshals a JAXB-generated object.

This utility class is useful to combine JAXB with other Java/XML technologies.

The following example shows how to use JAXB to marshal a document for transformation by XSLT.

```java
MyObject o = // get JAXB content tree

// jaxbContext is a JAXBContext object from which 'o' is cr
JAXBSource source = new JAXBSource( jaxbContext, o );

// set up XSLT transformation
TransformerFactory tf = TransformerFactory.newInstance();
Transformer t = tf.newTransformer(new StreamSource("test.xs

// run transformation
t.transform(source,new StreamResult(System.out));
```

The fact that JAXBSource derives from SAXSource is an implementation detail. Thus in general applications are strongly discouraged from accessing methods defined on SAXSource. In particular, the setXMLReader and setInputSource methods shall never be called. The XMLReader object obtained by the getXMLReader method shall be used only for parsing the InputSource object returned by the getInputSource method.

Similarly the InputSource object obtained by the getInputSource method shall be used only for being parsed by the XMLReader object returned by the getXMLReader.

**Author:**
Kohsuke Kawaguchi (kohsuke.kawaguchi@sun.com)
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>JAXBSource(JAXBContext context, Object contentObject)</code></td>
<td>Creates a new <code>Source</code> for the given content object.</td>
</tr>
<tr>
<td><code>JAXBSource(Marshaller marshall, Object contentObject)</code></td>
<td>Creates a new <code>Source</code> for the given content object.</td>
</tr>
</tbody>
</table>

### Method Summary

#### Methods inherited from class `javax.xml.transform.sax.SAXSource`
- `getInputSource`, `getSystemId`, `getXMLReader`, `setInputSource`, `setSystemId`, `setXMLReader`, `sourceToInputSource`

#### Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Constructor Detail

```java
public JAXBSource(JAXBContext context, Object contentObject) throws JAXBException
```

- `context` - `javax.xml.transform.Source`  
- `contentObject` - JAXBContext marshaller null  
- `contentObject` - `javax.xml.transform.Source` XML null  

**Throws**
- `JAXBException`: JAXBSource null
public JAXBSource(JAXBContext context, Object contentObject) throws JAXBException

Creates a new Source for the given content object.

Parameters:
- context - JAXBContext that was used to create contentObject. This context is used to create a new instance of marshaller and must not be null.
- contentObject - An instance of a JAXB-generated class, which will be used as a Source (by marshalling it into XML). It must not be null.

Throws:
- JAXBException - if an error is encountered while creating the JAXBSource or if either of the parameters are null.

public JAXBSource(Marshaller marshaller, Object contentObject) throws JAXBException

javax.xml.transform.Source marshaller contentObject XML marshaller contentObject JAXBContext null

Throws JAXBException: JAXBSource null

JAXBSource

public JAXBSource(Marshaller marshaller, Object contentObject) throws JAXBException

Creates a new Source for the given content object.

Parameters:
- marshaller - A marshaller instance that will be used to marshal contentObject into XML. This must be created from a
JAXBContext that was used to build `contentObject` and must not be null.

`contentObject` - An instance of a JAXB-generated class, which will be used as a `Source` (by marshalling it into XML). It must not be null.

**Throws:**

`JAXBException` - if an error is encountered while creating the JAXBSource or if either of the parameters are null.
javax.xml.registry Class JAXRException

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
        └ javax.xml.registry.JAXRException

All Implemented Interfaces:
  Serializable, JAXRResponse

Direct Known Subclasses:
  InvalidRequestException, RegistryException,
  UnexpectedObjectException, UnsupportedCapabilityException

public class JAXRException

extends Exception
implements JAXRResponse

Extends: Throwable > Exception
Implements: JAXRResponse
Extended by: InvalidRequestException, RegistryException,
UnexpectedObjectException, UnsupportedCapabilityException

JAXR String

See also javax.xml.registry.JAXRResponse

Signals that a JAXR exception has occurred. It contains no members other than the standard reason String.

Author:
  Nicholas Kassem, Mark Hapner, Rajiv Mordani, Farrukh Najmi

See Also:
  JAXRResponse, Serialized Form
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected</td>
<td></td>
</tr>
<tr>
<td>Throwable</td>
<td></td>
</tr>
<tr>
<td>cause</td>
<td></td>
</tr>
</tbody>
</table>

### Fields inherited from interface javax.xml.registry.JAXRResponse

- STATUS_FAILURE
- STATUS_SUCCESS
- STATUS_UNAVAILABLE
- STATUS_WARNING

### Constructor Summary

**JAXRException()**
- Constructs a JAXRException object with no reason or embedded Throwable.

**JAXRException(String reason)**
- Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**JAXRException(String reason, Throwable cause)**
- Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

**JAXRException(Throwable cause)**
- Constructs a JAXRException object initialized with the given Throwable object.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable</td>
<td>getCause()</td>
</tr>
<tr>
<td>String</td>
<td>getMessage()</td>
</tr>
<tr>
<td>String</td>
<td>getRequestId()</td>
</tr>
<tr>
<td>int</td>
<td>getStatus()</td>
</tr>
</tbody>
</table>
**Throwable**

- **initCause( Throwable cause )**
  - Initializes the `cause` of this throwable to the specified value.

- **boolean isAvailable()**
  - Returns true if a response is available, false otherwise.

**Methods inherited from class java.lang.Throwable**

- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

**Field Detail**

- `cause`

  protected `Throwable cause`

**Constructor Detail**

- **public JAXRException()**
  - `Throwable` JAXRException

  **JAXRException**

- **public JAXRException()**
  - Constructs a JAXRException object with no reason or embedded Throwable.
public JAXRException(String reason)

JAXRException  String  reason

JAXRException

public JAXRException(String reason)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

    Parameters:
    reason - a description of what caused the exception

public JAXRException(String reason, Throwable cause)

    Throwable  String  Throwable
    reason     cause     JAXRException  Throwable

JAXRException

public JAXRException(String reason, Throwable cause)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

    Parameters:
    reason - a description of what caused the exception
    cause - a Throwable object that is to be embedded in this JAXRException object
public JAXRException(Throwable cause)

    Throwable JAXRException
    cause Exception Throwable

JAXRException

public JAXRException(Throwable cause)

    Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
    cause - the Throwable that caused this Exception

Method Detail

public String getMessage()

    JAXRException

    Throwable return JAXRException Throwable

getMessage

public String getMessage()

    Returns the detail message for this JAXRException object.

    If there is an embedded Throwable, and if the JAXRException object has no detail message of its own, this method will return the detail message from the embedded Throwable.

    Overrides:
        getMessage in class Throwable
Returns:
the error or warning message for this JAXRException or, if it has none, the message of the embedded Throwable, if there is one.

```
public Throwable getCause()

JAXRException Throwable null
return Throwable Throwable null
```

getcause

```
public Throwable getCause()

Returns the Throwable embedded in this JAXRException if there is one. Otherwise, this method returns null.

Overrides:
getcause in class Throwable

>Returns:
the embedded Throwable or null if there is none
```

```
public Throwable initCause(Throwable cause)

throwable cause cause throwable
throwable

#JAXRException(Throwable)  #JAXRException(String,Throwable)

cause #getCause() null
return Throwable
Throws IllegalArgumentException: cause
IllegalStateException: throwable
Throws #JAXRException(Throwable) #JAXRException(String,Throwable)
throwable
**initCause**

public Throwable initCause(Throwable cause)

Initializes the cause of this throwable to the specified value. (The cause is the throwable that caused this throwable to get thrown.)

This method can be called at most once. It is generally called from within the constructor, or immediately after creating the throwable. If this throwable was created with JAXRException(Throwable) or JAXRException(String, Throwable), this method cannot be called even once.

Overrides:

initCause in class Throwable

Parameters:

cause - the cause (which is saved for later retrieval by the getCause() method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

Returns:

a reference to this Throwable instance.

Throws:

IllegalArgumentException - if cause is this throwable. (A throwable cannot be its own cause.)

IllegalStateException - if this throwable was created with JAXRException(Throwable) or JAXRException(String, Throwable), or this method has already been called on this throwable.

---

**public String getRequestId()**

getRequestId

public String getRequestId()
Returns the unique id for the request that generated this response.

**Capability Level:** 0

**Specified by:**
- `getRequestId` in interface `JAXRResponse`

**Returns:**
- the request id

---

```java
public int getStatus()
```

**getStatus**

```java
public int getStatus()
```

**Description copied from interface:** `JAXRResponse`
Returns the status for this response.

**Capability Level:** 0

**Specified by:**
- `getStatus` in interface `JAXRResponse`

**Returns:**
- the status which is an integer enumerated value

**See Also:**
- `JAXRResponse.STATUS_SUCCESS`

---

```java
public boolean isAvailable() throws JAXRException
```

**isAvailable**

```java
public boolean isAvailable()
```

**isAvailable**

```java
throws JAXRException
```
Returns true if a response is available, false otherwise. This is a polling method and must not block.

**Specified by:**  
`isAvailable` in interface `JAXRResponse`

**Returns:**  
true if the response is available; false otherwise

**Throws:**  
`JAXRException` - If the JAXR provider encounters an internal error

---

**Overview** | **Package** | **Tree** | **Deprecated** | **Index** | **Help**
---|---|---|---|---|---
PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES
SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc Class JAXRPCException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.xml.rpc.JAXRPCException

All Implemented Interfaces:
  Serializable

public class JAXRPCException
  extends RuntimeException

Extends: Throwable > Exception > RuntimeException

javax.xml.rpc.JAXRPCException JAX-RPC API JAX-RPC
  version 1.0

The javax.xml.rpc.JAXRPCException is thrown from the core JAX-RPC APIs to indicate an exception related to the JAX-RPC runtime mechanisms.

Version:
  1.0

Author:
  Rahul Sharma

See Also:
  Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>ClassMethod</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXRPCException()</td>
<td>Constructs a new exception with null as its detail message.</td>
</tr>
<tr>
<td>JAXRPCException(String message)</td>
<td></td>
</tr>
</tbody>
</table>

Constructs a new exception with the specified detail message.

**JAXRPCException** *(String message, Throwable cause)*

Constructs a new exception with the specified detail message and cause.

**JAXRPCException** *(Throwable cause)*

Constructs a new JAXRPCException with the specified cause and a detail message of `(cause==null ? null : null) ?`.

### Method Summary

<table>
<thead>
<tr>
<th>Throwable</th>
<th>getLinkedCause()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the Linked cause</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

- fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

**public JAXRPCException()**

nullcause

**JAXRPCException**

**public JAXRPCException()**

Constructs a new exception with null as its detail message. The cause is not initialized.
public JAXRPCException(String message)

Constructs a new exception with the specified detail message. The cause is not initialized.

Parameters:
message - The detail message which is later retrieved using the getMessage method

public JAXRPCException(String message, Throwable cause)

Constructs a new exception with the specified detail message and cause.

Parameters:
message - The detail message which is later retrieved using the getMessage method
cause - The cause which is saved for the later retrieval throw by the getCause method
public JAXRPCException(Throwables cause)
JAXRPCException cause
    (cause
    : cause.toString() cause
    cause cause getCause
    null cause

JAXRPCException

public JAXRPCException(Throwables cause)

Constructs a new JAXRPCException with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause).

Parameters:
  cause - The cause which is saved for the later retrieval throw by the getCause method. (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

Method Detail

public Throwable getLinkedCause()

getLinkedCause

public Throwable getLinkedCause()

Gets the Linked cause

Returns:
  The cause of this Exception or null if the cause is nonexistent or unknown
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry Interface JAXRResponse

All Known Subinterfaces:
   BulkResponse

All Known Implementing Classes:
   DeleteException, FindException, InvalidRequestException, JAXRException, RegistryException, SaveException, UnexpectedObjectException, UnsupportedCapabilityException

public interface JAXRResponse

Implemented by: BulkResponse, JAXRException

JAXR

See also javax.xml.registry.JAXRException

A JAXR requests' response.

Author: Farrukh S. Najmi

See Also: JAXRException

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
STATUS_SUCCESS

static final int STATUS_SUCCESS

Status indicating a successful response.

See Also:
Constant Field Values

STATUS_WARNING

static final int STATUS_WARNING

Status indicating a successful response that included at least one warning.
STATUS_FAILURE

static final int STATUS_FAILURE

Status indicating a failure response.

See Also:
Constant Field Values

STATUS_UNAVAILABLE

static final int STATUS_UNAVAILABLE

Status indicating that the results are currently unavailable.

See Also:
Constant Field Values

Method Detail

public String getRequestId() throws JAXRException

ID

0

return ID

Throws JAXRException: JAXR
getRequestId

```java
StringgetRequestId()
throws JAXRException
```

Returns the unique id for the request that generated this response.

**Capability Level: 0**

**Returns:**
- the request id

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

```java
public int getStatus() throws JAXRException
```

0

```java
    return
```

**Throws**
- `JAXRException`: JAXR

**See also**
- `STATUS_SUCCESS`

---

getStatus

```java
int getStatus()
throws JAXRException
```

Returns the status for this response.

**Capability Level: 0**

**Returns:**
- the status which is an integer enumerated value

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error
public boolean isAvailable() throws JAXRException
true false

0

return true false

Throws JAXRException: JAXR

isAvailable

boolean isAvailable() throws JAXRException

Returns true if a response is available, false otherwise. This is a polling method and must not block.

Capability Level: 0

Returns:
true if the response is available; false otherwise

Throws:
JAXRException - If the JAXR provider encounters an internal error
PS:
javax.management.j2ee.statistics Interface
JCAConnectionPoolStats

All Superinterfaces:
   JCAConnectionStats, Stats

public interface JCAConnectionPoolStats extends JCAConnectionStats

Implements: JCAConnectionStats

JCA

Specifies the statistics provided by a JCA Connection Pool

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CountStatistic</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>CountStatistic</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>BoundedRangeStatistic</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>BoundedRangeStatistic</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>RangeStatistic</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface
javax.management.j2ee.statistics.JCAConnectionStats
getConnectionFactory, getManagedConnectionFactory, getUseTime, getWaitTime
Methods inherited from interface
javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail

public CountStatistic getCloseCount()

    return CountStatistic

getCloseCount

CountStatistic getCloseCount()

    The number of connections closed

    Returns:
    CountStatistic

public CountStatistic getCreateCount()

    return CountStatistic

getCreateCount

CountStatistic getCreateCount()

    The number of connections created

    Returns:
    CountStatistic
public BoundedRangeStatistic getFreePoolSize()

return BoundedRangeStatistic

getFreePoolSize

BoundedRangeStatistic getFreePoolSize()

The number of free connections in the pool

Returns:
BoundedRangeStatistic

-----------------------------------

public BoundedRangeStatistic getPoolSize()

return BoundedRangeStatistic

getPoolSize

BoundedRangeStatistic getPoolSize()

The size of the connection pool

Returns:
BoundedRangeStatistic

-----------------------------------

public RangeStatistic getWaitingThreadCount()

return BoundedRangeStatistic

getWaitingThreadCount

RangeStatistic getWaitingThreadCount()


getWaitingThreadCount

RangeStatistic getWaitingThreadCount()

The number of threads waiting for a connection

Returns:
BoundedRangeStatistic
javax.management.j2ee.statistics Interface JCAConnectionStats

All Superinterfaces:
Stats

All Known Subinterfaces:
JCAConnectionPoolStats

public interface JCAConnectionStats
extends Stats

Implements: Stats
Implemented by: JCAConnectionPoolStats

JCA

Specifies the statistics provided by a JCA connection

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getConnectionFactory() Returns the associated JCAConnectionFactory OBJECT_NAME</td>
</tr>
<tr>
<td>String getManagedConnectionFactory() Returns the associated JCAManagedConnectionFactory OBJECT_NAME</td>
</tr>
<tr>
<td>TimeStatistic getUseTime() Returns the time spent using a connection</td>
</tr>
<tr>
<td>TimeStatistic getWaitTime() Returns the time spent waiting for a connection to be available</td>
</tr>
</tbody>
</table>

Methods inherited from interface
getStatistic, getStatisticNames, getStatistics

Method Detail

public String getConnectionFactory()

JCAConnectionFactory OBJECT_NAME

    return OBJECT_NAME (String)

cgetConnectionFactory

String getConnectionFactory()

    Returns the associated JCAConnectionFactory OBJECT_NAME

    Returns:
    String the OBJECT_NAME of the managed object that identifies
    the connection factory for this connection

public String getManagedConnectionFactory()

JCAManagedConnectionFactory OBJECT_NAME

    return OBJECT_NAME (String)

ggetManagedConnectionFactory

String getManagedConnectionFactory()

    Returns the associated JCAManagedConnectionFactory
    OBJECT_NAME

    Returns:
    String the OBJECT_NAME of the managed object that identifies
    the managed connection factory for this connection
public TimeStatistic getWaitTime()

    return TimeStatistic

getWaitTime

TimeStatistic getWaitTime()

    Returns the time spent waiting for a connection to be available

    Returns:
    TimeStatistic

public TimeStatistic getUseTime()

    return TimeStatistic

getUseTime

TimeStatistic getUseTime()

    Returns the time spent using a connection

    Returns:
    TimeStatistic
to license terms.

PS:
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.management.j2ee.statistics Interface JCAStats

All Superinterfaces:
   Stats

public interface JCAStats
    extends Stats

Implements: Stats

JCA

Specifies statistics provided by a JCA resource

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getConnectionPools()</td>
<td>Returns an array of JCACreationPoolStats that provide statistics about the connection pools associated with the referencing JCA resource statistics</td>
</tr>
<tr>
<td>getConnections()</td>
<td>Returns an array of JCACreationStats that provide statistics about the non-pooled connections associated with the referencing JCA resource statistics</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats

getStatistic, getStatisticNames, getStatistics
public **JCAConnectionStats**[] getConnections()

**JCAConnectionStats** return **JCAConnectionStats**[]

getConnections

**JCAConnectionStats**[] getConnections()

Returns an array of JCAConnectionStats that provide statistics about the non-pooled connections associated with the referencing JCA resource statistics

**Returns:**

**JCAConnectionStats**[]

---

public **JCAConnectionPoolStats**[] getConnectionPools()

**JCAConnectionPoolStats** return **JCAConnectionPoolStats**[]

getConnectionPools

**JCAConnectionPoolStats**[] getConnectionPools()

Returns an array of JCAConnectionPool Stats that provide statistics about the connection pools associated with the referencing JCA resource statistics

**Returns:**

**JCAConnectionPoolStats**[]
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.management.j2ee.statistics Interface

JDBCConnectionPoolStats

All Superinterfaces:
   JDBCConnectionStats, Stats

public interface JDBCConnectionPoolStats
extends JDBCConnectionStats

Implements: JDBCConnectionStats

JDBC

Specifies the statistics provided by a JDBC connection pool.

Method Summary

<table>
<thead>
<tr>
<th>Statistic Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CountStatistic</td>
<td>getCloseCount()</td>
<td>Number of connections closed.</td>
</tr>
<tr>
<td>CountStatistic</td>
<td>getCreateCount()</td>
<td>Number of connections created.</td>
</tr>
<tr>
<td>BoundedRangeStatistic</td>
<td>getFreePoolSize()</td>
<td>Number of free connections in the pool.</td>
</tr>
<tr>
<td>BoundedRangeStatistic</td>
<td>getPoolSize()</td>
<td>Size of the connection pool.</td>
</tr>
<tr>
<td>RangeStatistic</td>
<td>getWaitingThreadCount()</td>
<td>Number of threads waiting for a connection.</td>
</tr>
</tbody>
</table>

Methods inherited from interface
javax.management.j2ee.statistics.JDBCConnectionStats
getJdbcDataSource, getUseTime, getWaitTime
Methods inherited from interface
javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail

public CountStatistic getCreateCount()

getCreateCount

CountStatistic getCreateCount()

Number of connections created.

public CountStatistic getCloseCount()

getCloseCount

CountStatistic getCloseCount()

Number of connections closed.

public BoundedRangeStatistic getPoolSize()

getPoolSize

BoundedRangeStatistic getPoolSize()
Size of the connection pool.

```java
public BoundedRangeStatistic getFreePoolSize()
```

**getFreePoolSize**

`BoundedRangeStatistic getFreePoolSize()`

Number of free connections in the pool.

```java
public RangeStatistic getWaitingThreadCount()
```

**getWaitingThreadCount**

`RangeStatistic getWaitingThreadCount()`

Number of threads waiting for a connection.
javax.management.j2ee.statistics Interface

JDBCConnectionStats

All Superinterfaces:
   Stats

All Known Subinterfaces:
   JDBCConnectionPoolStats

public interface JDBCConnectionStats extends Stats

Implements: Stats
Implemented by: JDBCConnectionPoolStats

JDBC

Specifies the statistics provided by all (pooled and non-pooled) JDBC connections.

Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getJdbcDataSource()</td>
<td>Identifies the JDBC driver for the corresponding JDBCConnection.</td>
</tr>
<tr>
<td>TimeStatistic</td>
<td>getUseTime()</td>
<td>Time spent using a connection.</td>
</tr>
<tr>
<td>TimeStatistic</td>
<td>getWaitTime()</td>
<td>Time spent waiting for a connection to be available.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics Stats
getStatistic, getStatisticNames, getStatistics
Method Detail

public String getJdbcDataSource()

getJdbcDataSource

String getJdbcDataSource()

Identifies the JDBC driver for the corresponding JDBCConnection.

public TimeStatistic getWaitTime()

getWaitTime

TimeStatistic getWaitTime()

Time spent waiting for a connection to be available.

public TimeStatistic getUseTime()

getUseTime

TimeStatistic getUseTime()

Time spent using a connection.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.management.j2ee.statistics Interface JDBCStats

All Superinterfaces:

Stats

public interface JDBCStats extends Stats

Implements: Stats

JDBC

Statistics provided by a JDBC resource

Author:
Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>JDBCConnectionPoolStats[] getConnectionPools()</td>
</tr>
<tr>
<td>JDBCConnectionStats[] getConnections()</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats

getStatistic, getStatisticNames, getStatistics

Method Detail

public JDBCConnectionStats[] getConnections()
getConnections

JDBCConnectionStats[] getConnections()

public JDBCConnectionPoolStats[] getConnectionPools()

getConnectionPools

JDBCConnectionPoolStats[] getConnectionPools()
javax.management.j2ee.statistics Interface JMSConnectionStats

All Superinterfaces:
   Stats

public interface JMSConnectionStats
   extends Stats

Implements: Stats

JMS

Specifies the statistics provided by a JMS connection

Author:
   Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| **getSessions()**
| Returns an array of JMSSessionStats that provide statistics about the sessions associated with the referencing JMSConnectionStats. |
| **isTransactional()**
| Returns the transactional state of this JMS connection. |

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics
public JMSSessionStats[] getSessions()
JMSSessionStats JMSConnectionStats

gETCHENS

Returns an array of JMSSessionStats that provide statistics about the sessions associated with the referencing JMSConnectionStats.

public boolean isTransactional()
JMS true JMS

isTransactional

boolean isTransactional()

Returns the transactional state of this JMS connection. If true, indicates that this JMS connection is transactional.
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
javax.management.j2ee.statistics Interface JMSConsumerStats

All Superinterfaces:  
  JMSConsumerStats, Stats

public interface JMSConsumerStats
extends JMSConsumerStats

Implements: JMSConsumerStats

JMS

Specifies the statistics provided by a JMS message consumer

Author:  
  Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getOrigin</strong>()</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.JMSConsumerStats

getExpiredMessageCount, getMessageCount, getMessageWaitTime, getPendingMessageCount

Methods inherited from interface javax.management.j2ee.statistics.Stats

getStatistic, getStatisticNames, getStatistics
public String getOrigin()

getOrigin

String getOrigin()

Returns a string that encapsulates the identity of a message origin.
javax.management.j2ee.statistics Interface JMS Endpoint Stats

All Superinterfaces:
    Stats

All Known Subinterfaces:
    JMS Consumer Stats, JMS Producer Stats

public interface JMS Endpoint Stats
extends Stats

Implements: Stats
Implemented by: JMS Consumer Stats, JMS Producer Stats

JMS  JMS

Specifies the statistics provided by a JMS message producer or a JMS message consumer.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CountStatistic getExpiredMessageCount()</td>
<td>Number of messages that expired before delivery.</td>
</tr>
<tr>
<td>CountStatistic getMessageCount()</td>
<td>Number of messages sent or received.</td>
</tr>
<tr>
<td>TimeStatistic getMessageWaitTime()</td>
<td>Time spent by a message before being delivered.</td>
</tr>
<tr>
<td>CountStatistic getPendingMessageCount()</td>
<td>Number of pending messages.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats

getStatistic, getStatisticNames, getStatistics
## Method Detail

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>public CountStatistic getMessageCount()</code></td>
<td>Number of messages sent or received.</td>
</tr>
<tr>
<td><code>public CountStatistic getPendingMessageCount()</code></td>
<td>Number of pending messages.</td>
</tr>
<tr>
<td><code>public CountStatistic getExpiredMessageCount()</code></td>
<td>Number of messages that expired before delivery.</td>
</tr>
</tbody>
</table>
public TimeStatistic getMessageWaitTime()

getMessageWaitTime

TimeStatistic getMessageWaitTime()

Time spent by a message before being delivered.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jms Class JMSException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.jms.JMSException

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   IllegalStateException, InvalidClientIDException, InvalidDestinationException, InvalidSelectorException, JMSSecurityException, MessageEOFException, MessageFormatException, MessageNotReadableException, MessageNotWriteableException, ResourceAllocationException, TransactionInProgressException, TransactionRolledBackException

public class JMSException
extends Exception

Extends: Throwable > Exception
Extended by: IllegalStateException, InvalidClientIDException, InvalidDestinationException, InvalidSelectorException, JMSSecurityException, MessageEOFException, MessageFormatException, MessageNotReadableException, MessageNotWriteableException, ResourceAllocationException, TransactionInProgressException, TransactionRolledBackException

JMS API

• getMessage

•

• JMS API
This is the root class of all JMS API exceptions.

It provides the following information:

- A provider-specific string describing the error. This string is the standard exception message and is available via the `getMessage` method.
- A provider-specific string error code
- A reference to another exception. Often a JMS API exception will be the result of a lower-level problem. If appropriate, this lower-level exception can be linked to the JMS API exception.

### Constructor Summary

**`JMSException(String reason)`**

Constructs a `JMSException` with the specified reason and with the error code defaulting to null.

**`JMSException(String reason, String errorCode)`**

Constructs a `JMSException` with the specified reason and error code.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getErrorCode()</code></td>
<td>Gets the vendor-specific error code.</td>
</tr>
<tr>
<td><code>Exception getLinkedException()</code></td>
<td>Gets the exception linked to this one.</td>
</tr>
</tbody>
</table>
void setLinkedException(Exception ex)

    Adds a linked Exception.

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public JMSException(String reason, String errorCode)

JMSException

public JMSException(String reason)

null JMSException

Parameters:
reason - a description of the exception
errorCode - a string specifying the vendor-specific error code

public JMSException(String reason)
JMSException

public JMSException(String reason)

Constructs a JMSException with the specified reason and with the error code defaulting to null.

Parameters:
reason - a description of the exception

Method Detail

public String getErrorCode()

    return

getAddress

public String getErrorCode()

    Gets the vendor-specific error code.

Returns:
a string specifying the vendor-specific error code

public Exception getLinkedException()

    return Exception null

getLinkedException

public Exception getLinkedException()
Gets the exception linked to this one.

**Returns:**
the linked Exception, null if none

```java
public void setLinkedException(Exception ex)
```

**setLinkedException**

```java
public void setLinkedException(Exception ex)
```

Adds a linked Exception.

**Parameters:**
- `ex` - the linked Exception
javax.management.j2ee.statistics Interface JMSProducerStats

All Superinterfaces:
   JMSEndpointStats, Stats

public interface JMSProducerStats
extends JMSEndpointStats
Implements: JMSEndpointStats

JMS

Specifies the statistics provided by a JMS message producer

Author:
   Hans Hrasna

Method Summary

<table>
<thead>
<tr>
<th>String</th>
<th>getDestination()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns a string that encapsulates the identity of the message destination</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.JMSEndpointStats
getExpiredMessageCount, getMessageCount, getMessageWaitTime, getPendingMessageCount

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics
public String getDestination()

getDestination

String getDestination()

Returns a string that encapsulates the identity of the message destination.
Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES  NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
javax.jms  Class JMSSecurityException

java.lang.Object  
   | java.lang.Throwable  
   |    | java.lang.Exception  
   |    | javax.jms.JMSException  
   | javax.jms.JMSSecurityException

All Implemented Interfaces:
   Serializable

public class JMSecurityException
extends JMSException

Extends: Throwable > Exception > JMSException

/  

version 26 August 1998

This exception must be thrown when a provider rejects a user name/password submitted by a client. It may also be thrown for any case where a security restriction prevents a method from completing.

Version:  
   26 August 1998

Author:  
   Rahul Sharma

See Also:  
   Serialized Form

---

**Constructor Summary**

JMSecurityException(String reason)
Constructs a JMSSecurityException with the specified reason.

**JMSSecurityException** *(String reason, String errorCode)*

Constructs a JMSSecurityException with the specified reason and error code.

### Method Summary

**Methods inherited from class javax.jms.JMSException**

getErrorCode, getLinkedException, setLinkedException

**Methods inherited from class java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail

public JMSSecurityException(String reason, String errorCode)

JMSSecurityException

reason

errorCode

**JMSSecurityException**

public JMSSecurityException(String reason, String errorCode)

Constructs a JMSSecurityException with the specified reason and
error code.

**Parameters:**
- reason - a description of the exception
- errorCode - a string specifying the vendor-specific error code

```java
public JMSSecurityException(String reason)

JMSSecurityException null

reason
```

**JMSSecurityException**

```java
public JMSSecurityException(String reason)

Constructs a JMSSecurityException with the specified reason. The error code defaults to null.

**Parameters:**
- reason - a description of the exception
```
javax.management.j2ee.statistics Interface JMSSessionStats

All Superinterfaces:
   Stats

public interface JMSSessionStats extends Stats

Implements: Stats
   JMS

Specifies the statistics provided by a JMS session.

Author:
   Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JMSConsumerStats[]</strong> getConsumers()</td>
</tr>
<tr>
<td>Returns an array of JMSConsumerStats that provide statistics about the message consumers associated with the referencing JMS session statistics.</td>
</tr>
<tr>
<td><strong>CountStatistic</strong> getCountStatistic()</td>
</tr>
<tr>
<td><strong>getDurableSubscriptionCount()</strong></td>
</tr>
<tr>
<td>Number of durable subscriptions.</td>
</tr>
<tr>
<td><strong>getExpiredMessageCount()</strong></td>
</tr>
<tr>
<td>Number of expired messages.</td>
</tr>
<tr>
<td><strong>CountStatistic</strong> getCountStatistic()</td>
</tr>
<tr>
<td><strong>getMessageCount()</strong></td>
</tr>
<tr>
<td>Number of messages exchanged.</td>
</tr>
<tr>
<td><strong>TimeStatistic</strong> getTimeStatistic()</td>
</tr>
<tr>
<td><strong>getMessageWaitTime()</strong></td>
</tr>
<tr>
<td>Time spent by a message before being delivered.</td>
</tr>
<tr>
<td><strong>CountStatistic</strong> getCountStatistic()</td>
</tr>
<tr>
<td><strong>getPendingMessageCount()</strong></td>
</tr>
<tr>
<td>Number of pending messages.</td>
</tr>
</tbody>
</table>
### Method Detail

#### public JMSProducerStats[] getProducers()

Returns an array of JMSProducerStats that provide statistics about the message producers associated with the referencing JMS session statistics.

#### public JMSConsumerStats[] getConsumers()

Returns an array of JMSConsumerStats that provide statistics about the message consumers associated with the referencing JMS session statistics.

Methods inherited from interface javax.management.j2ee.statistics.Stats

- getStatistic, getStatisticNames, getStatistics
public CountStatistic getMessageCount()

getMessageCount

CountStatistic getMessageCount()

Number of messages exchanged.

public CountStatistic getPendingMessageCount()

getPendingMessageCount

CountStatistic getPendingMessageCount()

Number of pending messages.

public CountStatistic getExpiredMessageCount()

getExpiredMessageCount

CountStatistic getExpiredMessageCount()

Number of expired messages.

public TimeStatistic getMessageWaitTime()
getMessageWaitTime

TimeStatistic getMessageWaitTime()

Time spent by a message before being delivered.

public CountStatistic getDurableSubscriptionCount()

getDurableSubscriptionCount

CountStatistic getDurableSubscriptionCount()

Number of durable subscriptions.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.management.j2ee.statistics Interface JMSStats

All Superinterfaces: 
   Stats

public interface JMSStats
extends Stats

Implements: Stats

JMS

Specifies the statistics provided by a JMS Resource

Author:
   Hans Hrasna

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMSConnectionStats[] getConnections()</td>
<td>Returns an array of JMSConnectionStats that provide statistics about the connections associated with the referencing JMS resource.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail

public JMSConnectionStats[] getConnections()
JMSConnectionStats  JMS

getConnections

JMSConnectionStats[] getConnections()

Returns an array of JMSConnectionStats that provide statistics about the connections associated with the referencing JMS resource.
javax.persistence Annotation Type JoinColumn

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface JoinColumn

**Implements:** Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@ManyToOne
@JoinColumn(name="ADDR_ID")
public Address getAddress() { return address; }

**since** Java Persistence 1.0

Is used to specify a mapped column for joining an entity association.

Example:

@ManyToOne
@JoinColumn(name="ADDR_ID")
public Address getAddress() { return address; }

**Since:**
Java Persistence 1.0

---

**Optional Element Summary**

<table>
<thead>
<tr>
<th>String</th>
<th>columnDefinition</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Optional) The SQL fragment that is used when generating the DDL for the column.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>boolean insertable</td>
<td>(Optional) Whether the column is included in SQL INSERT statements generated by the persistence provider.</td>
</tr>
<tr>
<td>String name</td>
<td>(Optional) The name of the foreign key column.</td>
</tr>
<tr>
<td>boolean nullable</td>
<td>(Optional) Whether the foreign key column is nullable.</td>
</tr>
<tr>
<td>String referencedColumnName</td>
<td>(Optional) The name of the column referenced by this foreign key column.</td>
</tr>
<tr>
<td>String table</td>
<td>(Optional) The name of the table that contains the column.</td>
</tr>
<tr>
<td>boolean unique</td>
<td>(Optional) Whether the property is a unique key.</td>
</tr>
<tr>
<td>boolean updatable</td>
<td>(Optional) Whether the column is included in SQL UPDATE statements generated by the persistence provider.</td>
</tr>
</tbody>
</table>

abstract public String name()
OneToOne  Many-ToOne
ManyToMany                “ _”“ _”

name

public abstract String name

(Optional) The name of the foreign key column. The table in which it is found depends upon the context. If the join is for a OneToOne or Many-ToOne mapping, the foreign key column is in the table of the source entity. If the join is for a ManyToMany, the foreign key is in a join table. Default (only applies if a single join column is used): The concatenation of the following: the name of the referencing relationship property or field of the referencing entity; " _"; the name of the referenced primary key column. If there is no such referencing
relationship property or field in the entity, the join column name is formed as the concatenation of the following: the name of the entity; "_"; the name of the referenced primary key column.

**Default:**

```

```

**abstract public String referencedColumnName()**

**JoinTable**

**referencedColumnName**

```java
public abstract String referencedColumnName
```

(Optional) The name of the column referenced by this foreign key column. When used with relationship mappings, the referenced column is in the table of the target entity. When used inside a JoinTable annotation, the referenced key column is in the entity table of the owning entity, or inverse entity if the join is part of the inverse join definition. Default (only applies if single join column is being used): The same name as the primary key column of the referenced table.

**Default:**

```

```

**abstract public boolean unique()**

**UniqueConstraint**

**unique**

```java
public abstract boolean unique
```

(Optional) Whether the property is a unique key. This is a shortcut for
the UniqueConstraint annotation at the table level and is useful for when the unique key constraint is only a single field. It is not necessary to explicitly specify this for a join column that corresponds to a primary key that is part of a foreign key.

**Default:**
false

```java
abstract public boolean nullable()
null

nullable
```

```java
public abstract boolean nullable

(Optional) Whether the foreign key column is nullable.

**Default:**
true
```

```java
abstract public boolean insertable()
SQL INSERT

insertable
```

```java
public abstract boolean insertable

(Optional) Whether the column is included in SQL INSERT statements generated by the persistence provider.

**Default:**
true
abstract public boolean updatable()
SQL UPDATE

updatable

public abstract boolean updatable

(Optional) Whether the column is included in SQL UPDATE statements generated by the persistence provider.

Default:
true

abstract public String columnDefinition()
DDL SQL

SQL

columnDefinition

public abstract String columnDefinition

(Optional) The SQL fragment that is used when generating the DDL for the column.

Defaults to the generated SQL for the column.

Default:
"

abstract public String table()
public abstract String table

(Optional) The name of the table that contains the column. If a table is not specified, the column is assumed to be in the primary table of the applicable entity.

Default: 

""
javax.persistence  **Annotation Type JoinColumns**

```java
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface JoinColumns

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
```

```java
public @interface JoinColumns

JoinColumns  

JoinColumn

JoinColumns  JoinColumn  name  referencedColumnName

@ManyToOne
@JoinColumns(
    @JoinColumn(name="ADDR_ID", referencedColumnName="ID"),
    @JoinColumn(name="ADDR_ZIP", referencedColumnName="ZIP")
)
public Address getAddress() { return address; }

since  Java Persistence 1.0  

Defines mapping for the composite foreign keys. This annotation groups `JoinColumn` annotations for the same relationship.

When the `JoinColumns` annotation is used, both the `name` and the `referencedColumnName` elements must be specified in each such `JoinColumn` annotation.

Example:
@ManyToOne
@JoinColumns(
    @JoinColumn(name="ADDR_ID", referencedColumnName="ID"),
    @JoinColumn(name="ADDR_ZIP", referencedColumnName="ZIP")
)
public Address getAddress() { return address; }
```
### Required Element Summary

| JoinColumn[] | value |

### Element Detail

abstract public `JoinColumn[] value()`

**value**

public abstract `JoinColumn[] value`
javax.persistence  Annotation Type JoinTable

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface JoinTable

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

JoinTable

@JoinTable(
    name="CUST_PHONE",
    joinColumns=
        @JoinColumn(name="CUST_ID", referencedColumnName="ID"),
    inverseJoinColumns=
        @JoinColumn(name="PHONE_ID", referencedColumnName="ID")
  )

since Java Persistence 1.0

This annotation is used in the mapping of associations. It is specified on the owning side of a many-to-many association, or in a unidirectional one-to-many association.

If the JoinTable annotation is missing, the default values of the annotation elements apply. The name of the join table is assumed to be the table names of the associated primary tables concatenated together (owning side first) using an underscore.

Example:
@JoinTable(
    name="CUST_PHONE",
    joinColumns=
        @JoinColumn(name="CUST_ID", referencedColumnName="ID"),
    inverseJoinColumns=
        @JoinColumn(name="PHONE_ID", referencedColumnName="ID"),
    ...
```java
inverseJoinColumn=
   @JoinColumn(name="PHONE_ID", referencedColumnName="ID")
)

Since:
   Java Persistence 1.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>catalog</td>
<td>(Optional) The catalog of the table.</td>
</tr>
<tr>
<td>JoinColumn[]</td>
<td>inverseJoinColumns</td>
<td>(Optional) The foreign key columns of the join table which reference the primary table of the entity that does not own the association (i.e. the inverse side of the association).</td>
</tr>
<tr>
<td>JoinColumn[]</td>
<td>joinColumns</td>
<td>(Optional) The foreign key columns of the join table which reference the primary table of the entity owning the association (i.e. the owning side of the association).</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>(Optional) The name of the join table.</td>
</tr>
<tr>
<td>String</td>
<td>schema</td>
<td>(Optional) The schema of the table.</td>
</tr>
<tr>
<td>UniqueConstraint[]</td>
<td>uniqueConstraints</td>
<td>(Optional) Unique constraints that are to be placed on the table.</td>
</tr>
</tbody>
</table>

abstract public String name()
name

public abstract String name

(Optional) The name of the join table.

Defaults to the concatenated names of the two associated primary entity tables, separated by an underscore.

Default:
""

-------------

abstract public String catalog()

catalog

public abstract String catalog

(Optional) The catalog of the table.

Defaults to the default catalog.

Default:
""

-------------

abstract public String schema()
schema

public abstract String schema

(Optional) The schema of the table.

Defaults to the default schema for user.

Default: ""

abstract public JoinColumn[] joinColumns()

JoinColumn

joinColumns

public abstract JoinColumn[] joinColumns

(Optional) The foreign key columns of the join table which reference the primary table of the entity owning the association (i.e. the owning side of the association).

Uses the same defaults as for JoinColumn.

Default:

{}

abstract public JoinColumn[] inverseJoinColumns()
JoinColumn

inverseJoinColumns

public abstract JoinColumn[] inverseJoinColumns

(Optional) The foreign key columns of the join table which reference the primary table of the entity that does not own the association (i.e. the inverse side of the association).

Uses the same defaults as for JoinColumn.

Default:

{}  

abstract public UniqueConstraint[] uniqueConstraints()

uniqueConstraints

public abstract UniqueConstraint[] uniqueConstraints

(Optional) Unique constraints that are to be placed on the table. These are only used if table generation is in effect.

Defaults to no additional constraints.

Default:

{}
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
public interface JspApplicationContext

Stores application-scoped information relevant to JSP containers.

The JSP container must create a single instance of JspApplicationContext for each ServletContext instance.

An instance of JspApplicationContext is obtained by invoking the static JspFactory.getJspApplicationContext(javax.servlet.ServletContext) method, passing the ServletContext of the corresponding web application.

The JspApplicationContext provides the following services to JSP applications:

- Allows registration of ELResolvers, which are used to resolve variables in EL expressions contained in JSP pages and tag files.
- Provides an instance of ExpressionFactory for those applications or frameworks that need to perform programmatic evaluation of EL expressions instead of allowing the JSP container to do it for them.
- Allows the attachment of ELContextListener instances for notification whenever a new ELContext is created. This is necessary when an application wishes to make custom context objects available to their pluggable ELResolvers.

Since: JSP 2.1

See Also: ServletContext, JspFactory, ELResolver, ExpressionFactory, ELContextListener

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>addELContextListener(ELContextListener listener)</td>
</tr>
</tbody>
</table>
void addELResolver(ELResolver resolver)

Adds an ELResolver to affect the way EL variables and properties are resolved for EL expressions appearing in JSP pages and tag files.

For example, in the EL expression `${employee.lastName}`, an ELResolver determines what object "employee" references and how to find its "lastName" property.

When evaluating an expression, the JSP container will consult a set of standard resolvers as well as any resolvers registered via this method. The set of resolvers are consulted in the following order:

- ImplicitObjectELResolver
- ListELResolver
- MapELResolver
- ArrayELResolver
- BeanELResolver
- **ScopedAttributeELResolver**

  It is illegal to register an ELResolver after the application has received any request from the client. If an attempt is made to register an ELResolver after that time, an IllegalStateException is thrown.

  This restriction is in place to allow the JSP container to optimize for the common case where no additional ELResolvers are in the chain, aside from the standard ones. It is permissible to add ELResolvers before or after initialization to a CompositeELResolver that is already in the chain.

  It is not possible to remove an ELResolver registered with this method, once it has been registered.

  **Parameters:**
  - resolver - The new ELResolver

  **Throws:**
  - [IllegalStateException](#) - if an attempt is made to call this method after all ServletContextListenerS have had their contextInitialized methods invoked.

---

**getExpressionFactory**

`ExpressionFactory getExpressionFactory()`

Returns a factory used to create ValueExpressions and MethodExpressionS so that EL expressions can be parsed and evaluated.

**Returns:**
A concrete implementation of the an ExpressionFactory.

---

**addELContextListener**

`void addELContextListener(ELContextListener listener)`

Registers a ELContextListeners so that context objects can be
added whenever a new ELContext is created.

At a minimum, the ELContext objects created will contain a reference to the JspContext for this request, which is added by the JSP container. This is sufficient for all the default ELResolvers listed in addELResolver(javax.el.ELResolver). Note that JspContext.class is used as the key to ELContext.putContext() for the JspContext object reference.

This method is generally used by frameworks and applications that register their own ELResolver that needs context other than JspContext. The listener will typically add the necessary context to the ELContext provided in the event object. Registering a listener that adds context allows the ELResolvers in the stack to access the context they need when they do a resolution.

**Parameters:**

- listener - The listener to be notified when a new ELContext is created.
javax.servlet.jsp  **Class JspContext**

**java.lang.Object**
  |_ javax.servlet.jsp.JspContext

**Direct Known Subclasses:**
  PageContext

```java
public abstract class JspContext
  extends Object

Extended by: PageContext
```

JspContext  PageContext  servlet / Servlet

JspContext /
  - API
  - JspWriter
  -

JspWriter  pushBody()  popBody()

**JSP**

Servlet JSP
  setAttribute()  getAttribute()  findAttribute()  removeAttribute()  getAt
  getAttributeNamesInScope()

getOut()

getExpressionEvaluator()  getVariableResolver()

since  2.0
JspContext serves as the base class for the PageContext class and abstracts all information that is not specific to servlets. This allows for Simple Tag Extensions to be used outside of the context of a request/response Servlet.

The JspContext provides a number of facilities to the page/component author and page implementor, including:

- a single API to manage the various scoped namespaces
- a mechanism to obtain the JspWriter for output
- a mechanism to expose page directive attributes to the scripting environment

Methods Intended for Container Generated Code

The following methods enable the management of nested JspWriter streams to implement Tag Extensions: pushBody() and popBody()

Methods Intended for JSP authors

Some methods provide uniform access to the diverse objects representing scopes. The implementation must use the underlying machinery corresponding to that scope, so information can be passed back and forth between the underlying environment (e.g. Servlets) and JSP pages. The methods are: setAttribute(), getAttribute(), findAttribute(), removeAttribute(), getAttributesScope() and getAttributeNamesInScope().

The following methods provide convenient access to implicit objects: getOut()

The following methods provide programmatic access to the Expression Language evaluator: getExpressionEvaluator(), getVariableResolver()

Since:

JSP 2.0
## Constructor Summary

**JspContext()**

Sole constructor.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>findAttribute(String name)</strong></td>
<td>Searches for the named attribute in page, request, session (if valid), and application scope(s) in order and returns the value associated or null.</td>
</tr>
<tr>
<td><strong>getAttribute(String name)</strong></td>
<td>Returns the object associated with the name in the page scope or null if not found.</td>
</tr>
<tr>
<td><strong>getAttribute(String name, int scope)</strong></td>
<td>Return the object associated with the name in the specified scope or null if not found.</td>
</tr>
<tr>
<td><strong>getAttributeNamesInScope(int scope)</strong></td>
<td>Enumerate all the attributes in a given scope.</td>
</tr>
<tr>
<td><strong>getAttributesScope(String name)</strong></td>
<td>Get the scope where a given attribute is defined.</td>
</tr>
<tr>
<td><strong>getELContext()</strong></td>
<td>Returns the ELContext associated with this JspContext.</td>
</tr>
<tr>
<td><strong>getExpressionEvaluator()</strong></td>
<td>Deprecated. As of JSP 2.1, replaced by <code>JspApplicationContext.getExpressionFactory()</code></td>
</tr>
<tr>
<td><strong>getOut()</strong></td>
<td>The current value of the out object (a JspWriter).</td>
</tr>
<tr>
<td><strong>getVariableResolver()</strong></td>
<td>Deprecated. As of JSP 2.1, replaced by <code>ELContext.getELResolver()</code>, which can be obtained by <code>jspContext.getELContext().getELResolver()</code></td>
</tr>
<tr>
<td><strong>popBody()</strong></td>
<td>Return the previous JspWriter &quot;out&quot; saved by the matching pushBody(), and update the value of the &quot;out&quot;</td>
</tr>
</tbody>
</table>
attribute in the page scope attribute namespace of the JspContext.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pushBody(Writer writer)</code></td>
<td>Return a new JspWriter object that sends output to the provided Writer.</td>
</tr>
<tr>
<td><code>removeAttribute(String name)</code></td>
<td>Remove the object reference associated with the given name from all scopes.</td>
</tr>
<tr>
<td><code>removeAttribute(String name, int scope)</code></td>
<td>Remove the object reference associated with the specified name in the given scope.</td>
</tr>
<tr>
<td><code>setAttribute(String name, Object value)</code></td>
<td>Register the name and value specified with page scope semantics.</td>
</tr>
<tr>
<td><code>setAttribute(String name, Object value, int scope)</code></td>
<td>Register the name and value specified with appropriate scope semantics.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
abstract public void setAttribute(String name, Object value)

null  removeAttribute

name
value  name  null
Throws  NullPointerException:  null

setAttribute

public abstract void setAttribute(String name, Object value)

Register the name and value specified with page scope semantics. If the value passed in is null, this has the same effect as calling removeAttribute( name, PageContext.PAGE_SCOPE ).

Parameters:
     name - the name of the attribute to set
     value - the value to associate with the name, or null if the attribute is to be removed from the page scope.

Throws:
     NullPointerException - if the name is null

abstract public void setAttribute(String name, Object value, int scope)

null  removeAttribute

name
value  name  null
scope  /
Throws  NullPointerException:  null
Throws  IllegalArgumentException:  scope
Throws  IllegalStateException:  scope
**Throws** PageContext.SESSION_SCOPE

### setAttribute

```java
public abstract void setAttribute(String name, Object value, int scope)
```

Register the name and value specified with appropriate scope semantics. If the value passed in is `null`, this has the same effect as calling `removeAttribute(name, scope)`.

**Parameters:**
- `name` - the name of the attribute to set
- `value` - the object to associate with the name, or `null` if the attribute is to be removed from the specified scope.
- `scope` - the scope with which to associate the name/object

**Throws:**
- `NullPointerException` - if the name is `null`
- `IllegalArgumentException` - if the scope is invalid
- `IllegalStateException` - if the scope is `PageContext.SESSION_SCOPE` but the page that was requested does not participate in a session or the session has been invalidated.

---

### abstract public Object getAttribute(String name)

```java
null
```

```java
    name
    return null
    Throws NullPointerException: null
```

### getAttribute

```java
public abstract Object getAttribute(String name)
```
Returns the object associated with the name in the page scope or null if not found.

**Parameters:**
- name - the name of the attribute to get

**Returns:**
the object associated with the name in the page scope or null if not found.

**Throws:**
- NullPointerException - if the name is null

abstract public Object getAttribute(String name, int scope)
null

name
scope /
return null

Throws
- NullPointerException: null
- IllegalArgumentException: scope
- IllegalStateException: scope
- PageContext.SESSION_SCOPE

**getAttribute**

public abstract Object getAttribute(String name, int scope)

Return the object associated with the name in the specified scope or null if not found.

**Parameters:**
- name - the name of the attribute to set
- scope - the scope with which to associate the name/object

**Returns:**
the object associated with the name in the specified scope or null if not found.

**Throws:**
NullPointerException - if the name is null
IllegalArgumentException - if the scope is invalid
IllegalStateException - if the scope is PageContext.SESSION_SCOPE but the page that was requested does not participate in a session or the session has been invalidated.

abstract public Object findAttribute(String name)
null

name
return null

Throws
NullPointerException: null

findAttribute

public abstract Object findAttribute(String name)

Searches for the named attribute in page, request, session (if valid), and application scope(s) in order and returns the value associated or null.

Parameters:
name - the name of the attribute to search for

Returns:
the value associated or null

Throws:
NullPointerException - if the name is null

abstract public void removeAttribute(String name)

name

Throws
NullPointerException: null
removeAttribute

public abstract void removeAttribute(String name)

Remove the object reference associated with the given name from all scopes. Does nothing if there is no such object.

Parameters:
    name - The name of the object to remove.

Throws:
    NullPointerException - if the name is null

abstract public void removeAttribute(String name, int scope)

    name
    scope

Throws
    IllegalArgumentException: scope
    IllegalStateException: scope
    PageContext.SESSION_SCOPE
    NullPointerException: null

removeAttribute

public abstract void removeAttribute(String name, int scope)

Remove the object reference associated with the specified name in the given scope. Does nothing if there is no such object.

Parameters:
    name - The name of the object to remove.
    scope - The scope where to look.

Throws:
    IllegalArgumentException - if the scope is invalid
    IllegalStateException - if the scope is
PageContext.SESSION_SCOPE but the page that was requested does not participate in a session or the session has been invalidated.

**NullPointerException** - if the name is null

---

**abstract public int getAttributesScope(String name)**

```
name
return 0
Throws NullPointerException: null
```

**getAttributesScope**

**public abstract int getAttributesScope(String name)**

Get the scope where a given attribute is defined.

**Parameters:**
- `name` - the name of the attribute to return the scope for

**Returns:**
- the scope of the object associated with the name specified or 0

**Throws:**
- `NullPointerException` - if the name is null

---

**abstract public java.util Enumeration<E> getAttributeNamesInScope(int scope)**

```
scope
return (java.lang.String)
Throws IllegalArgumentException: scope
Throws IllegalStateException: scope
Throws PageContext.SESSION_SCOPE
```

---
get AttributeNamesInScope

public abstract Enumeration<String> getAttributeNamesInScope(int scope)

Enumerate all the attributes in a given scope.

Parameters:
  scope - the scope to enumerate all the attributes for

Returns:
  an enumeration of names (java.lang.String) of all the attributes
  the specified scope

Throws:
  IllegalArgumentException - if the scope is invalid
  IllegalStateException - if the scope is
  PageContext.SESSION_SCOPE but the page that was
  requested does not participate in a session or the session has
  been invalidated.

abstract public JspWriter getOut()

out JspWriter

return JspWriter

getOut

public abstract JspWriter getOut()

The current value of the out object (a JspWriter).

Returns:
  the current JspWriter stream being used for client response

abstract public ExpressionEvaluator getExpressionEvaluator()

gExpressionEvaluator JSP_EL
ExpressionEvaluator
getExpressionEvaluator

public abstract ExpressionEvaluator getExpressionEvaluator()

**Deprecated. As of JSP 2.1, replaced by**

`JspApplicationContext.getExpressionFactory()`

Provides programmatic access to the ExpressionEvaluator. The JSP Container must return a valid instance of an ExpressionEvaluator that can parse EL expressions.

**Returns:**
A valid instance of an ExpressionEvaluator.

**Since:**
JSP 2.0

---

getVariableResolver

public abstract VariableResolver getVariableResolver()

**Deprecated. As of JSP 2.1, replaced by** `ELContext.getELResolver()`, which can be obtained by `jspContext.getELContext().getELResolver()`.

Returns an instance of a VariableResolver that provides access to the implicit objects specified in the JSP specification using this JspContext as the context object.
Returns:
A valid instance of a VariableResolver.

Since:
JSP 2.0

---

class JspContext

getELContext

public abstract ELContext getELContext()

Returns the ELContext associated with this JspContext.

The ELContext is created lazily and is reused if it already exists. There is a new ELContext for each JspContext.

The ELContext must contain the ELResolver described in the JSP specification (and in the javadocs for JspApplicationContext.addELResolver(javax.el.ELResolver)).

Returns:
The ELContext associated with this JspContext.

Since:
JSP 2.1

---

public JspWriter pushBody(java.io.Writer writer)

Writer  JspWriter "out"  JspWriter  "out"  JspContext  "out"

JspWriter

- clear() IOException
- clearBuffer()
- getBufferSize() 0
- getRemaining() 0
public JspWriter pushBody(Writer writer)

Return a new JspWriter object that sends output to the provided Writer. Saves the current "out" JspWriter, and updates the value of the "out" attribute in the page scope attribute namespace of the JspContext.

The returned JspWriter must implement all methods and behave as though it were unbuffered. More specifically:

- clear() must throw an IOException
- clearBuffer() does nothing
- getBufferSize() always returns 0
- getRemaining() always returns 0

Parameters:
- writer - The Writer for the returned JspWriter to send output to.

Returns:
a new JspWriter that writes to the given Writer.

Since:
JSP 2.0

public JspWriter popBody()
public JspWriter popBody()

Return the previous JspWriter "out" saved by the matching pushBody(), and update the value of the "out" attribute in the page scope attribute namespace of the JspContext.

Returns:
the saved JspWriter.
public abstract class JspEngineInfo
extends Object

JspEngineInfo JSP

The JspEngineInfo is an abstract class that provides information on the current JSP engine.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>JspEngineInfo()</td>
</tr>
<tr>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract String getSpecificationVersion()</td>
</tr>
<tr>
<td>Return the version number of the JSP specification that is supported by this JSP engine.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
- wait

Constructor Detail
public JspEngineInfo()

JspEngineInfo

public JspEngineInfo()

    Sole constructor. (For invocation by subclass constructors, typically implicit.)

Method Detail

abstract public String getSpecificationVersion()

    "." "2.0" "1.2.3.4.5.6.7"

    return null

getSpecificationVersion

public abstract String getSpecificationVersion()

    Return the version number of the JSP specification that is supported by this JSP engine.

    Specification version numbers that consists of positive decimal integers separated by periods ".", for example, "2.0" or "1.2.3.4.5.6.7". This allows an extensible number to be used to represent major, minor, micro, etc versions. The version number must begin with a number.

    Returns:
    the specification version, null is returned if it is not known
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet.jsp  **Class JspException**

```java
java.lang.Object  
   \_ java.lang.Throwble  
      \_ java.lang.Exception  
         \_ javax.servlet.jsp.JspException
```

**All Implemented Interfaces:**

  Serializable

**Direct Known Subclasses:**

  JspTagException, SkipPageException

---

```java
public class JspException
extends Exception
```

**Extends:** Throwable > Exception

**Extended by:** JspTagException, SkipPageException

JSP  JspException  errorpage

A generic exception known to the JSP engine; uncaught JspExceptions will result in an invocation of the errorpage machinery.

**See Also:**

  Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>JspException()</code></td>
<td>Construct a JspException.</td>
</tr>
<tr>
<td><code>JspException(String msg)</code></td>
<td>Constructs a new JSP exception with the specified message.</td>
</tr>
<tr>
<td><code>JspException(String message, Throwable cause)</code></td>
<td>Constructs a new JspException with the specified detail message</td>
</tr>
</tbody>
</table>
and cause.

**JspException** *(Throwable cause)*

Constructs a new JspException with the specified cause.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getRootCause()</code></td>
<td>Deprecated. As of JSP 2.1, replaced by <code>Throwable.getCause()</code></td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getCause`, `getLocaleMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

**public JspException()**

*JspException*

**public JspException()**

Construct a JspException.

**public JspException(String msg)**

*JSP /

  msg  String*
JspException

public JspException(String msg)

Constructs a new JSP exception with the specified message. The message can be written to the server log and/or displayed for the user.

Parameters:
msg - a String specifying the text of the exception message

public JspException(String message, Throwable rootCause)

JSP   JSP “” JSP

message  String
rootCause  servlet  Throwable  servlet

JspException

public JspException(String message, Throwable cause)

Constructs a new JspException with the specified detail message and cause. The cause is saved for later retrieval by the Throwable.getCause() and getRootCause() methods.

See Also:
Exception.Exception(String, Throwable)

public JspException(Throwable rootCause)

JSP   JSP “” JSP

Throwable  getLocalizedMessage  JspException
public JspException(Throwable cause)

Constructs a new JspException with the specified cause. The cause is saved for later retrieval by the Throwable.getCause() andgetRootCause() methods.

See Also:
Exception.Exception(Throwable)

public ThrowablegetRootCause()

Returns the exception that caused this JspException.

Returns:
the Throwable that caused this JspException

Deprecated. As of JSP 2.1, replaced by Throwable.getCause()
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The JspFactory is an abstract class that defines a number of factory methods available to a JSP page at runtime for the purposes of creating instances of various interfaces and classes used to support the JSP implementation.

A conformant JSP Engine implementation will, during its initialization instantiate an implementation dependent subclass of this class, and make it globally available for use by JSP implementation classes by registering the instance created with this class via the static setDefaultFactory() method.

The only implementation-dependent classes that can be created from the factory are: PageContext, JspEngineInfo, and JspApplicationContext.

With the exception of JspApplicationContext, JspFactory objects should not be used by JSP application developers.
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>JspFactory()</td>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static JspFactory</td>
<td>getDefaultFactory()</td>
<td>Returns the default factory for this implementation.</td>
</tr>
<tr>
<td>abstract JspEngineInfo</td>
<td>getEngineInfo()</td>
<td>called to get implementation-specific information on the current JSP engine.</td>
</tr>
<tr>
<td>abstract JspApplicationContext</td>
<td>getJspApplicationContext(ServletContext context)</td>
<td>Obtains the JspApplicationContext instance associated with the web application for the given ServletContext.</td>
</tr>
<tr>
<td>abstract PageContext</td>
<td>getPageContext(Servlet servlet, ServletRequest request, ServletResponse response, String errorPageURL, boolean needsSession, int buffer, boolean autoflush)</td>
<td>obtains an instance of an implementation dependent javax.servlet.jsp.PageContext abstract class for the calling Servlet and currently pending request and response.</td>
</tr>
<tr>
<td>abstract void</td>
<td>releasePageContext(PageContext pc)</td>
<td>called to release a previously allocated PageContext object.</td>
</tr>
<tr>
<td>static void</td>
<td>setDefaultFactory(JspFactory deflt)</td>
<td>set the default factory for this implementation.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detail

public JspFactory()

JspFactory

public JspFactory()

   Sole constructor. (For invocation by subclass constructors, typically implicit.)

Method Detail

public static void setDefaultFactory(JspFactory deflt)

   JSP

       deflt

setDefaultFactory

public static void setDefaultFactory(JspFactory deflt)

   set the default factory for this implementation. It is illegal for any principal other than the JSP Engine runtime to call this method.

   Parameters:
       deflt - The default factory implementation

public static JspFactory getDefaultFactory()}

   return
**getDefualtFactory**

public static JspFactory getDefaultFactory()

Returns the default factory for this implementation.

**Returns:**

the default factory for this implementation

---

**abstract public** PageContext getPageContext(Servlet servlet, ServletRequest request, ServletResponse response, String errorPageURL, boolean needsSession, int buffer, boolean autoflush)

Servlet  javservlet.jsp.PageContext

PageContext  JSP  _jspService()

PageContext.initialize()  PageContext

PageContext  releasePageContext()

```java
servlet
request
response
errorPageURL
needsSession
buffer
autoflush
return
```

JSP  URL null

JSP  true

PageContext.NO_BUFFER

PageContext.DEFAULT_BUFFER

IOException

See also  javaservlet.jsp.PageContext
getPageContext

```java
public abstract PageContext getPageContext(Servlet servlet,
                                      ServletRequest request,
                                      ServletResponse response,
                                      String errorPageURL,
                                      boolean needsSession,
                                      int buffer,
                                      boolean autoflush)
```

obtains an instance of an implementation dependent
javax.servlet.jsp.PageContext abstract class for the calling Servlet
and currently pending request and response.

This method is typically called early in the processing of the
_jspService() method of a JSP implementation class in order to
obtain a PageContext object for the request being processed.

Invoking this method shall result in the PageContext.initialize() method being invoked. The PageContext returned is properly initialized.

All PageContext objects obtained via this method shall be released
by invoking releasePageContext().

**Parameters:**
- `servlet` - the requesting servlet
- `request` - the current request pending on the servlet
- `response` - the current response pending on the servlet
- `errorPageURL` - the URL of the error page for the requesting JSP, or null
- `needsSession` - true if the JSP participates in a session
- `buffer` - size of buffer in bytes, PageContext.NO_BUFFER if no buffer, PageContext.DEFAULT_BUFFER if implementation default.
- `autoflush` - should the buffer autoflush to the output stream on buffer overflow, or throw an IOException?

**Returns:**
the page context

**See Also:**
PageContext
abstract public void releasePageContext(PageContext pc)

PageContext  PageContext.release()  JSP
  _jspService()

  pc             getPageContext()  PageContext

releasePageContext

public abstract void releasePageContext(PageContext pc)

called to release a previously allocated PageContext object. Results in PageContext.release() being invoked. This method should be invoked prior to returning from the _jspService() method of a JSP implementation class.

Parameters:
   pc - A PageContext previously obtained by getPageContext()

abstract public JspEngineInfo getEngineInfo()

JSP

  return       JSP  JspEngineInfo

g EngelInfo

public abstract JspEngineInfo getEngineInfo()

called to get implementation-specific information on the current JSP engine.

Returns:  
a JspEngineInfo object describing the current JSP engine
getJspApplicationContext

public abstract JspApplicationContext getJspApplicationContext(ServletContext context)

Obtains the JspApplicationContext instance associated with the web application for the given ServletContext.

Parameters:
context - The ServletContext for the web application the desired JspApplicationContext is associated with.

Returns:
The JspApplicationContext associated with the web application.

Since:
2.1
Encapsulates a portion of JSP code in an object that can be invoked as many times as needed. JSP Fragments are defined using JSP syntax as the body of a tag for an invocation to a SimpleTag handler, or as the body of a `<jsp:attribute>` standard action specifying the value of an attribute that is declared as a fragment, or to be of type JspFragment in the TLD.

The definition of the JSP fragment must only contain template text and JSP action elements. In other words, it must not contain scriptlets or scriptlet expressions. At translation time, the container generates an implementation of the JspFragment abstract class capable of executing the defined fragment.

A tag handler can invoke the fragment zero or more times, or pass it along to other tags, before returning. To communicate values to/from a
JSP fragment, tag handlers store/retrieve values in the JspContext associated with the fragment.

Note that tag library developers and page authors should not generate JspFragment implementations manually.

*Implementation Note:* It is not necessary to generate a separate class for each fragment. One possible implementation is to generate a single helper class for each page that implements JspFragment. Upon construction, a discriminator can be passed to select which fragment that instance will execute.

**Since:**
- JSP 2.0

### Constructor Summary

<table>
<thead>
<tr>
<th>JspFragment()</th>
</tr>
</thead>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>JspContext</th>
<th>abstract getJspContext() Returns the JspContext that is bound to this JspFragment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>invoke(Writer out) Executes the fragment and directs all output to the given Writer, or the JspWriter returned by the getOut() method of the JspContext associated with the fragment if out is null.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail
public JspFragment()

JspFragment

public JspFragment()

### Method Detail

abstract public void invoke(java.io.Writer out) throws JspException, java.io.IOException

Writer out  null JspContext  getOut()

JspWriter

```
out  Writer JspContext.getOut()  null
```

**Throws**

- JspException:
- SkipPageException: Tag.SKIP_PAGE
- SkipPageException
- java.io.IOException:

### invoke

public abstract void invoke(Writer out) throws JspException, IOException

Executes the fragment and directs all output to the given Writer, or
the JspWriter returned by the getOut() method of the JspContext
associated with the fragment if out is null.

**Parameters:**

- out - The Writer to output the fragment to, or null if output should
  be sent to JspContext.getOut().

**Throws:**

- JspException - Thrown if an error occurred while invoking this
  fragment.
- SkipPageException - Thrown if the page that (either directly or
indirectly) invoked the tag handler that invoked this fragment is to cease evaluation. The container must throw this exception if a Classic Tag Handler returned Tag.SKIP_PAGE or if a Simple Tag Handler threw SkipPageException.

IOException - If there was an error writing to the stream.

abstract public JspContext getJspContext()
JspFragment JspContext

getJspContext

public abstract JspContext getJspContext()

Returns the JspContext that is bound to this JspFragment.

Returns:
The JspContext used by this fragment at invocation time.
public interface JspIdConsumer

This interface indicates to the container that a tag handler wishes to be provided with a compiler generated ID.

The container sets the jspId attribute of the tag handler with an identification string, as part of tag property initialization. Each tag in a JSP page has a unique jspId, and a given tag in a JSP page always has the same jspId, even for multiple requests to the page.

Tag handler instances that implement JspIdConsumer cannot be reused.

Even though the jspId attribute is similar in concept to the jsp:id attribute of an XML view (see Section JSP.10.1.13 of the spec), they are not related. The jsp:id attribute is available only at translation time, and the jspId attribute is available only at request time.

The JSP container must provide a value for jspId that conforms to the following rules:

- It must start with a letter (as defined by the Character.isLetter() method) or underscore ('_').
- Subsequent characters may be letters (as defined by the Character.isLetter() method), digits (as defined by the Character.isDigit() method), dashes ('-'), or underscores ('_').

Note that the rules exclude colons ':' in a jspId, and that they are the same rules used for a component ID in JavaServer Faces.

Since:
   JSP 2.1
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setJspId</td>
<td>void setJspId(String id)</td>
<td>Called by the container generated code to set a value for the jspId attribute.</td>
</tr>
</tbody>
</table>

Method Detail

setJspId

void setJspId(String id)

Called by the container generated code to set a value for the jspId attribute. An unique identification string, relative to this page, is generated at translation time.
javax.servlet.jsp  Interface JspPage

All Superinterfaces:
   Servlet

All Known Subinterfaces:
   HttpJspPage

public interface JspPage
extends Servlet

Implements: Servlet
Implemented by: HttpJspPage

JspPage  JSP  HTTP  HttpJspPage

jspInit()  jspDestroy()  _jspService()  Java

Servlet

jspInit()  jspDestroy()  JSP  _jspService()  JSP  JSP

_jjspService()

_jjspService()  JSP  JSP  JSP

_jjspService()  service()  JSP  JSP_Engine

JSP

public void _jspService(ServletRequestSubtype request,
ServletResponseSubtype response)
throws ServletException, IOException;
The JspPage interface describes the generic interaction that a JSP Page Implementation class must satisfy; pages that use the HTTP protocol are described by the HttpJspPage interface.

**Two plus One Methods**

The interface defines a protocol with 3 methods; only two of them: jspInit() and jspDestroy() are part of this interface as the signature of the third method: _jspService() depends on the specific protocol used and cannot be expressed in a generic way in Java.

A class implementing this interface is responsible for invoking the above methods at the appropriate time based on the corresponding Servlet-based method invocations.

The jspInit() and jspDestroy() methods can be defined by a JSP author, but the _jspService() method is defined automatically by the JSP processor based on the contents of the JSP page.

_**jspService()**_

The _jspService() method corresponds to the body of the JSP page. This method is defined automatically by the JSP container and should never be defined by the JSP page author.

If a superclass is specified using the extends attribute, that superclass may choose to perform some actions in its service() method before or after calling the _jspService() method. See using the extends attribute in the JSP_Engine chapter of the JSP specification.

The specific signature depends on the protocol supported by the JSP page.

    public void _jspService(ServletRequestSubtype request, ServletResponseSubtype response) throws ServletException, IOException;
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void jspDestroy()</td>
<td>The jspDestroy() method is invoked when the JSP page is about to be destroyed.</td>
</tr>
<tr>
<td>void jspInit()</td>
<td>The jspInit() method is invoked when the JSP page is initialized.</td>
</tr>
</tbody>
</table>

## Methods inherited from interface javax.servlet.Servlet

- destroy
- getServletConfig
- getServletInfo
- init
- service

## Method Detail

```java
public void jspInit()

dependencies:
jspInit() JSP
getServletConfig() JSP
jspInit JSP
```

### jspInit

```java
void jspInit()
```

The jspInit() method is invoked when the JSP page is initialized. It is the responsibility of the JSP implementation (and of the class mentioned by the extends attribute, if present) that at this point invocations to the getServletConfig() method will return the desired value. A JSP page can override this method by including a definition for it in a declaration element. A JSP page should redefine the init() method from Servlet.

```java
public void jspDestroy()

dependencies:
jspDestroy() JSP
```

### jspDestroy

```java
void jspDestroy()
```
jspDestroy

void jspDestroy()

The jspDestroy() method is invoked when the JSP page is about to be destroyed. A JSP page can override this method by including a definition for it in a declaration element. A JSP page should redefine the destroy() method from Servlet.
javax.servlet.jsp.tagext  Interface JspTag

All Known Subinterfaces:
    BodyTag, IterationTag, SimpleTag, Tag

All Known Implementing Classes:
    AttributeTag, BodyTagSupport, ConverterELTag, ConverterTag,
    FacetTag, SimpleTagSupport, TagAdapter, TagSupport,
    UIComponentBodyTag, UIComponentClassicTagBase,
    UIComponentELTag, UIComponentTag, UIComponentTagBase,
    ValidatorELTag, ValidatorTag

public interface JspTag

Implemented by: SimpleTag, Tag, UIComponentTagBase

Tag  SimpleTag

    since  2.0

Serves as a base class for Tag and SimpleTag. This is mostly for
organizational and type-safety purposes.

Since:
    JSP 2.0
PS :
javax.servlet.jsp  **Class JspTagException**

**java.lang.Object**
  └── **java.lang.Throwable**
      └── **java.lang.Exception**
          └── **javax.servlet.jsp.JspException**
              └── **javax.servlet.jsp.JspTagException**

**All Implemented Interfaces:**
  **Serializable**

```java
public class JspTagException
extends JspException

**Extends:** Throwable > Exception > JspException

JSP

Exception to be used by a Tag Handler to indicate some unrecoverable error. This error is to be caught by the top level of the JSP page and will result in an error page.

**See Also:**
  Serialized Form
```

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JspTagException</strong> ()</td>
</tr>
<tr>
<td>Constructs a new JspTagException with no message.</td>
</tr>
<tr>
<td><strong>JspTagException</strong> (String msg)</td>
</tr>
<tr>
<td>Constructs a new JspTagException with the specified message.</td>
</tr>
<tr>
<td><strong>JspTagException</strong> (String message, Throwable rootCause)</td>
</tr>
<tr>
<td>Constructs a new JspTagException when the JSP Tag needs to throw an exception and include a message about the &quot;root cause&quot; exception that interfered with its normal operation, including a</td>
</tr>
</tbody>
</table>
description message.

```
JspTagException(Throwable rootCause)

Constructs a new JSP Tag exception when the JSP Tag needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation.
```

**Method Summary**

**Methods inherited from class javax.servlet.jsp.JspException**

```java
getRootCause
```

**Methods inherited from class java.lang.Throwable**

```java
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString
```

**Methods inherited from class java.lang.Object**

```java
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
```

**Constructor Detail**

```java
public JspTagException(String msg)
```

```
JspTagException/

msg
String
```

**JspTagException**

```java
public JspTagException(String msg)
```

```
Constructs a new JspTagException with the specified message. The message can be written to the server log and/or displayed for the user.
```
Parameters:
- **msg** - a String specifying the text of the exception message

```java
public JspTagException()
JspTagException

JspTagException

public JspTagException()

Constructs a new JspTagException with no message.

public JspTagException(String message, Throwable rootCause)

Constructs a new JspTagException when the JSP Tag needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation, including a description message.

Parameters:
- **message** - a String containing the text of the exception message
- **rootCause** - the Throwable exception that interfered with the JSP Tag's normal operation, making this JSP Tag exception necessary
public JspTagException(Throwable rootCause)

Constructs a new JSP Tag exception when the JSP Tag needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation. The exception's message is based on the localized message of the underlying exception.

This method calls the `getLocalizedMessage` method on the `Throwable` exception to get a localized exception message. When subclassing `JspTagException`, this method can be overridden to create an exception message designed for a specific locale.

**Parameters:**
- `rootCause` - the `Throwable` exception that interfered with the JSP Tag's normal operation, making the JSP Tag exception necessary

**Since:**
- JSP 2.0
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet.jsp Class JspWriter

java.lang.Object
   java.io.Writer
      javax.servlet.jsp.JspWriter

All Implemented Interfaces:
   Closeable, Flushable, Appendable

Direct Known Subclasses:
   BodyContent

public abstract class JspWriter
   extends Writer

Extends: java.io.Writer
Extended by: BodyContent

JSP JspWriter out PageContext

   java.io.BufferedWriter java.io.PrintWriter print
   java.io.IOException PrintWriter

   JspWriter ServletResponse PrintWriter JspWriter
   PrintWriter getWriter() PrintWriter PrintWriter
   setContentType() JSP

   •
   •

   JSP autoFlush true JSP autoFlush false JSP autoFlush true
The actions and template data in a JSP page is written using the JspWriter object that is referenced by the implicit variable out which is initialized automatically using methods in the PageContext object.

This abstract class emulates some of the functionality found in the java.io.BufferedWriter and java.io.PrintWriter classes, however it differs in that it throws java.io.IOException from the print methods while PrintWriter does not.

Buffering

The initial JspWriter object is associated with the PrintWriter object of the ServletResponse in a way that depends on whether the page is or is not buffered. If the page is not buffered, output written to this JspWriter object will be written through to the PrintWriter directly, which will be created if necessary by invoking the getWriter() method on the response object. But if the page is buffered, the PrintWriter object will not be created until the buffer is flushed and operations like setContentType() are legal. Since this flexibility simplifies programming substantially, buffering is the default for JSP pages.

Buffering raises the issue of what to do when the buffer is exceeded. Two approaches can be taken:

- Exceeding the buffer is not a fatal error; when the buffer is exceeded, just flush the output.
- Exceeding the buffer is a fatal error; when the buffer is exceeded, raise an exception.

Both approaches are valid, and thus both are supported in the JSP technology. The behavior of a page is controlled by the autoFlush attribute, which defaults to true. In general, JSP pages that need to be
sure that correct and complete data has been sent to their client may want to set autoFlush to false, with a typical case being that where the client is an application itself. On the other hand, JSP pages that send data that is meaningful even when partially constructed may want to set autoFlush to true; such as when the data is sent for immediate display through a browser. Each application will need to consider their specific needs.

An alternative considered was to make the buffer size unbounded; but, this had the disadvantage that runaway computations would consume an unbounded amount of resources.

The "out" implicit variable of a JSP implementation class is of this type. If the page directive selects autoflush="true" then all the I/O operations on this class shall automatically flush the contents of the buffer if an overflow condition would result if the current operation were performed without a flush. If autoflush="false" then all the I/O operations on this class shall throw an IOException if performing the current operation would result in a buffer overflow condition.

See Also:
Writer, BufferedWriter, PrintWriter

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected boolean autoFlush</td>
</tr>
<tr>
<td>protected int bufferSize</td>
</tr>
<tr>
<td>static int DEFAULT_BUFFER</td>
</tr>
<tr>
<td>static int NO_BUFFER</td>
</tr>
<tr>
<td>static int UNBOUNDED_BUFFER</td>
</tr>
</tbody>
</table>
Fields inherited from class java.io.Writer

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock</td>
</tr>
</tbody>
</table>

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected JspWriter(int bufferSize, boolean autoFlush)</td>
</tr>
<tr>
<td>Protected constructor.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void clear()</td>
<td>Clear the contents of the buffer.</td>
</tr>
<tr>
<td>void clearBuffer()</td>
<td>Clears the current contents of the buffer.</td>
</tr>
<tr>
<td>void close()</td>
<td>Close the stream, flushing it first.</td>
</tr>
<tr>
<td>void flush()</td>
<td>Flush the stream.</td>
</tr>
<tr>
<td>int getBufferSize()</td>
<td>This method returns the size of the buffer used by the JspWriter.</td>
</tr>
<tr>
<td>int getRemaining()</td>
<td>This method returns the number of unused bytes in the buffer.</td>
</tr>
<tr>
<td>boolean isAutoFlush()</td>
<td>This method indicates whether the JspWriter is autoFlushing.</td>
</tr>
<tr>
<td>void newLine()</td>
<td>Write a line separator.</td>
</tr>
<tr>
<td>void print(boolean b)</td>
<td>Print a boolean value.</td>
</tr>
<tr>
<td>void print(char c)</td>
<td>Print a character.</td>
</tr>
<tr>
<td>void print(char[] s)</td>
<td></td>
</tr>
<tr>
<td>abstract void</td>
<td>Method</td>
</tr>
<tr>
<td>---------------</td>
<td>--------</td>
</tr>
<tr>
<td><code>println()</code></td>
<td></td>
</tr>
<tr>
<td><code>println(boolean x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(char x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(char[] x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(double x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(float x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(int x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(long x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(Object x)</code></td>
<td></td>
</tr>
<tr>
<td><code>println(String x)</code></td>
<td></td>
</tr>
</tbody>
</table>
Print a String and then terminate the line.

### Methods inherited from class java.io.Writer

- `append`, `append`, `append`, `write`, `write`, `write`, `write`, `write`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

**NO_BUFFER**

- `public static final int NO_BUFFER`

  Constant indicating that the Writer is not buffering output.

  **See Also:**
  - [Constant Field Values](#)

**DEFAULT_BUFFER**

- `public static final int DEFAULT_BUFFER`

  Constant indicating that the Writer is buffered and is using the implementation default buffer size.

  **See Also:**
  - [Constant Field Values](#)
UNBOUNDDED_BUFFER

```java
public static final int UNBOUNDDED_BUFFER
```

Constant indicating that the Writer is buffered and is unbounded; this is used in BodyContent.

See Also:
Constant Field Values

---

bufferSize

```java
protected int bufferSize
```

The size of the buffer used by the JspWriter.

---

autoFlush

```java
protected boolean autoFlush
```

Whether the JspWriter is autoflushing.

### Constructor Detail

```java
protected JspWriter(int bufferSize, boolean autoFlush)
```

bufferSize  JspWriter
autoFlush  JspWriter
JspWriter

protected JspWriter(int bufferSize,
                    boolean autoFlush)

Protected constructor.

Parameters:
  bufferSize - the size of the buffer to be used by the JspWriter
  autoFlush - whether the JspWriter should be autoflushing

Method Detail

abstract public void newLine() throws java.io.IOException

Throws
  java.io.IOException: I/O

newLine

public abstract void newLine()
    throws IOException

Write a line separator. The line separator string is defined by the system property line.separator, and is not necessarily a single newline ('\n') character.

Throws:
  IOException - If an I/O error occurs

abstract public void print(boolean b) throws java.io.IOException

boolean valueOf(boolean) JspWriter Writer

Throws
  java.io.IOException:
print

public abstract void print(boolean b) throws IOException

    Print a boolean value. The string produced by
    String.valueOf(boolean) is written to the JspWriter's buffer or, if no
    buffer is used, directly to the underlying writer.

Parameters:
    b - The boolean to be printed

Throws:
    IOException - If an error occurred while writing

abstract public void print(char c) throws java.io.IOException

    Print a character. The character is written to the JspWriter's buffer or,
    if no buffer is used, directly to the underlying writer.

Parameters:
    c - The char to be printed

Throws:
    IOException - If an error occurred while writing
abstract public void print(int i) throws java.io.IOException
    \text{valueOf(int)} JspWriter Writer
        \text{i} \quad \text{int}
    \text{Throws} \quad \text{java.io.IOException: toString(int)}

print

public abstract void print(int i)
    \text{throws IOException}

    Print an integer. The string produced by \text{String.valueOf(int)} is
    written to the JspWriter's buffer or, if no buffer is used, directly to the
    underlying writer.

Parameters:
    \text{i} - The \text{int} to be printed

Throws:
    \text{IOException} - If an error occurred while writing

See Also:
    \text{Integer.toString(int)}

abstract public void print(long l) throws java.io.IOException
    \text{valueOf(long)} JspWriter Writer
        \text{l} \quad \text{long}
    \text{Throws} \quad \text{java.io.IOException: toString(long)}

print

public abstract void print(long l)
    \text{throws IOException}

    Print a long integer. The string produced by \text{String.valueOf(long)} is
written to the JspWriter's buffer or, if no buffer is used, directly to the underlying writer.

**Parameters:**
- \( l \) - The long to be printed

**Throws:**
- \( 	ext{IOException} \) - If an error occurred while writing

**See Also:**
- \( 	ext{Long.toString(long)} \)

---

abstract public void print(float \( f \)) throws java.io.IOException

\( \text{valueOf(float)} \) JspWriter Writer

\( f \) float

**Throws**
- java.io.IOException:

**See also**
- \( 	ext{toString(float)} \)

---

**print**

public abstract void print(float \( f \))
throws IOException

Print a floating-point number. The string produced by \( \text{String.valueOf(float)} \) is written to the JspWriter's buffer or, if no buffer is used, directly to the underlying writer.

**Parameters:**
- \( f \) - The float to be printed

**Throws:**
- \( 	ext{IOException} \) - If an error occurred while writing

**See Also:**
- \( 	ext{Float.toString(float)} \)

---

abstract public void print(double \( d \)) throws java.io.IOException
valueOf(double)  JspWriter  Writer

\(d\)  \hspace{1cm} \text{double}

Throws  \hspace{1cm} \text{java.io.IOException: toString(double)}

See also

print

public abstract void print(double d)  
throws IOException

Print a double-precision floating-point number. The string produced by String.valueOf(double) is written to the JspWriter’s buffer or, if no buffer is used, directly to the underlying writer.

Parameters:

\(d\) - The double to be printed

Throws:

IOException - If an error occurred while writing

See Also:

Double.toString(double)

abstract public void print(char[] s) throws 
java.io.IOException  
JspWriter  Writer

\(s\)

Throws  \hspace{1cm} \text{NullPointerException: s null}

Throws  \hspace{1cm} \text{java.io.IOException:}

print

public abstract void print(char[] s)  
throws IOException

Print an array of characters. The characters are written to the JspWriter's buffer or, if no buffer is used, directly to the underlying
abstract public void print(String s) throws IOException

Print a string. If the argument is null then the string "null" is printed. Otherwise, the string's characters are written to the JspWriter's buffer or, if no buffer is used, directly to the underlying writer.

Parameters:
  s - The String to be printed

Throws:
  IOException - If an error occurred while writing

See also
  toString()
print

public abstract void print(Object obj)
    throws IOException

Print an object. The string produced by the String.valueOf(Object) method is written to the JspWriter's buffer or, if no buffer is used, directly to the underlying writer.

Parameters:
    obj - The object to be printed

Throws:
    IOException - If an error occurred while writing

See Also:
    Object.toString()

abstract public void println() throws java.io.IOException

Terminate the current line by writing the line separator string. The line separator string is defined by the system property line.separator, and is not necessarily a single newline character ('\n').

Throws:
    IOException - If an error occurred while writing

abstract public void println(boolean x) throws
javax.printing

---

**println**

```java
public abstract void println(boolean x)
throws IOException
```

Print a boolean value and then terminate the line. This method behaves as though it invokes `print(boolean)` and then `println()`.

**Parameters:**
- `x` - the boolean to write

**Throws:**
- `IOException` - If an error occurred while writing

---

**abstract public void println(char x) throws java.io.IOException**

```java
public abstract void println(char x)
throws IOException
```

Print a character and then terminate the line. This method behaves as though it invokes `print(char)` and then `println()`.

**Parameters:**
- `x` - the char to write

**Throws:**
IOException - If an error occurred while writing

abstract public void println(int x) throws java.io.IOException

public abstract void println(int x) throws IOException

Print an integer and then terminate the line. This method behaves as though it invokes print(int) and then println().

Parameters:
x - the int to write

Throws:
IOException - If an error occurred while writing

abstract public void println(long x) throws java.io.IOException

public abstract void println(long x) throws IOException

Print a long integer and then terminate the line. This method behaves as though it invokes print(long) and then println().
abstract public void println(float x) throws IOException

print ln

public abstract void println(float x)
throws IOException

Print a floating-point number and then terminate the line. This method behaves as though it invokes print(float) and then println().

Parameters:
  x - the float to write

Throws:
  IOException - If an error occurred while writing

abstract public void println(double x) throws IOException

print ln

public abstract void println(double x)
throws IOException

Print a double-precision number and then terminate the line. This method behaves as though it invokes print(double) and then println().

Parameters:
  x - the double to write

Throws:
  IOException - If an error occurred while writing
public abstract void println(double x)  
   throws IOException

Print a double-precision floating-point number and then terminate the line. This method behaves as though it invokes print(double) and then println().

Parameters:
        x - the double to write

Throws:
       IOException - If an error occured while writing

abstract public void println(char[] x) throws java.io.IOException

print(char[]) prin
        x                         char[]
        char[]
Thros         java.io.IOException:

println

public abstract void println(char[] x)  
   throws IOException

Print an array of characters and then terminate the line. This method behaves as though it invokes print(char[]) and then println().

Parameters:
        x - the char[] to write

Throws:
       IOException - If an error occured while writing

abstract public void println(String x) throws java.io.IOException

String
        #print(String)   #pr

x                 String
Throws: java.io.IOException:

println

public abstract void println(String x) throws IOException

Print a String and then terminate the line. This method behaves as though it invokes print(String) and then println().

Parameters:
  x - the String to write

Throws:
  IOException - If an error occurred while writing

abstract public void println(Object x) throws java.io.IOException

Print an Object and then terminate the line. This method behaves as though it invokes print(Object) and then println().

Parameters:
  x - the Object to write

Throws:
  IOException - If an error occurred while writing
abstract public void clear() throws java.io.IOException

clear

epublic abstract void clear()
    throws IOException

    Clear the contents of the buffer. If the buffer has been already been flushed then the clear operation shall throw an IOException to signal the fact that some data has already been irrevocably written to the client response stream.

    Throws:
        IOException - If an I/O error occurs

------------------------------------------------------------------------

abstract public void clearBuffer() throws java.io.IOException

clearBuffer

epublic abstract void clearBuffer()
    throws IOException

    Clears the current contents of the buffer. Unlike clear(), this method will not throw an IOException if the buffer has already been flushed. It merely clears the current content of the buffer and returns.

    Throws:
        IOException - If an I/O error occurs
abstract public void flush() throws java.io.IOException

write()  flush()  Writer
OutputStream

write()  flush()  IOException

Throws

java.io.IOException: I/O

flush

public abstract void flush()

throws IOException

Flush the stream. If the stream has saved any characters from the various write() methods in a buffer, write them immediately to their intended destination. Then, if that destination is another character or byte stream, flush it. Thus one flush() invocation will flush all the buffers in a chain of Writers and OutputStreams.

The method may be invoked indirectly if the buffer size is exceeded.

Once a stream has been closed, further write() or flush() invocations will cause an IOException to be thrown.

Specified by:

flush in interface Flushable

Specified by:

flush in class Writer

Throws:

IOException - If an I/O error occurs

abstract public void close() throws java.io.IOException
JspWriter JSP close()
flush()

Throws java.io.IOException: I/O

close

public abstract void close()
    throws IOException

Close the stream, flushing it first.

This method needs not be invoked explicitly for the initial JspWriter as the code generated by the JSP container will automatically include a call to close().

Closing a previously-closed stream, unlike flush(), has no effect.

Specified by:
    close in interface Closeable
Specified by:
    close in class Writer
Throws:
    IOException - If an I/O error occurs

public int getBufferSize()
JspWriter

    return 0

getBufferSize

public int getBufferSize()

This method returns the size of the buffer used by the JspWriter.
abstract public int getRemaining()

    return

getRemaining

class public abstract int getRemaining()

    This method returns the number of unused bytes in the buffer.

    Returns:
    the number of bytes unused in the buffer

public boolean isAutoFlush()

    return  JspWriter  IOException

isAutoFlush

    This method indicates whether the JspWriter is autoFlushing.

    Returns:
    if this JspWriter is auto flushing or throwing IOExceptions on
    buffer overflow conditions
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface JTAStats

extends Stats

Implements: Stats

JTA

Specifies the statistics provided by a JTA resource.

### Method Summary

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getActiveCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of active transactions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getCommittedCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of committed transactions.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getRolledbackCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of rolled-back transactions.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Stats

getStatistic, getStatisticNames, getStatistics

### Method Detail

public CountStatistic getActiveCount()
getActiveCount

**CountStatistic** getActiveCount()

Number of active transactions.

---

public **CountStatistic** getCommittedCount()

getCommittedCount

**CountStatistic** getCommittedCount()

Number of committed transactions.

---

public **CountStatistic** getRolledbackCount()

getRolledbackCount

**CountStatistic** getRolledbackCount()

Number of rolled-back transactions.
to license terms.

PS:
javax.management.j2ee.statistics Interface JVMStats

All Superinterfaces:
   Stats

public interface JVMStats
extends Stats
Implements: Stats

Java VM

Specifies the statistics provided by a Java VM.

Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoundedRangeStatistic</td>
<td>getHeapSize()</td>
<td>Size of the JVM's heap.</td>
</tr>
<tr>
<td>CountStatistic</td>
<td>getUpTime()</td>
<td>Amount of time the JVM has been running.</td>
</tr>
</tbody>
</table>

Methods inherited from interface
javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail

public CountStatistic getUpTime()

JVM
**getUptime**

`CountStatistic` `getUptime()`

Amount of time the JVM has been running.

**public `BoundedRangeStatistic` `getHeapSize()`**

`JVM`

**getHeapSize**

`BoundedRangeStatistic` `getHeapSize()`

Size of the JVM's heap.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.registry.infomodel Interface Key

public interface Key

RegistryObject  DCE 128 UUID

See also  javax.xml.registry.infomodel.RegistryObject

Represents a unique key that identifies a RegistryObject. Must be a DCE 128 UUID.

Author:
  Farrukh S. Najmi

See Also:
  RegistryObject

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>String getId()</td>
<td>Returns the unique Id of this key.</td>
</tr>
<tr>
<td>void</td>
<td>void setId(String id)</td>
<td>Sets the unique id associated with this key.</td>
</tr>
</tbody>
</table>

Method Detail

public String getId() throws JAXRException
Id NULL

    0

        return ID

Throws JAXRException: JAXR
**getld**

```java
String getId() throws JAXRException
```

Returns the unique Id of this key. Default is a NULL String.

**Capability Level: 0**

**Returns:**
- the id for this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public void setId(String id) throws JAXRException**

**ID**

```
0
```

```java
id
```

**Throws**
- `JAXRException`: JAXR

---

**setId**

```java
void setId(String id) throws JAXRException
```

Sets the unique id associated with this key.

**Capability Level: 0**

**Parameters:**
- `id` - the id being defined for this object

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi Interface LazyAssociatableConnectionManager

public interface LazyAssociatableConnectionManager

ConnectionManager
　version 1.0

This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation. An implementation of this interface must support the lazy connection association optimization.

Version: 1.0
Author: Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void associateConnection(Object connection,</td>
<td>This method is called by a resource adapter (that is capable of lazy connection association optimization) in order to lazily associate a connection object with a ManagedConnection instance.</td>
</tr>
<tr>
<td>ManagedConnectionFactory mcf,</td>
<td></td>
</tr>
<tr>
<td>ConnectionRequestInfo cxReqInfo) throws</td>
<td>ResourceException</td>
</tr>
<tr>
<td>ResourceException</td>
<td></td>
</tr>
</tbody>
</table>

Method Detail

public void associateConnection(Object connection, ManagedConnectionFactory mcf, ConnectionRequestInfo cxReqInfo) throws ResourceException
associateConnection

void associateConnection(Object connection,
                         ManagedConnectionFactory mcf,
                         ConnectionRequestInfo cxReqInfo)
throws ResourceException

This method is called by a resource adapter (that is capable of lazy connection association optimization) in order to lazily associate a connection object with a ManagedConnection instance.

Parameters:
connection - the connection object that is to be associated.
mcf - The ManagedConnectionFactory instance that was originally used to create the connection object.
cxReqInfo - connection request information. This information must be the same as that used to originally create the connection object.

Throws:
ResourceException - Generic exception.
ApplicationServerInternalException - Application server specific exception.
SecurityException - Security related error.
ResourceAllocationException - Failed to allocate system resources for connection request.
ResourceAdapterInternalException - Resource adapter related error condition.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Interface LazyEnlistableConnectionManager

public interface LazyEnlistableConnectionManager

    ConnectionManager (enlistment)
        version 1.0

This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation. An implementation of this interface must support the lazy transaction enlistment optimization.

Version:
    1.0

Author:
    Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>void lazyEnlist(ManagedConnection mc)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This method is called by a resource adapter (that is capable of lazy transaction enlistment optimization) in order to lazily enlist a connection object with a XA transaction.</td>
</tr>
</tbody>
</table>

Method Detail

public void lazyEnlist(ManagedConnection mc) throws ResourceException
    XA
        mc ManagedConnection
lazyEnlist

void lazyEnlist(ManagedConnection mc)
throws ResourceException

This method is called by a resource adapter (that is capable of lazy transaction enlistment optimization) in order to lazily enlist a connection object with a XA transaction.

Parameters:
  mc - The ManagedConnection instance that needs to be lazily associated.

Throws:
  ResourceException - Generic exception.
  ApplicationServerInternalException - Application server specific exception.
  ResourceAllocationException - Failed to allocate system resources for connection request.
  ResourceAdapterInternalException - Resource adapter related error condition.
public interface LazyEnlistableManagedConnection

ManagedConnection (enlistment)
version 1.0

This is a mix-in interface that may be optionally implemented by a ManagedConnection implementation. An implementation of this interface must support the lazy transaction enlistment optimization.

Version: 1.0
Author: Ram Jeyaraman

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
**javax.faces.validator**  **Class LengthValidator**

**java.lang.Object**

- javax.faces.validator.LengthValidator

**All Implemented Interfaces:**
  - `EventListener`, `StateHolder`, `Validator`

```java
public class LengthValidator
extends Object
implements Validator, StateHolder

Implements: Validator, StateHolder
```

**LengthValidator**  **Validator**  **String**

- `toString`  **String**
- `Validator`  `maximum`  **String**  **String**
  - `MAXIMUM_MESSAGE_ID`  **ValidatorException**
- `Validator`  `minimum`  **String**  **String**
  - `MINIMUM_MESSAGE_ID`  **ValidatorException**

**ValidatorException**  **ID**  **javax.faces.Number**  **Converter**  **Locale**

---

**LengthValidator** is a **Validator** that checks the number of characters in the String representation of the value of the associated component. The following algorithm is implemented:

- Convert the passed value to a String, if necessary, by calling its `toString()` method.
- If a `maximum` property has been configured on this **validator**, check the length of the converted String against this limit. If the String length is larger than the specified maximum, throw a
ValidatorException containing a a MAXIMUM_MESSAGE_ID message.
- If a minimum property has been configured on this validator, check the length of the converted String against this limit. If the String length is less than the specified minimum, throw a ValidatorException containing a a MINIMUM_MESSAGE_ID message.

For all of the above cases that cause a ValidatorException to be thrown, if there are parameters to the message that match up with validator parameters, the values of these parameters must be converted using the Converter registered in the application under the converter id javax.faces.Number. This allows the values to be localized according to the current Locale.

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXIMUM_MESSAGE_ID</td>
<td>The message identifier of the FacesMessage to be created if the maximum length check fails.</td>
</tr>
<tr>
<td>MINIMUM_MESSAGE_ID</td>
<td>The message identifier of the FacesMessage to be created if the minimum length check fails.</td>
</tr>
<tr>
<td>VALIDATOR_ID</td>
<td>The standard validator id for this validator.</td>
</tr>
</tbody>
</table>

### Fields inherited from interface javax.faces.validator.Validator

NOT_IN_RANGE_MESSAGE_ID

### Constructor Summary

**LengthValidator()**

Construct a Validator with no preconfigured limits.

**LengthValidator(int maximum)**

Construct a Validator with the specified preconfigured limit.

**LengthValidator(int maximum, int minimum)**
Construct a Validator with the specified preconfigured limits.

## Method Summary

<table>
<thead>
<tr>
<th>boolean equals(Object otherObj)</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getMaximum()</td>
</tr>
<tr>
<td>     Return the maximum length to be enforced by this Validator, or 0 if the maximum has not been set.</td>
</tr>
<tr>
<td>int getMinimum()</td>
</tr>
<tr>
<td>     Return the minimum length to be enforced by this Validator, or 0 if the minimum has not been set.</td>
</tr>
<tr>
<td>int hashCode()</td>
</tr>
<tr>
<td>boolean isTransient()</td>
</tr>
<tr>
<td>     If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void restoreState(FacesContext context, Object state)</td>
</tr>
<tr>
<td>     Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object saveState(FacesContext context)</td>
</tr>
<tr>
<td>     Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void setMaximum(int maximum)</td>
</tr>
<tr>
<td>     Set the maximum length to be enforced by this Validator.</td>
</tr>
<tr>
<td>void setMinimum(int minimum)</td>
</tr>
<tr>
<td>     Set the minimum length to be enforced by this Validator.</td>
</tr>
<tr>
<td>void setTransient(boolean transientValue)</td>
</tr>
<tr>
<td>     Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void validate(FacesContext context, UIComponent component, Object value)</td>
</tr>
<tr>
<td>     Perform the correctness checks implemented by this Validator against the specified UIComponent.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
VALIDATOR_ID

public static final String VALIDATOR_ID

The standard validator id for this validator.

See Also:
Constant Field Values

MAXIMUM_MESSAGE_ID

public static final String MAXIMUM_MESSAGE_ID

The message identifier of the FacesMessage to be created if the maximum length check fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the configured maximum length.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

MINIMUM_MESSAGE_ID
public static final String MINIMUM_MESSAGE_ID

The message identifier of the FacesMessage to be created if the minimum length check fails. The message format string for this message may optionally include the following placeholders:

- \{0\} replaced by the configured minimum length.
- \{1\} replaced by a string whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail

public LengthValidator()

Validator

LengthValidator

public LengthValidator()

Construct a Validator with no preconfigured limits.

public LengthValidator(int maximum)

Validator

maximum

LengthValidator
public LengthValidator(int maximum)

Construct a **Validator** with the specified preconfigured limit.

**Parameters:**
- `maximum` - Maximum value to allow

public LengthValidator(int maximum, int minimum)

**Validator**

```java
  public int getMaximum()
  
  getMaximum
```
Return the maximum length to be enforced by this `Validator`, or 0 if the maximum has not been set.

```java
public void setMaximum(int maximum)
```

Set the maximum length to be enforced by this `Validator`. 

**Parameters:**
- `maximum` - The new maximum value

```java
public int getMinimum()
```

Return the minimum length to be enforced by this `Validator`, or 0 if the minimum has not been set.

```java
public void setMinimum(int minimum)
```

```java
Validator
```
**setMinimum**

```java
public void setMinimum(int minimum)
```

Set the minimum length to be enforced by this **Validator**.

**Parameters:**

- `minimum` - The new minimum value

---

**public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException**

Throws

- `NullPointerException`: `NullPointerException`
- `context`: `null`
- `component`: `null`

Throws

- **ValidatorException**: `NullPointerException`

---

**validate**

```java
public void validate(FacesContext context, UIComponent component, Object value)
```

throws **ValidatorException**

**Description copied from interface:** Validator

Perform the correctness checks implemented by this **Validator** against the specified **UIComponent**. If any violations are found, a **ValidatorException** will be thrown containing the **FacesMessage** describing the failure.

**Specified by:**

- **validate** in interface **Validator**

**Parameters:**

- `context` - FacesContext for the request we are processing
- `component` - UIComponent we are checking for correctness
value - the value to validate

**Throws:**
- NullPointerException - if context or component is null
- ValidatorException - if validation fails

---

```java
public boolean equals(Object otherObj)
```

**equals**

```java
public boolean equals(Object otherObj)
```

**Overrides:**
- equals in class Object

---

```java
public int hashCode()
```

**hashCode**

```java
public int hashCode()
```

**Overrides:**
- hashCode in class Object

---

```java
public Object saveState(FacesContext context)
```

**saveState**

```java
public Object saveState(FacesContext context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.
If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**

`saveState` in interface `StateHolder`

```
public void restoreState(FacesContext context, Object state)
```

```
restoreState
```

```
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface:** `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.
public boolean isTransient()

isTransient

public boolean isTransient()

  Description copied from interface: StateHolder

  If true, the Object implementing this interface must not participate in state saving or restoring.

  Specified by:
  isTransient in interface StateHolder

public void setTransient(boolean transientValue)

setTransient

public void setTransient(boolean transientValue)

  Description copied from interface: StateHolder

  Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

  Specified by:
  setTransient in interface StateHolder

Parameters:
  transientValue - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.lifecycle Class Lifecycle

java.lang.Object
   javax.faces.lifecycle.Lifecycle

public abstract class Lifecycle
extends Object

Lifecycle  JavaServer Faces  JavaServer Faces

Lifecycle manages the processing of the entire lifecycle of a particular JavaServer Faces request. It is responsible for executing all of the phases that have been defined by the JavaServer Faces Specification, in the specified order, unless otherwise directed by activities that occurred during the execution of each phase.

An instance of Lifecycle is created by calling the getLifecycle() method of LifecycleFactory, for a specified lifecycle identifier. Because this instance is shared across multiple simultaneous requests, it must be implemented in a thread-safe manner.

Constructor Summary

| Lifecycle() |

Method Summary

<p>| addPhaseListener(PhaseListener listener) |
| Register a new PhaseListener instance that is |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract void interested(FacesContext context)</code></td>
<td>Execute all of the phases of the request processing lifecycle, up to but not including the Render Response phase, as described in the JavaServer Faces Specification, in the specified order.</td>
</tr>
<tr>
<td><code>abstract void execute(FacesContext context)</code></td>
<td>Execute all of the phases of the request processing lifecycle, up to but not including the Render Response phase, as described in the JavaServer Faces Specification, in the specified order.</td>
</tr>
<tr>
<td><code>abstract void removePhaseListener(PhaseListener listener)</code></td>
<td>Deregister an existing PhaseListener instance that is no longer interested in being notified before and after the processing for standard phases of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>abstract void render(FacesContext context)</code></td>
<td>Execute the Render Response phase of the request processing lifecycle, unless the responseComplete() method has been called on the FacesContext instance associated with the current request.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
abstract public void addPhaseListener(PhaseListener listener)

*PhaseListener*

<table>
<thead>
<tr>
<th>listener</th>
<th>PhaseListener</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>NullPointerException: listener null</td>
</tr>
</tbody>
</table>

addPhaseListener

public abstract void addPhaseListener(PhaseListener listener)

Register a new PhaseListener instance that is interested in being notified before and after the processing for standard phases of the request processing lifecycle.

Parameters:

- listener - The PhaseListener to be registered

Throws:

- NullPointerException - if listener is null

abstract public void execute(FacesContext context) throws FacesException

JavaServer Faces

renderResponse() responseComplete()

<table>
<thead>
<tr>
<th>context</th>
<th>FacesContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>FacesException:</td>
</tr>
<tr>
<td>Throws</td>
<td>NullPointerException: context null</td>
</tr>
</tbody>
</table>
**execute**

```java
public abstract void execute(FacesContext context)
    throws FacesException
```

Execute all of the phases of the request processing lifecycle, up to but not including the *Render Response* phase, as described in the JavaServer Faces Specification, in the specified order. The processing flow can be affected (by the application, by components, or by event listeners) by calls to the `renderResponse()` or `responseComplete()` methods of the `FacesContext` instance associated with the current request.

**Parameters:**
- `context` - FacesContext for the request to be processed

**Throws:**
- `FacesException` - if thrown during the execution of the request processing lifecycle
- `NullPointerException` - if context is null

**abstract public PhaseListener[] getPhaseListeners()**

```java
Lifecycle PhaseListener 0
```

**getPhaseListeners**

```java
public abstract PhaseListener[] getPhaseListeners()
```

Return the set of registered `PhaseListener`s for this `Lifecycle` instance. If there are no registered listeners, a zero-length array is returned.

**abstract public void removePhaseListener(PhaseListener listener)**
**PhaseListener**

*listener*  
Throws  
NullPointerException: listener null

**removePhaseListener**

public abstract void removePhaseListener(*PhaseListener* listener)

Deregister an existing *PhaseListener* instance that is no longer interested in being notified before and after the processing for standard phases of the request processing lifecycle. If no such listener instance has been registered, no action is taken.

**Parameters:**
- listener - The *PhaseListener* to be deregistered

**Throws:**
- NullPointerException - if listener is null

**abstract public void render**(*FacesContext* context) throws *FacesException*

<table>
<thead>
<tr>
<th>context</th>
<th>FacesContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td><em>FacesException</em>:</td>
</tr>
<tr>
<td>Throws</td>
<td>NullPointerException: context null</td>
</tr>
</tbody>
</table>

**render**

public abstract void render(*FacesContext* context) throws *FacesException*

Execute the *Render Response* phase of the request processing lifecycle, unless the responseComplete() method has been called on
the `FacesContext` instance associated with the current request.

**Parameters:**
context - FacesContext for the request being processed

**Throws:**
- `FacesException` - if an exception is thrown during the execution of the request processing lifecycle
- `NullPointerException` - if context is null
javax.faces.lifecycle Class LifecycleFactory

java.lang.Object
  └─javax.faces.lifecycle.LifecycleFactory

public abstract class LifecycleFactory
  extends Object

LifecycleFactory Lifecycle JavaServer Faces Lifecycle Lifecycle

JavaServer Faces Web LifecycleFactory

LifecycleFactory factory = (LifecycleFactory)
FactoryFinder.getFactory(FactoryFinder.LIFECYCLE_FACTORY);

LifecycleFactory is a factory object that creates (if needed) and returns Lifecycle instances. Implementations of JavaServer Faces must provide at least a default implementation of Lifecycle. Advanced implementations (or external third party libraries) MAY provide additional Lifecycle implementations (keyed by lifecycle identifiers) for performing different types of request processing on a per-request basis.

There must be one LifecycleFactory instance per web application that is utilizing JavaServer Faces. This instance can be acquired, in a portable manner, by calling:

   LifecycleFactory factory = (LifecycleFactory)
     FactoryFinder.getFactory(FactoryFinder.LIFECYCLE_FACTORY);
### Constructor Summary

**LifecycleFactory()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addLifecycle(String lifecycleId, Lifecycle lifecycle)</strong></td>
<td>Register a new Lifecycle instance, associated with the specified lifecycleId, to be supported by this LifecycleFactory.</td>
</tr>
<tr>
<td><strong>getLifecycle(String lifecycleId)</strong></td>
<td>Create (if needed) and return a Lifecycle instance for the specified lifecycle identifier.</td>
</tr>
<tr>
<td><strong>getLifecycleIds()</strong></td>
<td>Return an Iterator over the set of lifecycle identifiers supported by this factory.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

**DEFAULT_LIFECYCLE**

public static final String DEFAULT_LIFECYCLE

The lifecycle identifier for the default Lifecycle instance for this JavaServer Faces implementation.
JavaServer Faces implementation.

See Also:
   Constant Field Values

### Constructor Detail

```java
public LifecycleFactory()
```

**LifecycleFactory**

```java
public LifecycleFactory()
```

### Method Detail

```java
abstract public void addLifecycle(String lifecycleId, Lifecycle lifecycle)
```

<table>
<thead>
<tr>
<th>lifecycleId</th>
<th>Lifecycle</th>
<th>LifecycleFactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lifecycle</td>
<td>Web</td>
<td></td>
</tr>
</tbody>
</table>

**Throws**  
IllegalArgumentException: lifecycleId  
NullPointerException: lifecycleId lifecycle null

```java
addLifecycle
```

public abstract void addLifecycle(String lifecycleId, Lifecycle lifecycle)

Register a new Lifecycle instance, associated with the specified lifecycleId, to be supported by this LifecycleFactory. This method may be called at any time, and makes the corresponding Lifecycle
instance available throughout the remaining lifetime of this web application.

**Parameters:**
- lifecycleId - Identifier of the new Lifecycle lifecycle - Lifecycle instance that we are registering

**Throws:**
- IllegalArgumentException - if a Lifecycle with the specified lifecycleId has already been registered
- NullPointerException - if lifecycleId or lifecycle is null

```java
abstract public Lifecycle getLifecycle(String lifecycleId)
```

```java
public abstract Lifecycle getLifecycle(String lifecycleId)
```

Create (if needed) and return a Lifecycle instance for the specified lifecycle identifier. The set of available lifecycle identifiers is available via the getLifecycleIds() method.

Each call to getLifecycle() for the same lifecycleId, from within the same web application, must return the same Lifecycle instance.

**Parameters:**
- lifecycleId - Lifecycle identifier of the requested Lifecycle instance

**Throws:**
IllegalArgumentException - if no Lifecycle instance can be returned for the specified identifier

NullPointerException - if lifecycleId is null

abstract public java.util.Iterator<E> getLifecycleIds()

Iterator LifecycleFactory.DEFAULT_LIFECYCLE

getLifecycleIds

public abstract Iterator<String> getLifecycleIds()

Return an Iterator over the set of lifecycle identifiers supported by this factory. This set must include the value specified by LifecycleFactory.DEFAULT_LIFECYCLE.
javax.xml.registry Interface LifeCycleManager

All Known Subinterfaces:
   BusinessLifeCycleManager

public interface LifeCycleManager

Implemented by: BusinessLifeCycleManager

LifeCycleManager API

JAXR UnsupportedCapabilityException

The LifeCycleManager interface is the main interface in the API for managing life cycle operations on objects defined by the information model.

The factory methods of this interface must throw an UnsupportedCapabilityException if the client attempts to create an instance of an infomodel interface that is not supported by the capability level of the JAXR provider.

Author:
   Farrukh S. Najmi

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>ASSOCIATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>AUDITABLE_EVENT</td>
</tr>
<tr>
<td>static String</td>
<td>CLASSIFICATION</td>
</tr>
<tr>
<td>static String</td>
<td>CLASSIFICATION_SCHEME</td>
</tr>
<tr>
<td>----------------</td>
<td>-------------------------------</td>
</tr>
<tr>
<td>static String</td>
<td>CONCEPT</td>
</tr>
<tr>
<td>static String</td>
<td>EMAIL_ADDRESS</td>
</tr>
<tr>
<td>static String</td>
<td>EXTERNAL_IDENTIFIER</td>
</tr>
<tr>
<td>static String</td>
<td>EXTERNAL_LINK</td>
</tr>
<tr>
<td>static String</td>
<td>EXTRINSIC_OBJECT</td>
</tr>
<tr>
<td>static String</td>
<td>INTERNATIONAL_STRING</td>
</tr>
<tr>
<td>static String</td>
<td>KEY</td>
</tr>
<tr>
<td>static String</td>
<td>LOCALIZED_STRING</td>
</tr>
<tr>
<td>static String</td>
<td>ORGANIZATION</td>
</tr>
<tr>
<td>static String</td>
<td>PERSON_NAME</td>
</tr>
<tr>
<td>static String</td>
<td>POSTAL_ADDRESS</td>
</tr>
<tr>
<td>static String</td>
<td>REGISTRY_ENTRY</td>
</tr>
<tr>
<td>static String</td>
<td>REGISTRY_PACKAGE</td>
</tr>
<tr>
<td>static String</td>
<td>SERVICE</td>
</tr>
<tr>
<td>static String</td>
<td>SERVICE_BINDING</td>
</tr>
<tr>
<td>static String</td>
<td>SLOT</td>
</tr>
</tbody>
</table>
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Association</strong></td>
<td><code>createAssociation(RegistryObject targetObject, Concept associationType)</code></td>
<td>Creates an Association instance using the specified RegistryObject and Concept.</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td><code>createClassification(ClassificationScheme scheme, Concept concept)</code></td>
<td>Creates a Classification instance for an internal Classification using the specified Concept.</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td><code>createClassification(ClassificationScheme scheme, InternationalString name, String value)</code></td>
<td>Creates a Classification instance for an external Classification using the specified InternationalString name and String value.</td>
</tr>
<tr>
<td><strong>Classification</strong></td>
<td><code>createClassification(ClassificationScheme scheme, String name, String description)</code></td>
<td>Creates a Classification instance for an external Classification using the specified String name and InternationalString description.</td>
</tr>
<tr>
<td><strong>ClassificationScheme</strong></td>
<td><code>createClassificationScheme(ClassificationScheme scheme, Concept concept)</code></td>
<td>Creates a ClassificationScheme from a Concept.</td>
</tr>
<tr>
<td><strong>ClassificationScheme</strong></td>
<td><code>createClassificationScheme(ClassificationScheme scheme, InternationalString name, InternationalString description)</code></td>
<td>Creates a ClassificationScheme given the specified InternationalString parameters.</td>
</tr>
<tr>
<td><strong>ClassificationScheme</strong></td>
<td><code>createClassificationScheme(String name, String description)</code></td>
<td>Creates a ClassificationScheme given the specified String parameters.</td>
</tr>
<tr>
<td>Class</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Concept</td>
<td><code>createConcept(RegistryObject parent, InternationalString value)</code></td>
<td>Creates a Concept instance using the specified parameters, where the name is an InternationalString.</td>
</tr>
<tr>
<td>Concept</td>
<td><code>createConcept(RegistryObject parent, String name, String type)</code></td>
<td>Creates a Concept instance using the specified parameters, where the name is a String.</td>
</tr>
<tr>
<td>EmailAddress</td>
<td><code>createEmailAddress(String address)</code></td>
<td>Creates an EmailAddress instance using an address as the parameter.</td>
</tr>
<tr>
<td>EmailAddress</td>
<td><code>createEmailAddress(String address, String type)</code></td>
<td>Creates an EmailAddress instance using both an address and a type as parameters.</td>
</tr>
<tr>
<td>ExternalIdentifier</td>
<td><code>createExternalIdentifier(ClassificationScheme identificationScheme, InternationalString name, String value)</code></td>
<td>Creates an ExternalIdentifier instance using the specified parameters, where the name is an InternationalString.</td>
</tr>
<tr>
<td>ExternalIdentifier</td>
<td><code>createExternalIdentifier(ClassificationScheme identificationScheme, String name, String value)</code></td>
<td>Creates an ExternalIdentifier instance using the specified parameters, where the name is a String.</td>
</tr>
<tr>
<td>ExternalLink</td>
<td><code>createExternalLink(String externalURI, InternationalString description)</code></td>
<td>Creates an ExternalLink instance using the specified parameters, where the description is an InternationalString.</td>
</tr>
<tr>
<td>ExternalLink</td>
<td><code>createExternalLink(String externalURI, String description)</code></td>
<td>Creates an ExternalLink instance using the specified parameters, where the description is a String.</td>
</tr>
<tr>
<td>ExtrinsicObject</td>
<td><code>createExtrinsicObject(DataHandler repositoryItem)</code></td>
<td>Creates an ExtrinsicObject instance using the specified parameters.</td>
</tr>
<tr>
<td>InternationalString</td>
<td><code>createInternationalString()</code></td>
<td>Creates an empty InternationalString instance.</td>
</tr>
<tr>
<td>InternationalString</td>
<td><code>createInternationalString(Locale l, String s)</code></td>
<td>Creates an InternationalString instance using the specified Locale and String parameters.</td>
</tr>
<tr>
<td>Class</td>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InternationalString</td>
<td><code>createInternationalString(String s)</code></td>
<td>Creates an InternationalString instance using a String and the default Locale.</td>
</tr>
<tr>
<td>Key</td>
<td><code>createKey(String id)</code></td>
<td>Creates a Key instance from an ID.</td>
</tr>
<tr>
<td>LocalizedString</td>
<td><code>createLocalizedString(Locale l, String s)</code></td>
<td>Creates a LocalizedString instance using the specified String parameters.</td>
</tr>
<tr>
<td>LocalizedString</td>
<td><code>createLocalizedString(Locale l, String s, String charSetName)</code></td>
<td>Creates a LocalizedString instance using the specified String, and character set parameters.</td>
</tr>
<tr>
<td>Object</td>
<td><code>createObject(String interfaceName)</code></td>
<td>Creates instances of information model interfaces (factory method).</td>
</tr>
<tr>
<td>Organization</td>
<td><code>createOrganization(InternationalString name)</code></td>
<td>Creates an Organization instance using the specified where the name is an InternationalString.</td>
</tr>
<tr>
<td>Organization</td>
<td><code>createOrganization(String name)</code></td>
<td>Creates an Organization instance using the specified where the name is a String.</td>
</tr>
<tr>
<td>PersonName</td>
<td><code>createPersonName(String fullName)</code></td>
<td>Creates a PersonName instance using the specified full name.</td>
</tr>
<tr>
<td>PersonName</td>
<td><code>createPersonName(String firstName, String middleName, String lastName)</code></td>
<td>Creates a PersonName instance using the specified and last names.</td>
</tr>
<tr>
<td>PostalAddress</td>
<td><code>createPostalAddress(String streetNumber, String street, String city, String stateOrProvince, String country, String postalCode, String type)</code></td>
<td>Creates a PostalAddress instance using the specified parameters.</td>
</tr>
<tr>
<td>RegistryPackage</td>
<td><code>createRegistryPackage(InternationalString name)</code></td>
<td>Creates a RegistryPackage instance using the specified where the name is an InternationalString.</td>
</tr>
<tr>
<td>RegistryPackage</td>
<td><code>createRegistryPackage(String name)</code></td>
<td>Creates a RegistryPackage instance using the specified where the name is a String.</td>
</tr>
<tr>
<td>Service</td>
<td><code>createService(InternationalString name)</code></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a Service instance using the specified name, where the name is an InternationalString.</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td><code>createService(String name)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a Service instance using the specified name, where the name is a String.</td>
<td></td>
</tr>
<tr>
<td>ServiceBinding</td>
<td><code>createServiceBinding()</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates an empty ServiceBinding instance.</td>
<td></td>
</tr>
<tr>
<td>Slot</td>
<td><code>createSlot(String name, Collection values, String slotType)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a Slot instance using the specified parameters, where the value is a Collection of Strings.</td>
<td></td>
</tr>
<tr>
<td>Slot</td>
<td><code>createSlot(String name, String value, String slotType)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a Slot instance using the specified parameters, where the value is a String.</td>
<td></td>
</tr>
<tr>
<td>SpecificationLink</td>
<td><code>createSpecificationLink()</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates an empty SpecificationLink instance.</td>
<td></td>
</tr>
<tr>
<td>TelephoneNumber</td>
<td><code>createTelephoneNumber()</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates an empty TelephoneNumber instance.</td>
<td></td>
</tr>
<tr>
<td>User</td>
<td><code>createUser()</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates an empty User instance.</td>
<td></td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteObjects(Collection keys)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deletes one or more previously submitted objects using the object keys.</td>
<td></td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deleteObjects(Collection keys, String objectType)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deletes one or more previously submitted objects using the object keys and a specified objectType attribute.</td>
<td></td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>deprecateObjects(Collection keys)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Deprecates one or more previously submitted objects.</td>
<td></td>
</tr>
<tr>
<td>RegistryService</td>
<td><code>getRegistryService()</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Returns the parent RegistryService that created this object.</td>
<td></td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>saveObjects(Collection objects)</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Saves one or more Objects to the registry.</td>
<td></td>
</tr>
<tr>
<td>BulkResponse</td>
<td><code>unDeprecateObjects(Collection keys)</code></td>
<td></td>
</tr>
</tbody>
</table>
|         | Undeprecates one or more previously deprecated objects.
ASSOCIATION
static final String ASSOCIATION

See Also:
Constant Field Values

AUDITABLE_EVENT
static final String AUDITABLE_EVENT

See Also:
Constant Field Values

CLASSIFICATION
static final String CLASSIFICATION

See Also:
Constant Field Values

CLASSIFICATION_SCHEME
static final String CLASSIFICATION_SCHEME

See Also:
Constant Field Values

---

**CONCEPT**

static final `String` `CONCEPT`

See Also:

[Constant Field Values](#)

---

**EMAIL_ADDRESS**

static final `String` `EMAIL_ADDRESS`

See Also:

[Constant Field Values](#)

---

**EXTERNAL_IDENTIFIER**

static final `String` `EXTERNAL_IDENTIFIER`

See Also:

[Constant Field Values](#)

---

**EXTERNAL_LINK**

static final `String` `EXTERNAL_LINK`

See Also:
Constant Field Values

EXTRINSIC_OBJECT

static final String EXTRINSIC_OBJECT

See Also:

Constant Field Values

INTERNATIONAL_STRING

static final String INTERNATIONAL_STRING

See Also:

Constant Field Values

KEY

static final String KEY

See Also:

Constant Field Values

LOCALIZED_STRING

static final String LOCALIZED_STRING

See Also:
Constant Field Values

ORGANIZATION

static final String ORGANIZATION

See Also:
Constant Field Values

PERSON_NAME

static final String PERSON_NAME

See Also:
Constant Field Values

POSTAL_ADDRESS

static final String POSTAL_ADDRESS

See Also:
Constant Field Values

REGISTRY_ENTRY

static final String REGISTRY_ENTRY

See Also:
Constant Field Values

REGISTRY_PACKAGE

static final String REGISTRY_PACKAGE

See Also:
Constant Field Values

SERVICE

static final String SERVICE

See Also:
Constant Field Values

SERVICE_BINDING

static final String SERVICE_BINDING

See Also:
Constant Field Values

SLOT

static final String SLOT

See Also:
**Constant Field Values**

---

**SPECIFICATION_LINK**

static final `String` `SPECIFICATION_LINK`

See Also: [Constant Field Values](#)

---

**TELEPHONE_NUMBER**

static final `String` `TELEPHONE_NUMBER`

See Also: [Constant Field Values](#)

---

**USER**

static final `String` `USER`

See Also: [Constant Field Values](#)

---

**VERSIONABLE**

static final `String` `VERSIONABLE`

See Also:
### Method Detail

**public Object createObject(String interfaceName) throws JAXRException, InvalidRequestException, UnsupportedCapabilityException**

**Organization**

Organization org = (Organization)
lifeCycleMgr.createObject(LifeCycleManager.ORGANIZATION);

0

<table>
<thead>
<tr>
<th>interfaceName</th>
<th>javax.xml.registry.infomodel</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>Object</td>
</tr>
<tr>
<td>Throws</td>
<td>JAXRException: JAXR</td>
</tr>
<tr>
<td></td>
<td>InvalidRequestException: javax.xml.registry.infomodel</td>
</tr>
<tr>
<td></td>
<td>UnsupportedCapabilityException: JAXR</td>
</tr>
</tbody>
</table>

**createObject**

Object createObject(String interfaceName)

Throws JAXRException, InvalidRequestException, UnsupportedCapabilityException

Creates instances of information model interfaces (factory method).  
To create an Organization, use this method as follows:

Organization org = (Organization)
lifeCycleMgr.createObject(LifeCycleManager.ORGANIZATION);

**Capability Level: 0**
Parameters:
- `interfaceName` - the unqualified name of an interface in the `javax.xml.registry.infomodel` package

Returns:
an Object that can then be cast to an instance of the interface

Throws:
- `JAXRException` - if the JAXR provider encounters an internal error
- `InvalidRequestException` - if the interface is not an interface in the `javax.xml.registry.infomodel` package
- `UnsupportedCapabilityException` - if the client attempts to create an instance of an infomodel interface that is not supported by the capability level of the JAXR provider

```java
public Association createAssociation(RegistryObject targetObject, Concept associationType) throws JAXRException

Association sourceObject  null Association RegistryObject

UDDI  Organization  Association

0

targetObject  Association  RegistryObject
associationType  Association
return  Association
Throws  JAXRException:  JAXR
```

`createAssociation`

```java
Association createAssociation(RegistryObject targetObject, Concept associationType) throws JAXRException

Creates an Association instance using the specified parameters. The`
sourceObject is left null and will be set when the Association is added to a RegistryObject.

Note that for a UDDI provider an Association may only be created between Organizations.

**Capability Level: 0**

**Parameters:**
- targetObject - the target RegistryObject for the association
- associationType - the association type for the Association

**Returns:**
the Association instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

```java
public Classification createClassification(ClassificationScheme scheme, String name, String value) throws JAXRException
Classification createClassification(ClassificationScheme scheme, String name, String value)
```

<table>
<thead>
<tr>
<th>parameter</th>
<th>type</th>
</tr>
</thead>
<tbody>
<tr>
<td>scheme</td>
<td>ClassificationScheme</td>
</tr>
<tr>
<td>name</td>
<td>String</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
</tr>
<tr>
<td>return</td>
<td>Classification</td>
</tr>
</tbody>
</table>

**Throws**
- JAXRException: JAXR

**createClassification**

```java
classification createClassification(ClassificationScheme scheme, String name, String value)
```
throws JAXRException

Creates a Classification instance for an external Classification using the specified String name and String value that identify a taxonomy element within the specified ClassificationScheme.

**Capability Level: 0**

**Parameters:**
- `scheme` - the ClassificationScheme to be used
- `name` - the name of the taxonomy element (a String)
- `value` - the value of the taxonomy element

**Returns:**
the Classification instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

```java
public Classification createClassification(ClassificationScheme scheme, InternationalString name, String value) throws JAXRException {
    Classification return Classification
    \[\text{Scheme} \text{Classification} \text{InternationalString} \text{String} \text{ClassificationScheme}\]
    \[\text{0}\]
    \[\text{scheme} \text{Classification} \text{Scheme}\]
    \[\text{name} \text{InternationalString}\]
    \[\text{value} \text{InternationalString}\]
    \[\text{return} \text{Classification}\]
    \[\text{Throws} \text{JAXRException: JAXR}\]
```

createClassification
**Classification** createClassification(**ClassificationScheme** scheme, **InternationalString** name, **String** value)

throws **JAXRException**

Creates a Classification instance for an external Classification using the specified InternationalString name and String value that identify a taxonomy element within the specified ClassificationScheme.

**Capability Level:** 0

**Parameters:**
- `scheme` - the ClassificationScheme to be used
- `name` - the name of the taxonomy element (an InternationalString)
- `value` - the value of the taxonomy element

**Returns:**
- the Classification instance created

**Throws:**
- **JAXRException** - if the JAXR provider encounters an internal error

---

**public Classification** createClassification(**Concept** concept)

throws **JAXRException**, **InvalidRequestException**

Classification Classification Concept ClassificationScheme

0

```
  concept               Concept
  return               Classification
  Throws                JAXRException: JAXR
  Throws                InvalidRequestException: Concept ClassificationScheme
```

**createClassification**

**Classification** createClassification(**Concept** concept)
throws JAXRException, InvalidRequestException

Creates a Classification instance for an internal Classification using
the specified Concept that identifies a taxonomy element within an
internal ClassificationScheme.

**Capability Level: 0**

**Parameters:**
- concept - the Concept that identifies the taxonomy element

**Returns:**
- the Classification instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error
- InvalidRequestException - if the Concept is not under a ClassificationScheme

```java
public ClassificationScheme createClassificationScheme(String name, String description) throws JAXRException, InvalidRequestException

String  ClassificationScheme

0

name ClassificationScheme String
description ClassificationScheme String
return  ClassificationScheme
Throws JAXRException: JAXR
```

createClassificationScheme

```java
ClassificationScheme createClassificationScheme(String name, String description)
```
Creates a ClassificationScheme given the specified String parameters.

**Capability Level: 0**

**Parameters:**
- name - the name of the ClassificationScheme (a String)
- description - a description of the ClassificationScheme (a String)

**Returns:**
the ClassificationScheme instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error
- InvalidRequestException

```java
public ClassificationScheme createClassificationScheme(InternationalString name, InternationalString description) throws JAXRException, InvalidRequestException
  InternationalString ClassificationScheme

0

    name               ClassificationScheme  InternationalString
    description  ClassificationScheme  InternationalString
    return          ClassificationScheme
    Throws          JAXRException: JAXR
```

createClassificationScheme

```java
ClassificationScheme createClassificationScheme(InternationalString description) throws JAXRException
```
InvalidRequestException

Creates a ClassificationScheme given the specified InternationalString parameters.

**Capability Level: 0**

**Parameters:**
- name - the name of the ClassificationScheme (an InternationalString)
- description - a description of the ClassificationScheme (an InternationalString)

**Returns:**
- the ClassificationScheme instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error
- InvalidRequestException

```java
public ClassificationScheme createClassificationScheme(Concept concept) throws JAXRException, InvalidRequestException
ClassificationScheme Concept Concept ClassificationScheme

Concept ClassificationScheme

BusinessQueryManager findConcept Concept ClassificationScheme Concept ClassificationScheme

UDDI tModel tModel Concept tModel ClassificationScheme UDDI

0
```
createClassificationScheme

ClassificationScheme createClassificationScheme(Concept concept) throws JAXRException

Creates a ClassificationScheme from a Concept that has no ClassificationScheme or parent Concept.

This method is a special-case method to do a type-safe conversion from Concept to ClassificationScheme.

This method is provided to allow for Concepts returned by the BusinessQueryManager findConcepts call to be safely cast to ClassificationScheme. It is up to the programmer to make sure that the Concept is indeed semantically a ClassificationScheme.

This method is necessary because in the UDDI specification a tModel may serve multiple purposes, and there is no way to know when a tModel maps to a Concept and when it maps to a ClassificationScheme. The UDDI specification leaves the determination to the programmer, and consequently so does this method.

Capability Level: 0

Parameters:
  concept - the Concept to be used

Returns:
  the ClassificationScheme instance created

Throws:
  JAXRException - if the JAXR provider encounters an internal error
InvalidRequestException - if the Concept has a parent Concept or is under a ClassificationScheme

public Concept createConcept(RegistryObject parent, String name, String value) throws JAXRException
Concept name String

0

parent ClassificationScheme Concept
name Concept String
value Concept
return Concept
Throws JAXRException: JAXR

createConcept

Concept createConcept(RegistryObject parent, String name, String value)
throws JAXRException

Creates a Concept instance using the specified parameters, where the name is a String.

Capability Level: 0

Parameters:
  parent - a reference either to a parent ClassificationScheme or to a Concept
  name - the name of the concept (a String)
  value - the value of the concept

Returns:
  the Concept instance created

Throws:
  JAXRException - if the JAXR provider encounters an internal
public Concept createConcept(RegistryObject parent, InternationalString name, String value) throws JAXRException

Concept name InternationalString

0

parent ClassificationScheme Concept
name Concept InternationalString
value Concept
return Concept
Throws JAXRException: JAXR

createConcept

Concept createConcept(RegistryObject parent, InternationalString name, String value) throws JAXRException

Creates a Concept instance using the specified parameters, where the name is an InternationalString.

Capability Level: 0

Parameters:
parent - a reference either to a parent ClassificationScheme or to a Concept
name - the name of the concept (an InternationalString)
value - the value of the concept

Returns:
the Concept instance created

Throws:
JAXRException - if the JAXR provider encounters an internal
public EmailAddress createEmailAddress(String address) throws JAXRException
EmailAddress address

0

address
return EmailAddress
Throws JAXRException: JAXR

createEmailAddress

EmailAddress createEmailAddress(String address)
throws JAXRException

Creates an EmailAddress instance using an address as the parameter.

Capability Level: 0

Parameters:
address - the email address

Returns:
the EmailAddress instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

public EmailAddress createEmailAddress(String address, String type) throws JAXRException
EmailAddress address type
createEmailAddress

EmailAddress createEmailAddress(String address, String type) throws JAXRException

Creates an EmailAddress instance using both an address and a type as parameters.

Capability Level: 0

Parameters:
    address - the email address
    type  - the type of the address

Returns:
    the EmailAddress instance created

Throws:
    JAXRException - if the JAXR provider encounters an internal error

public ExternalIdentifier createExternalIdentifier(ClassificationScheme identificationScheme, String name, String value) throws JAXRException
    ExternalIdentifier name String
createExternalIdentifier

ExternalIdentifier createExternalIdentifier(ClassificationScheme identificationScheme, String name, String value) throws JAXRException

Creates an ExternalIdentifier instance using the specified parameters, where the name is a String.

Capability Level: 0

Parameters:
- identificationScheme - the ClassificationScheme used
- name - the name of the external identifier (a String)
- value - the value of the external identifier

Returns:
the ExternalIdentifier instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

---

public ExternalIdentifier createExternalIdentifier(ClassificationScheme identificationScheme, InternationalString name, String value) throws JAXRException

ExternalIdentifier name InternationalString

0

identificationScheme ClassificationScheme
createExternalIdentifier

```java
ExternalIdentifier createExternalIdentifier(ClassificationScheme ide
InternationalString name
String value)
throws JAXRException
```

Creates an ExternalIdentifier instance using the specified parameters, where the name is an InternationalString.

**Capability Level: 0**

**Parameters:**
- `identificationScheme` - the ClassificationScheme used
- `name` - the name of the external identifier (an InternationalString)
- `value` - the value of the external identifier

**Returns:**
- the ExternalIdentifier instance created

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

---

```java
public ExternalLink createExternalLink(String externalURI,
String description) throws JAXRException
```

**0**

- `externalURI` - URI
- `description` - String
- `return` - ExternalLink
createExternalLink

createExternalLink(String externalURI, String description) throws JAXRException

Creates an ExternalLink instance using the specified parameters, where the description is a String.

Capability Level: 0

Parameters:
- externalURI - the external URI
- description - a description of the link (a String)

Returns:
- the ExternalLink instance created

Throws:
- JAXRException - if the JAXR provider encounters an internal error

public ExternalLink createExternalLink(String externalURI, InternationalString description) throws JAXRException

Returns: ExternalLink description InternationalString

0

externalURI URI
description InternationalString
return ExternalLink
Throws JAXRException: JAXR

createExternalLink
**ExternalLink** `createExternalLink(String externalURI, InternationalString description) throws JAXRException`

Creates an ExternalLink instance using the specified parameters, where the description is an InternationalString.

**Capability Level: 0**

**Parameters:**
- `externalURI` - the external URI
- `description` - a description of the link (an InternationalString)

**Returns:**
- the ExternalLink instance created

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

---

```java
public ExtrinsicObject createExtrinsicObject(DataHandler repositoryItem) throws JAXRException
ExtrinsicObject
```

1

```java
repositoryItem DataHandler null
return ExtrinsicObject
Throws JAXRException JAXR
```

**createExtrinsicObject**

**ExtrinsicObject createExtrinsicObject(DataHandler repositoryItem) throws JAXRException**

Creates an ExtrinsicObject instance using the specified parameters.

**Capability Level: 1**
Parameters:
repositoryItem - the DataHandler for the repository item. Must not be null.

Returns:
the ExtrinsicObject instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

public InternationalString createInternationalString() throws JAXRException
InternationalString

return InternationalString

Throws JAXRException: JAXR

createInternationalString

InternationalString createInternationalString() throws JAXRException

Creates an empty InternationalString instance.

Capability Level: 0

Returns:
the InternationalString instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

public InternationalString createInternationalString(String s) throws JAXRException
createInternationalString

`createInternationalString(String s)` throws `JAXRException`

Creates an InternationalString instance using a String parameter and the default Locale.

**Capability Level**: 0

**Parameters:**

- `s` - the String from which to create the InternationalString

**Returns:**

the InternationalString instance created

**Throws:**

- `JAXRException` - if the JAXR provider encounters an internal error
createInternationalString

```java
InternationalString createInternationalString(Locale l, String s) throws JAXRException
```

Creates an InternationalString instance using the specified Locale and String parameters.

**Capability Level:** 0

**Parameters:**
- `l` - the Locale in which to create the InternationalString
- `s` - the String from which to create the InternationalString

**Returns:**
- the InternationalString instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

---

```java
public Key createKey(String id) throws JAXRException
```

**Key**

```
0
```

**Parameters:**
- `id` - Key ID

**Returns:**
- Key

**Throws:**
- JAXRException - JAXR

---

createKey

```java
Key createKey(String id) throws JAXRException
```
Creates a Key instance from an ID.

**Capability Level: 0**

**Parameters:**
- `id` - the ID string from which to create the Key

**Returns:**
- the Key instance created

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public LocalizedString createLocalizedString(java.util.Locale l, String s) throws JAXRException
Locale String LocalizedString

0

/ LocalizedString Locale
s LocalizedString String
return
Throws JAXRException: JAXR
```

**createLocalizedString**

```java
LocalizedString createLocalizedString(Locale l, String s) throws JAXRException
```

Creates a LocalizedString instance using the specified Locale and String parameters.

**Capability Level: 0**
public LocalizedString createLocalizedString(java.util.Locale l, String s, String charSetName) throws JAXRException

LocaleString LocalizedString

createLocalizedString

LocalizedString createLocalizedString(Locale l, String s, String charSetName) throws JAXRException

Creates a LocalizedString instance using the specified Locale, String, and character set parameters.

Capability Level: 0

Parameters:

1 - the Locale in which to create the LocalizedString
s - the String from which to create the LocalizedString
charSetName - the name of the character set to use

Returns:
the LocalizedString instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

public Organization createOrganization(String name) throws JAXRException
Organization name String
0

name Organization
return Organization

Throws JAXRException: JAXR

createOrganization

Organization createOrganization(String name) throws JAXRException

Creates an Organization instance using the specified name, where the name is a String.

Capability Level: 0

Parameters:
  name - the name of the Organization

Returns:
  the Organization instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error
public **Organization** createOrganization(**InternationalString** name) throws **JAXRException**

Organization name **InternationalString**

0

- **name**
- **Organization**
- **return**
- **Organization**
- **Throws**
  - **JAXRException**: JAXR

**createOrganization**

**Organization** createOrganization(**InternationalString** name) throws **JAXRException**

Creates an Organization instance using the specified name, where the name is an **InternationalString**.

**Capability Level**: 0

**Parameters:**
- **name** - the name of the Organization

**Returns:**
- the Organization instance created

**Throws:**
- **JAXRException** - if the JAXR provider encounters an internal error

---

public **PersonName** createPersonName(String firstName, String middleName, String lastName) throws **JAXRException**

**PersonName**

1
createPersonName

```java
PersonName createPersonName(String firstName,
String middleName,
String lastName)
throws JAXRException
```

Creates a PersonName instance using the specified first, middle, and last names.

**Capability Level: 1**

**Parameters:**
- `firstName` - the person's first name
- `middleName` - the person's middle name
- `lastName` - the person's last name

**Returns:**
- the PersonName instance created

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public PersonName createPersonName(String fullName)
throws JAXRException
PersonName
```

```java
fullName
```

```java
return PersonName
```
createPersonName

createPersonName(String fullName) throws JAXRException

Creates a PersonName instance using the specified full name.

Capability Level: 0

Parameters:
  fullName - the person's full name

Returns:
  the PersonName instance created

 Throws:
  JAXRException - if the JAXR provider encounters an internal error

---

public PostalAddress createPostalAddress(String streetNumber, String street, String city, String stateOrProvince, String country, String postalCode, String type) throws JAXRException

PostalAddress

0

streetNumber
street
city
stateOrProvince
country
postalCode
US ZIP
type
return
PostalAddress
Throws

JAXRException: JAXR

createPostalAddress

PostalAddress createPostalAddress(String streetNumber,
    String street,
    String city,
    String stateOrProvince,
    String country,
    String postalCode,
    String type)
throws JAXRException

Creates a PostalAddress instance using the specified parameters.

Capability Level: 0

Parameters:
- streetNumber - the street number
- street - the street name
- city - the city name
- stateOrProvince - the state or province name
- country - the country name
- postalCode - the postal code (such as a US ZIP code)
- type - the type of the address

Returns:
the PostalAddress instance created

Throws:
  JAXRException - if the JAXR provider encounters an internal error

---

public RegistryPackage createRegistryPackage(String name) throws JAXRException
RegistryPackage name String

1

name String
createRegistryPackage

RegistryPackage createRegistryPackage(String name) throws JAXRException

Creates a RegistryPackage instance using the specified name, where the name is a String.

Capability Level: 1

Parameters:
name - the name of the registry package (a String)

Returns:
the RegistryPackage instance created

Throws:
JAXRException - if the JAXR provider encounters an internal error

public RegistryPackage createRegistryPackage(InternationalString name) throws JAXRException

RegistryPackage name InternationalString

1

name InternationalString
return RegistryPackage
Throws JAXRException: JAXR

createRegistryPackage

RegistryPackage createRegistryPackage(InternationalString name)
throws JAXRException

Creates a RegistryPackage instance using the specified name, where the name is an InternationalString.

**Capability Level: 1**

**Parameters:**
- **name** - the name of the registry package (an InternationalString)

**Returns:**
- the RegistryPackage instance created

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

```java
public Service createService(String name) throws JAXRException
```

Service name String

0

```
name Service String return Service
Throws JAXRException: JAXR
```

createService

```java
Service createService(String name)
```

Throws JAXRException

Creates a Service instance using the specified name, where the name is a String.

**Capability Level: 0**

**Parameters:**
- **name** - the name of the Service (a String)
Returns:
the Service instance created

Throws:

- **JAXRException** - if the JAXR provider encounters an internal error

```java
public Service createService(InternationalString name) throws JAXRException
Service name InternationalString

0

name Service InternationalString
return Service
Throws JAXRException: JAXR
```

### createService

**Service createService(InternationalString name)**

throws **JAXRException**

Creates a Service instance using the specified name, where the name is an InternationalString.

**Capability Level: 0**

**Parameters:**

- `name` - the name of the Service (an InternationalString)

**Returns:**

the Service instance created

**Throws:**

- **JAXRException** - if the JAXR provider encounters an internal error

```java
public ServiceBinding createServiceBinding() throws
```
**JAXRException**

ServiceBinding

0

```
return ServiceBinding
Throws JAXRException: JAXR
```

**createServiceBinding**

```java
createServiceBinding() throws JAXRException
```

Creates an empty ServiceBinding instance.

**Capability Level:** 0

**Returns:**
the ServiceBinding instance created

**Throws:**

- JAXRException - if the JAXR provider encounters an internal error

```
public Slot createSlot(String name, String value, String slotType) throws JAXRException
Slot value String
```

```
name Slot
value String
slotType Slot
return Slot
Throws JAXRException: JAXR
```
createSlot

```java
Slot createSlot(String name,
String value,
String slotType)
throws JAXRException
```

Creates a Slot instance using the specified parameters, where the value is a String.

**Capability Level: 0**

**Parameters:**
- `name` - the name of the Slot
- `value` - the value (a String)
- `slotType` - the slot type

**Returns:**
the Slot instance created

**Throws:**
- `JAXRException` - if the JAXR provider encounters an internal error

```java
public Slot createSlot(String name, java.util.Collection<E> values, String slotType) throws JAXRException
Slot value String String Collection
```

<table>
<thead>
<tr>
<th>name</th>
<th>Slot</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String Collection</td>
</tr>
<tr>
<td>slotType</td>
<td>Slot</td>
</tr>
<tr>
<td>return</td>
<td>Slot</td>
</tr>
</tbody>
</table>

**Throws**

- `JAXRException`: JAXR
Slot createSlot(String name, Collection values, String slotType) throws JAXRException

Creates a Slot instance using the specified parameters, where the value is a Collection of Strings.

Capability Level: 0

Parameters:
- name - the name of the Slot
- value - the value (a Collection of Strings)
- slotType - the slot type

Returns:
- the Slot instance created

Throws:
- JAXRException - if the JAXR provider encounters an internal error

public SpecificationLink createSpecificationLink() throws JAXRException

SpecificationLink

0

return SpecificationLink

Throws JAXRException: JAXR

createSpecificationLink

SpecificationLink createSpecificationLink() throws JAXRException

Creates an empty SpecificationLink instance.

Capability Level: 0
public **TelephoneNumber** createTelephoneNumber() throws **JAXRException**

**TelephoneNumber**

0

    return **TelephoneNumber**

Throws **JAXRException**: JAXR

---

**createTelephoneNumber**

**TelephoneNumber** createTelephoneNumber() throws **JAXRException**

Creates an empty TelephoneNumber instance.

**Capability Level: 0**

**Returns:**

the TelephoneNumber instance created

**Throws:**

**JAXRException** - if the JAXR provider encounters an internal error

---

public **User** createUser() throws **JAXRException**

**User**

0
createElement

public BulkResponse saveObjects(java.util.Collection<E> objects) throws JAXRException
Object RegistryObject

0

objects RegistryObject Collection
return BulkResponse Collection SaveException
Throws JAXRException: JAXR

saveObjects

BulkResponse saveObjects(Collection objects) throws JAXRException
Saves one or more Objects to the registry. An object may be a RegistryObject subclass instance.

If an object is not in the registry, it is created in the registry. If it already exists in the registry and has been modified, then its state is updated (replaced) in the registry.

**Capability Level: 0**

**Parameters:**
- objects - a potentially heterogeneous Collection of RegistryObject instances

**Returns:**
- a BulkResponse containing the Collection of keys for those objects that were saved successfully and any SaveException that was encountered in case of partial commit

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

```java
public BulkResponse deprecateObjects(java.util.Collection<E> keys) throws JAXRException
"soon to be deleted"JAXR AssociationsClassifications  ExternalLinks API  JAXR  JAXRException java.lang.IllegalStateException

1

keys Collection
return BulkResponse Collection JAXRException
Throws JAXRException: JAXR

deprecateObjects
```
BulkResponse deprecateObjects(Collection keys)
throws JAXRException

Deprecates one or more previously submitted objects. Deprecation marks an object as "soon to be deleted". Once an object is deprecated, the JAXR provider must not allow any new references (e.g. new Associations, Classifications and ExternalLinks) to that object to be submitted. If a client makes an API call that results in a new reference to a deprecated object, the JAXR provider must throw a java.lang.IllegalStateException within a JAXRException. However, existing references to a deprecated object continue to function normally.

Capability Level: 1

Parameters:
keys - a Collection of keys for the objects to be deprecated

Returns:
a BulkResponse containing the Collection of keys for those objects that were deprecated successfully and any JAXRException that was encountered in case of partial commit

Throws:
JAXRException - if the JAXR provider encounters an internal error

public BulkResponse unDeprecateObjects(java.util.Collection<E> keys) throws JAXRException
JAXR Associations
Classifications ExternalLinks

keys Collection
return BulkResponse Collection JAXRException
Throws JAXRException: JAXR
unDeprecateObjects

BulkResponse unDeprecateObjects(Collection keys) throws JAXRException

Undeprecates one or more previously deprecated objects. If an object was not previously deprecated, it is not an error, and no exception is thrown. Once an object is undeprecated, the JAXR provider must again allow new references (e.g. new Associations, Classifications and ExternalLinks) to that object to be submitted.

Capability Level: 1

Parameters:
- keys - a Collection of keys for the objects to be undeprecated

Returns:
- a BulkResponse containing the Collection of keys for those objects that were deprecated successfully and any JAXRException that was encountered in case of partial commit

Throws:
- JAXRException - if the JAXR provider encounters an internal error

public BulkResponse deleteObjects(java.util.Collection<E> keys) throws JAXRException

1

keys Collection
return BulkResponse Collection DeleteException
Throws JAXRException: JAXR

deleteObjects

BulkResponse deleteObjects(Collection keys)
Deletes one or more previously submitted objects from the registry using the object keys.

**Capability Level: 1**

**Parameters:**
- keys - a Collection of keys for the objects to be deleted

**Returns:**
- a BulkResponse containing the Collection of keys for those objects that were deleted successfully and any DeleteException that was encountered in case of partial commit

**Throws:**
- JAXRException - if the JAXR provider encounters an internal error

```java
public BulkResponse deleteObjects(java.util.Collection<E> keys, String objectType) throws JAXRException
```

**deleteObjects**

BulkResponse deleteObjects(Collection keys, String objectType) throws JAXRException

Deletes one or more previously submitted objects from the registry using the object keys and a specified objectType attribute.
public RegistryService getRegistryService() throws JAXRException

RegistryService

0

return RegistryService

Throws JAXRException: JAXR

associates <javax.xml.registry.RegistryService>

getRegistryService

RegistryService getRegistryService() throws JAXRException

Returns the parent RegistryService that created this object.

Capability Level: 0

Returns:

the parent RegistryService

throws JAXRException

Throws:
JAXRException - if the JAXR provider encounters an internal error

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.model Class ListDataModel

java.lang.Object
   ↓ javax.faces.model.DataModel
      ↓ javax.faces.model.ListDataModel

public class ListDataModel
extends DataModel

Extends: DataModel

ListDataModel is a convenience implementation of DataModel that wraps an List of Java objects.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ListDataModel()</td>
<td>Construct a new ListDataModel with no specified wrapped data.</td>
</tr>
<tr>
<td>ListDataModel(List list)</td>
<td>Construct a new ListDataModel wrapping the specified list.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getRowCount()</td>
<td>If there is wrappedData available, return the length of the list.</td>
</tr>
<tr>
<td>Object getRowData()</td>
<td>If row data is available, return the array element at the index specified by rowIndex.</td>
</tr>
<tr>
<td>getRowIndex()</td>
<td></td>
</tr>
</tbody>
</table>
### Methods

**int**

Return the zero-relative index of the currently selected row.

**Object**

`getWrappedData()`

Return the object representing the data wrapped by this `DataModel`, if any.

**boolean**

`isRowAvailable()`

Return `true` if there is `wrappedData` available, and the current value of `rowIndex` is greater than or equal to zero, and less than the size of the list.

**void**

`setRowIndex(int rowIndex)`

Set the zero-relative index of the currently selected row, or `-1` to indicate that we are not positioned on a row.

`setWrappedData(Object data)`

Set the object representing the data collection wrapped by this `DataModel`.

---

**Methods inherited from class `javax.faces.model.DataModel`**

- `addDataModelListener`, `getDataModelListeners`, `removeDataModelListener`

**Methods inherited from class `java.lang.Object`**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

---

### Constructor Detail

**public ListDataModel()**

`ListDataModel`
public ListDataModel()

Construct a new ListDataModel with no specified wrapped data.

public ListDataModel(java.util.List<E> list)

ListDataModel

list

ListDataModel

public ListDataModel(List list)

Construct a new ListDataModel wrapping the specified list.

Parameters:
list - List to be wrapped (if any)

Method Detail

public boolean isRowAvailable()

wrappedData  rowIndex  0  true  false

Throws  FacesException:

isRowAvailable

public boolean isRowAvailable()

Return true if there is wrappedData available, and the current value of rowIndex is greater than or equal to zero, and less than the size of the list. Otherwise, return false.
public int getRowCount()

wrappedData     wrappedData -1

Throws            FacesException:

getRowCount

public int getRowCount()

If there is wrappedData available, return the length of the list. If no wrappedData is available, return -1.

Specified by:
    getRowCount in class DataModel
Throws:
    FacesException - if an error occurs getting the row count

public Object getRowData()

rowIndex         null

Throws            FacesException:
Throws            IllegalArgumentException:

getRowData

public Object getRowData()
If row data is available, return the array element at the index specified by rowIndex. If no wrapped data is available, return null.

**Specified by:**
getRowData in class DataModel

**Throws:**
FacesException - if an error occurs getting the row data
IllegalArgumentException - if now row data is available at the currently specified row index

```java
public int getRowIndex()

Throws  FacesException: NullPointerException
```

### getRowIndex

**Description copied from class:** DataModel

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no wrappedData is available, return -1.

**Specified by:**
getRowIndex in class DataModel

**Throws:**
FacesException - if an error occurs getting the row index

```java
public void setRowIndex(int rowIndex)

Throws  FacesException: NullPointerException
Throws  IllegalArgumentException: NullPointerException rowIndex
```

### setRowIndex

```java
setRowIndex
```
public void setRowIndex(int rowIndex)

**Description copied from class:** [DataModel](#)

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the isRowAvailable() method to detect whether row data will be available for use by the getRowData() method.

If there is no wrappedData available when this method is called, the specified rowIndex is stored (and may be retrieved by a subsequent call to getRowData()), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a DataModelEvent will be sent to the rowSelected() method of all registered DataModelListeners.

**Specified by:**
setRowIndex in class [DataModel](#)

**Parameters:**
- rowIndex - The new zero-relative index (must be non-negative)

**Throws:**
- [FacesException](#) - if an error occurs setting the row index
- [IllegalArgumentException](#) - if rowIndex is less than -1

---

public Object getWrappedData()

**getWrappedData**

public Object getWrappedData()

**Description copied from class:** [DataModel](#)

Return the object representing the data wrapped by this DataModel, if any.
specified by:

**getWrappedData** in class **DataModel**

---

**public void setWrappedData(Object data)**

Throws

**ClassCastException: data**

**List**

---

**setWrappedData**

**public void setWrappedData(Object data)**

Description copied from class: **DataModel**

Set the object representing the data collection wrapped by this **DataModel**. If the specified data is **null**, detach this **DataModel** from any previously wrapped data collection instead.

If data is non-null, the currently selected row index must be set to zero, and a **DataModelEvent** must be sent to the **rowSelected()** method of all registered **DataModelListener**s indicating that this row is now selected.

**Specified by:**

**setWrappedData** in class **DataModel**

**Parameters:**

data - Data collection to be wrapped, or **null** to detach from any previous data collection

**Throws:**

**ClassCastException** - if data is non-null and is not a **List**
PS:
javax.el Class ListELResolver

java.lang.Object
   ↓ javax.el.ELResolver
      ↓ javax.el.ListELResolver

public class ListELResolver extends ELResolver

Extend: ELResolver

java.util.List

java.util.List base

#isReadOnly true#setValue PropertyNotWritableException

ELResolver CompositeELResolver ELResolver

Javadoc

since JSP 2.1
See javax.el.CompositeELResolver, javax.el.ELResolver
also java.util.List

Defines property resolution behavior on instances of List.

This resolver handles base objects of type java.util.List. It accepts any object as a property and coerces that object into an integer index into the list. The resulting value is the value in the list at that index.

This resolver can be constructed in read-only mode, which means that isReadOnly will always return true and setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always throw PropertyNotWritableException.

ELResolvers are combined together using CompositeELResolver, to
define rich semantics for evaluating an expression. See the javadocs for
ELResolver for details.

Since:
   JSP 2.1
See Also:
   CompositeELResolver, ELResolver, List

---

### Field Summary

**Fields inherited from class javax.el.ELResolver**

| RESOLVABLE_AT_DESIGN_TIME, TYPE |

### Constructor Summary

**ListELResolver()**

Creates a new read/write ListELResolver.

**ListELResolver(boolean isReadOnly)**

Creates a new ListELResolver whose read-only status is
determined by the given parameter.

### Method Summary

<table>
<thead>
<tr>
<th>Class&lt;?&gt;</th>
<th>getCommonPropertyType(ELContext context, Object base)</th>
</tr>
</thead>
</table>
|          | If the base object is a list, returns the most
general type that this resolver accepts for the
property argument. |

<table>
<thead>
<tr>
<th>Iterator&lt;FeatureDescriptor&gt;</th>
<th>getFeatureDescriptors(ELContext context, Object base)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Always returns null, since there is no reason to iterate through set set of all integers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class&lt;?&gt;</th>
<th>getType(ELContext context, Object base, Object property)</th>
</tr>
</thead>
</table>
|          | If the base object is a list, returns the most
general acceptable type for a value in this list. |
### Constructor Detail

**public ListELResolver()**

/     ListELResolver

**ListELResolver**

**public ListELResolver()**

Creates a new read/write ListELResolver.

**public ListELResolver(boolean isReadOnly)**

ListELResolver

    isReadOnly true false
ListELResolver

public ListELResolver(boolean isReadOnly)

Creates a new ListELResolver whose read-only status is determined by the given parameter.

Parameters:
  isReadOnly - true if this resolver cannot modify lists; false otherwise.

Method Detail

getType

public Class<T> getType(ELContext context, Object base, Object property)

base             List   ELContext   propertyResolved   true
true             base               List               base

context
base
property
return            ELContext   propertyResolved   true

Throws            PropertyNotFoundException:
Throws            NullPointerException: context
Throws            ELException: cause

getType

public Class<?> getType(ELContext context,
If the base object is a list, returns the most general acceptable type for a value in this list.

If the base is a List, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Assuming the base is a List, this method will always return Object.class. This is because lists accept any object as an element.

Specified by:

g getType in class ELResolver

Parameters:

context - The context of this evaluation.
base - The list to analyze. Only bases of type List are handled by this resolver.
property - The index of the element in the list to return the acceptable type for. Will be coerced into an integer, but otherwise ignored by this resolver.

Returns:

If the propertyResolved property of ELContext was set to true, then the most general acceptable type; otherwise undefined.

Throws:

PropertyNotFoundException - if the given index is out of bounds for this list.
NullPointerException - if context is null 
ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

public Object getValue(ELContext context, Object base, Object property)
IllegalArgumentException null

base List ELContext propertyResolved true
true

class

context
base
property
return ELContext propertyResolved true null

Throws IllegalArgumentException:
Throws NullPointerException: context null
Throws ELException: cause

getValue

public Object getValue(ELContext context,
Object base,
Object property)

If the base object is a list, returns the value at the given index. The index is specified by the property argument, and coerced into an integer. If the coercion could not be performed, an IllegalArgumentException is thrown. If the index is out of bounds, null is returned.

If the base is a List, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Specified by:
getValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - The list to be analyzed. Only bases of type List are handled by this resolver.
property - The index of the value to be returned. Will be coerced
into an integer.

**Returns:**
If the `propertyResolved` property of `ELContext` was set to `true`, then the value at the given index or `null` if the index was out of bounds. Otherwise, undefined.

**Throws:**
- `IllegalArgumentException` - if the property could not be coerced into an integer.
- `NullPointerException` - if `context` is `null`.
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public void setValue(ELContext context, Object base, Object property, Object val) throws IllegalArgumentException, PropertyNotFoundException, PropertyNotWritableException
```

```java
base List ELContext propertyResolved true
```

```java
PropertyNotWritableException
```

```java
java.util.Collections#unmodifiableList List
```

```java
PropertyNotWritableException Collection API
```

```java
List base
```

```java
context base property val
```

**Throws**
- `ClassCastException`
- `NullPointerException`
- `IllegalArgumentException`
- `PropertyNotWritableException`:
Throws `PropertyNotFoundException`:

Throws `ELException`: cause

```java
public void setValue(ELContext context, Object base, Object property, Object val)
```

If the base object is a list, attempts to set the value at the given index with the given value. The index is specified by the `property` argument, and coerced into an integer. If the coercion could not be performed, an `IllegalArgumentException` is thrown. If the index is out of bounds, a `PropertyNotFoundException` is thrown.

If the base is a `List`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller can safely assume no value was set.

If this resolver was constructed in read-only mode, this method will always throw `PropertyNotWritableException`.

If a `List` was created using `Collections.unmodifiableList(java.util.List)`, this method must throw `PropertyNotWritableException`. Unfortunately, there is no Collections API method to detect this. However, an implementation can create a prototype unmodifiable `List` and query its runtime type to see if it matches the runtime type of the base object as a workaround.

**Specified by:**

`setValue` in class `ELResolver`

**Parameters:**

- `context` - The context of this evaluation.
- `base` - The list to be modified. Only bases of type `List` are handled by this resolver.
- `property` - The index of the value to be set. Will be coerced into
an integer.
val - The value to be set at the given index.

Throws:

- ClassCastException - if the class of the specified element prevents it from being added to this list.
- NullPointerException - if context is null, or if the value is null and this List does not support null elements.
- IllegalArgumentException - if the property could not be coerced into an integer, or if some aspect of the specified element prevents it from being added to this list.
- PropertyNotWritableException - if this resolver was constructed in read-only mode, or if the set operation is not supported by the underlying list.
- PropertyNotFoundException - if the given index is out of bounds for this list.
- ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

public boolean isReadOnly(ELContext context, Object base, Object property)
base

base List ELContext propertyResolved true
true

java.util.Collections#unmodifiableList List true

Collection API

collection
base List base
property

return ELContext propertyResolved true setValue true
false
Throws  `PropertyNotFoundException`
Throws  `NullPointerException: context null`
Throws  `ELException: cause`

isReadonly

```java
public boolean isReadonly(ELContext context,
                           Object base,
                           Object property)
```

If the base object is a list, returns whether a call to

If the base is a `List`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

If this resolver was constructed in read-only mode, this method will always return `true`.

If a `List` was created using `Collections.unmodifiableList(java.util.List)`, this method must return `true`. Unfortunately, there is no Collections API method to detect this. However, an implementation can create a prototype unmodifiable `List` and query its runtime type to see if it matches the runtime type of the base object as a workaround.

Specified by:
  `isReadonly` in class `ELResolver`

Parameters:
  context - The context of this evaluation.
  base - The list to analyze. Only bases of type `List` are handled by this resolver.
  property - The index of the element in the list to return the acceptable type for. Will be coerced into an integer, but otherwise ignored by this resolver.

Returns:
If the propertyResolved property of ELContext was set to true, then true if calling the setValue method will always fail or false if it is possible that such a call may succeed; otherwise undefined.

Throws:
- PropertyNotFoundException - if the given index is out of bounds for this list.
- NullPointerException - if context is null
- ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public java.util.Iterator<E> getFeatureDescriptors(ELContext context, Object base) null
#getCommonPropertyType
c
context
d
base
List base
e
return
null
f

getFeatureDescriptors
g
public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext c
Object base
d
Always returns null, since there is no reason to iterate through set
e
set of all integers.

The getCommonPropertyType(javax.el.ELContext, java.lang.Object) method returns sufficient information about what properties this resolver accepts.

Specified by:
- getFeatureDescriptors in class ELResolver
Parameters:
  
  - context - The context of this evaluation.
  - base - The list. Only bases of type List are handled by this resolver.

Returns:
  
  null.

See Also:
  
  FeatureDescriptor

---

public Class<T> getCommonPropertyType(ELContext context, Object base)
base property null

base List Integer.class List

context
base List base
return base List null Integer.class

getCommonPropertyType

public Class<?> getCommonPropertyType(ELContext context, Object base)

If the base object is a list, returns the most general type that this resolver accepts for the property argument. Otherwise, returns null.

Assuming the base is a List, this method will always return Integer.class. This is because Lists accept integers as their index.

Specified by:
  
  getCommonPropertyType in class ELResolver

Parameters:
  
  - context - The context of this evaluation.
  - base - The list to analyze. Only bases of type List are handled by this resolver.
Returns:

null if base is not a List; otherwise Integer.class.
javax.management.j2ee  Interface ListenerRegistration

All Superinterfaces:
   Serializable

public interface ListenerRegistration

extends Serializable

Implements: java.io.Serializable

ListenerRegistration  MEJB

ListenerRegistration defines the methods which clients of the MEJB use to add and remove event listeners.

Author:
   Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addNotificationListener</strong>(ObjectName name, NotificationListener listener, NotificationFilter filter, Object handback)</td>
</tr>
<tr>
<td>Add a listener to a registered managed object.</td>
</tr>
<tr>
<td><strong>removeNotificationListener</strong>(ObjectName name, NotificationListener listener)</td>
</tr>
<tr>
<td>Remove a listener from a registered managed object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void addNotificationListener(javax.management.ObjectName name,</td>
</tr>
<tr>
<td>NotificationListener listener, NotificationFilter filter,</td>
</tr>
<tr>
<td>Object handback)</td>
</tr>
<tr>
<td>Add a listener to a registered managed object.</td>
</tr>
<tr>
<td>void removeNotificationListener(ObjectName name, NotificationListener listener)</td>
</tr>
<tr>
<td>Remove a listener from a registered managed object.</td>
</tr>
</tbody>
</table>
Add a listener to a registered managed object.

**Parameters:**

- `name`: The name of the managed object on which the listener should be added.
- `listener`: The listener object which will handle the notifications emitted by the registered managed object.
- `filter`: The filter object. If filter is null, no filtering will be performed before handling notifications.
- `handback`: The context to be sent to the listener when a notification is emitted.

**Throws:**

- `InstanceNotFoundException`: The managed object name provided does not match any of the registered managed objects.
- `RemoteException`: A communication exception occurred during
the execution of a remote method call

```java
public void removeNotificationListener(javax.management.ObjectName name, javax.management.NotificationListener listener)
    throws javax.management.InstanceNotFoundException,
            javax.management.ListenerNotFoundException,
            java.rmi.RemoteException
```

**name**

**listener**

**Throws** javax.management.InstanceNotFoundException:

**Throws** javax.management.ListenerNotFoundException:

**Throws** java.rmi.RemoteException:

---

**removeNotificationListener**

```java
void removeNotificationListener(ObjectName name, NotificationListener listener)
    throws InstanceNotFoundException,
            ListenerNotFoundException,
            RemoteException
```

Remove a listener from a registered managed object.

**Parameters:**

- **name** - The name of the managed object on which the listener should be removed.
- **listener** - The listener object which will handle the notifications emitted by the registered managed object. This method will remove all the information related to this listener.

**Throws:**

- **InstanceNotFoundException** - The managed object name provided does not match any of the registered managed objects.
- **ListenerNotFoundException** - The listener is not registered in the managed object.
RemoteException - A communication exception occurred during the execution of a remote method call

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type Lob

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Lob

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Lob Blob

@Lob @Basic(fetch=LAZY)
@Column(name="REPORT")
protected String report;

Since: Java Persistence 1.0

Specifies that a persistent property or field should be persisted as a large object to a database-supported large object type. The Lob annotation may be used in conjunction with the Basic annotation. A Lob may be either a binary or character type.

The Lob type is inferred from the type of the persistent field or property, and except for string and character-based types defaults to Blob.

Example:

@Lob @Basic(fetch=LAZY)
@Column(name="REPORT")
protected String report;

Since:
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.ejb Annotation Type Local

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Local

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean Bean value()

When used on the bean class, declares the local business interface(s) for a session bean. When used on an interface, designates that interface as a local business interface. In this case, no value() is provided.

Optional Element Summary

<table>
<thead>
<tr>
<th>Class[]</th>
<th>value</th>
</tr>
</thead>
</table>

abstract public Class<T>[] value()

value

public abstract Class[] value

Default:

{}
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.ejb  Annotation Type LocalHome

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface LocalHome

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean  Local Home  Local Home

Declares the Local Home or adapted Local Home interface for a session bean.

## Required Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>value</th>
</tr>
</thead>
</table>

## Element Detail

abstract public Class<T> value()

**value**

public abstract Class value
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel Interface LocalizedString

public interface LocalizedString

String Locale InternationalString LocalizedString
Collection LocalizedString Locale String

See also  javax.xml.registry.infomodel.InternationalString

This interface is used as a simple wrapper interface that associates a String with its Locale. The interface is needed in the InternationalString interface where a Collection of LocalizedString instances are kept. Each LocalizedString instance has a Locale and a String instance.

Author:
Farrukh S. Najmi

See Also:
InternationalString

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>DEFAULT_CHARSET_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The default name returned by getCharsetName if no other name has explicitly been set.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>String</th>
<th>getCharsetName()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the canonical name for the charset for this object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Locale</th>
<th>getLocale()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the Locale for this object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getValue()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the String value for this object.</td>
</tr>
</tbody>
</table>
### void setValue(String value)
Set the String value for the specified object.

### void setLocale(Locale locale)
Set the Locale for this object.

### void setCharsetName(String charsetName)
Set the canonical name for the charset for this object.

### Field Detail

**DEFAULT_CHARSET_NAME**

```java
static final String DEFAULT_CHARSET_NAME
```

The default name returned by getCharsetName if no other name has explicitly been set.

See Also:
- [Constant Field Values](#)

### Method Detail

```java
public String getCharsetName() throws JAXRException
```

0

<table>
<thead>
<tr>
<th>return</th>
<th>JAXRException: JAXR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td></td>
</tr>
<tr>
<td>See also</td>
<td>DEFAULT_CHARSET_NAME</td>
</tr>
</tbody>
</table>

getCharsetName
String getCharsetName() throws JAXRException

Gets the canonical name for the charset for this object. Must return the default charset when there is no charset name defined.

**Capability Level: 0**

**Returns:**
the character set name for the character set used by this object

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

**See Also:**
DEFAULT_CHARSET_NAME

```java
public java.util.Locale getLocale() throws JAXRException

Locale Locale Locale Locale

0

return Locale

Throws JAXRException: JAXR

See also getDefault()
```

getLocale

Locale getLocale() throws JAXRException

Get the Locale for this object. Must return the default Locale when no Locale has been defined.

**Capability Level: 0**

**Returns:**
the Locale used by this object
Throws:

- `JAXRException` - If the JAXR provider encounters an internal error

See Also:

- `Locale.getDefault()`

---

```java
public String getValue() throws JAXRException

String

0

return

Throws

- `JAXRException`: JAXR

---

`getValue`

```java
String getValue() throws JAXRException

Get the String value for this object.

Capability Level: 0

Returns:

- the value defined by this object

Throws:

- `JAXRException` - If the JAXR provider encounters an internal error

---

```java
public void setCharsetName(String charsetName) throws JAXRException

0
```
**setCharsetName**

```java
void setCharsetName(String charsetName)
throws JAXRException
```

Set the canonical name for the charset for this object.

**Capability Level: 0**

**Parameters:**
- `charsetName` - the character set name for the character set used by this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public void setLocale(java.util.Locale locale) throws JAXRException**

**Locale**

```java
locale
```

**Throws**
- `JAXRException` - JAXR

---

**setLocale**

```java
void setLocale(Locale locale)
throws JAXRException
```

Set the Locale for this object.
public void setValue(String value) throws JAXRException

String

0

setValue

void setValue(String value)
throws JAXRException

Set the String value for the specified object.

Capability Level: 0

Parameters:
value - the value defined by this object

Throws:
JAXRException - If the JAXR provider encounters an internal error
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface LocalTransaction

LocalTransaction

javax.resource.spi

CCI

Connection.getLocalTransaction

LocalTransaction

EIS

Connection

since 0.8

See also javax.resource.cci.Connection

The LocalTransaction defines a transaction demarcation interface for resource manager local transactions. Note that this interface is used for application level local transaction demarcation. The system contract level LocalTransaction interface (as defined in the javax.resource.spi package) is used by the container for local transaction management.

A local transaction is managed internal to a resource manager. There is no external transaction manager involved in the coordination of such transactions.

A CCI implementation can (but is not required to) implement the LocalTransaction interface. If the LocalTransaction interface is supported by a CCI implementation, then the method Connection.getLocalTransaction should return a LocalTransaction instance. A component can then use the returned LocalTransaction to demarcate a resource manager local transaction (associated with the Connection instance) on the underlying EIS instance.

Since:
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>begin()</code></td>
<td>Begins a local transaction on an EIS instance.</td>
</tr>
<tr>
<td>void <code>commit()</code></td>
<td>Commits the current local transaction and release all locks held by the underlying EIS instance.</td>
</tr>
<tr>
<td>void <code>rollback()</code></td>
<td>Rollbacks the current resource manager local transaction.</td>
</tr>
</tbody>
</table>

### Method Detail

**public void `begin()` throws `ResourceException`**

**EIS**

*ResourceException:*

**Throws:**
- EIS
- JTA

**begin**

**void `begin()` throws `ResourceException`**

Begins a local transaction on an EIS instance.

**Throws:**
- *ResourceException* - Failed to begin a local transaction.
Examples of error cases are:
- Resource adapter internal or EIS-specific error
- Connection is already participating in a local or JTA transaction

```java
public void commit() throws ResourceException
EIS

  ResourceException:
  • EIS
  Throws
  • JTA
  •
```

```java
commit
void commit()
  throws ResourceException

  Commits the current local transaction and release all locks held by the underlying EIS instance.

  Throws:
  ResourceException - Failed to commit a local transaction.
Examples of error cases are:
  • Resource adapter internal or EIS-specific error
  • Violation of integrity constraints, deadlock detection, communication failure during transaction completion, or any retry requirement
  • Connection is participating in an active JTA transaction
  • Invalid transaction context; commit operation invoked without an active transaction context
```

```java
public void rollback() throws ResourceException
```
**ResourceException:**

Throws

- EIS
- JTA

**rollback**

```java
void rollback()
throws ResourceException
```

Rollbacks the current resource manager local transaction.

**Throws:**

- ResourceException - Failed to rollback a local transaction.

Examples of error cases are:

- Resource adapter internal or EIS-specific error
- Connection is participating in an active JTA transaction
- Invalid transaction context; rollback operation invoked without an active transaction context
javax.resource.spi  Interface LocalTransaction

public interface LocalTransaction

LocalTransaction LocalTransaction
javax.resource.spi

CCI LocalTransaction CCI LocalTransaction
    Connection.getLocalTransaction LocalTransaction
LocalTransaction EIS Connection
    since 0.8
    See also  javax.resource.cci.Connection

LocalTransaction interface provides support for transactions that are managed internal to an EIS resource manager, and do not require an external transaction manager.

A resource adapter implements the javax.resource.spi.LocalTransaction interface to provide support for local transactions that are performed on the underlying resource manager.

If a resource adapter supports the LocalTransaction interface, then the application server can choose to perform local transaction optimization (uses local transaction instead of a JTA transaction for a single resource manager case).

Version: 0.5
Author: Rahul Sharma
See Also: ManagedConnection
<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>begin()</code></td>
<td>Begin a local transaction</td>
</tr>
<tr>
<td>void <code>commit()</code></td>
<td>Commit a local transaction</td>
</tr>
<tr>
<td>void <code>rollback()</code></td>
<td>Rollback a local transaction</td>
</tr>
</tbody>
</table>

**Method Detail**

**public void begin() throws** `ResourceException`

**EIS**

`ResourceException`: 

**Throws**:

- EIS
- JTA

**begin**

```java
void begin() throws ResourceException
```

Begin a local transaction

**Throws:**

- `ResourceException` - generic exception if operation fails
- `LocalTransactionException` - error condition related to local transaction management
- `ResourceAdapterInternalException` - error condition internal to resource adapter
- `EISSystemException` - EIS instance specific error condition
public void commit() throws ResourceException

EIS

ResourceException:

Throws

- EIS
- JTA

commit

void commit()
throws ResourceException

Commit a local transaction

Throws:

ResourceException - generic exception if operation fails
LocalTransactionException - error condition related to local transaction management
ResourceAdapterInternalException - error condition internal to resource adapter
EISSystemException - EIS instance specific error condition

public void rollback() throws ResourceException

rollback
void rollback() throws ResourceException

Rollback a local transaction

**Throws:**
- ResourceException - generic exception if operation fails
- LocalTransactionException - error condition related to local transaction management
- ResourceAdapterInternalException - error condition internal to resource adapter
- EISSystemException - EIS instance specific error condition

PS:
javax.resource.spi  Class LocalTransactionException

java.lang.Object  
  ▼ java.lang.Throwable  
   ▼ java.lang.Exception  
    ▼ javax.resource.ResourceException  
     ▼ javax.resource.spi.LocalTransactionException

All Implemented Interfaces:
  Serializable

public class LocalTransactionException
  extends ResourceException

Extends: Throwable > Exception > ResourceException

LocalTransactionException  Java Transaction API  XAResource
  javax.transaction.xa.XAException

LocalTransactionException

  • LocalTransaction  commit
  • LocalTransaction  commit
  • ManagedConnection
  •

  version  1.0

A LocalTransactionException represents various error conditions related to the local transaction management contract. The Java Transaction API specification specifies the javax.transaction.xa.XAException class for exceptions related to XAResource based transaction management contract.

The LocalTransactionException is used for the local transaction management contract to indicate the following common error conditions:
• Invalid transaction context when a transaction operation is executed. For example, calling commit method on LocalTransaction object without an active local transaction is an error condition.
• Transaction is rolled back instead of getting committed during a commit method call on the LocalTransaction object.
• An attempt to start a local transaction from the same thread on a ManagedConnection that is already associated with an active local transaction.
• Any resource adapter or resource manager specific error conditions related to local transaction management. Examples are violation of integrity of resources, deadlock detection, communication failure during transaction completion, retry required or any internal error in a resource manager.

Version:
  1.0
Author:
  Rahul Sharma, Ram Jeyaraman
See Also:
  Serialized Form

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LocalTransactionException()</code></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><code>LocalTransactionException(String message)</code></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><code>LocalTransactionException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and an error code.</td>
</tr>
<tr>
<td><code>LocalTransactionException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>LocalTransactionException(Throwable cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>
**Method Summary**

<table>
<thead>
<tr>
<th>Methods inherited from class <code>javax.resource.ResourceException</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getErrorCode</code>, <code>getLinkedException</code>, <code>getMessage</code>, <code>setErrorCode</code>, <code>setLinkedException</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>java.lang.Throwable</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fillInStackTrace</code>, <code>getCause</code>, <code>getLocalizedMessage</code>, <code>getStackTrace</code>, <code>initCause</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>setStackTrace</code>, <code>toString</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class <code>java.lang.Object</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
</tr>
</tbody>
</table>

**Constructor Detail**

public `LocalTransactionException()`

null

`LocalTransactionException`

public `LocalTransactionException()`

Constructs a new instance with null as its detail message.

public `LocalTransactionException(String message)`

`message`

`LocalTransactionException`
public LocalTransactionException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.

public LocalTransactionException(Throwable cause)

cause throwable

cause Throwable

LocalTransactionException

public LocalTransactionException(Throwable cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

public LocalTransactionException(String message, Throwable cause)

cause throwable

message

cause Throwable

LocalTransactionException

public LocalTransactionException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.
Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public LocalTransactionException(String message, String errorCode)
throwable

message
errorCode

LocalTransactionException

public LocalTransactionException(String message, String errorCode)

Constructs a new throwable with the specified detail message and
an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS
SUMMARY | NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES
DETAIL | FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
public interface Location

Location

version 1.0

Provides information on the location of an event. All the information provided by a Location is optional. For example an application may only report line numbers.

Version:
  1.0
Author:
  Copyright (c) 2003 by BEA Systems. All Rights Reserved.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int <code>getCharacterOffset()</code></td>
<td>Return the byte or character offset into the input source this location is pointing to.</td>
</tr>
<tr>
<td>int <code>getColumnNumber()</code></td>
<td>Return the column number where the current event ends, returns -1 if none is available.</td>
</tr>
<tr>
<td>int <code>getLineNumber()</code></td>
<td>Return the line number where the current event ends, returns -1 if none is available.</td>
</tr>
<tr>
<td>String <code>getPublicId()</code></td>
<td>Returns the public ID of the XML</td>
</tr>
<tr>
<td>String <code>getSystemId()</code></td>
<td>Returns the system ID of the XML</td>
</tr>
</tbody>
</table>
public int getLineNumber()  
-1

    return

getLineNumber

int getLineNumber()  

    Return the line number where the current event ends, returns -1 if none is available.

    Returns:  
        the current line number

public int getColumnNumber()  
-1

    return

getColumnNumber

int getColumnNumber()  

    Return the column number where the current event ends, returns -1 if none is available.

    Returns:  
        the current column number

public int getCharacterOffset()  
-1
**getCharacterOffset**

```java
int getCharacterOffset()
```

Return the byte or character offset into the input source this location is pointing to. If the input source is a file or a byte stream then this is the byte offset into that stream, but if the input source is a character media then the offset is the character offset. Returns -1 if there is no offset available.

**Returns:**

the current offset

---

**public String getPublicId()**

```java
XML ID
```

```java
return ID null
```

**getPublicId**

```java
String getPublicId()
```

Returns the public ID of the XML

**Returns:**

the public ID, or null if not available

---

**public String getSystemId()**

```java
XML ID
```

```java
return ID null
```

**getSystemId**
String getSystemId()

Returns the system ID of the XML

**Returns:**
the system ID, or null if not available
javax.persistence Enum LockModeType

java.lang.Object
    - java.lang.Enum<LockModeType>
        - javax.persistence.LockModeType

All Implemented Interfaces:
    Serializable, Comparable&lt;LockModeType&gt;

public enum LockModeType
    extends Enum&lt;LockModeType&gt;

Extends: Enum&lt;E&gt;

    lock(EntityManager.lock())

    LockModeType.READ   LockModeType.WRITE

T1  lock(entity, LockModeType.READ)

    • P1 T1  T2  T1  T2  T1  T2
    • P2 T1  T2  T1

P1  P2

lock(entity, LockModeType.WRITE)

    lock(EntityManager.lock())

since  Java Persistence 1.0

Lock modes that can be specified by means of the EntityManager.lock() method.

The semantics of requesting locks of type LockModeType.READ and
LockModeType.WRITE are the following.

If transaction T1 calls lock(entity, LockModeType.READ) on a versioned object, the entity manager must ensure that neither of the following phenomena can occur:

- **P1 (Dirty read):** Transaction T1 modifies a row. Another transaction T2 then reads that row and obtains the modified value, before T1 has committed or rolled back. Transaction T2 eventually commits successfully; it does not matter whether T1 commits or rolls back and whether it does so before or after T2 commits.

- **P2 (Non-repeatable read):** Transaction T1 reads a row. Another transaction T2 then modifies or deletes that row, before T1 has committed. Both transactions eventually commit successfully.

Lock modes must always prevent the phenomena P1 and P2.

In addition, calling lock(entity, LockModeType.WRITE) on a versioned object, will also force an update (increment) to the entity's version column.

The persistence implementation is not required to support calling EntityManager.lock() on a non-versioned object. When it cannot support a such lock call, it must throw the PersistenceException.

Since:
Java Persistence 1.0

---

**Enum Constant Summary**

<table>
<thead>
<tr>
<th>Enum Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>READ</td>
<td>Read lock</td>
</tr>
<tr>
<td>WRITE</td>
<td>Write lock</td>
</tr>
</tbody>
</table>

---

**Method Summary**
static LockModeType.valueOf(String name)  
Returns the enum constant of this type with the specified name.

static LockModeType[] values()  
Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

READ

public static final LockModeType READ

  Read lock

WRITE

public static final LockModeType WRITE

  Write lock

Method Detail
final public static LockModeType[] values()

values

public static final LockModeType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

for(LockModeType c : LockModeType.values())
    System.out.println(c);

Returns:
    an array containing the constants of this enum type, in the order they're declared

public static LockModeType valueOf(String name)

valueOf

public static LockModeType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
    name - the name of the enum constant to be returned.

Returns:
    the enum constant with the specified name

Throws:
    IllegalArgumentException - if this enum type has no constant with the specified name
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
public interface LogicalHandler<C extends LogicalMessageContext>
extends Handler<C>

Implements: »Handler

LogicalHandler Handler

since JAX-WS 2.0

The LogicalHandler extends Handler to provide typesafety for the message context parameter.

Since: JAX-WS 2.0
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.ws Interface LogicalMessage

public interface LogicalMessage

LogicalMessage (protocol agnostic) XML

since JAX-WS 2.0

The LogicalMessage interface represents a protocol agnostic XML message and contains methods that provide access to the payload of the message.

Since:
JAX-WS 2.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Source&gt;getPayload()&lt;br&gt;Gets the message payload as an XML source, may be called multiple times on the same LogicalMessage instance, always returns a new Source that may be used to retrieve the entire message payload.</td>
</tr>
<tr>
<td>&lt;Object&gt;getPayload(JAXBContext context)&lt;br&gt;Gets the message payload as a JAXB object.</td>
</tr>
<tr>
<td>void setPayload(Object payload, JAXBContext context)&lt;br&gt;Sets the message payload</td>
</tr>
<tr>
<td>void setPayload(Source payload)&lt;br&gt;Sets the message payload</td>
</tr>
</tbody>
</table>

Method Detail
public javax.xml.transform.Source getPayload()

XML LogicalMessage Source

Source DOMSource DOM Source
setPayload

return null

getPayload

getPayload()

Gets the message payload as an XML source, may be called multiple times on the same LogicalMessage instance, always returns a new Source that may be used to retrieve the entire message payload.

If the returned Source is an instance of DOMSource, then modifications to the encapsulated DOM tree change the message payload in-place, there is no need to subsequently call setPayload. Other types of Source provide only read access to the message payload.

Returns:
The contained message payload; returns null if no payload is present in this message.

public void setPayload(javax.xml.transform.Source payload)

payload

Throws WebServiceException:

Throws UnsupportedOperationException:
setPayload

void setPayload(Source payload)

Sets the message payload

Parameters:
    payload - message payload

Throws:
    WebServiceException - If any error during the setting of the payload in this message
    UnsupportedOperationException - If this operation is not supported

public Object getPayload(JAXBContext context)

JAXB

context JAXBContext
return null

Throws 
    WebServiceException: JAXBContext
    WebServiceException cause JAXBException

getPayload

Object getPayload(JAXBContext context)

Gets the message payload as a JAXB object. Note that there is no connection between the returned object and the message payload, changes to the payload require calling setPayload.

Parameters:
    context - The JAXBContext that should be used to unmarshall the message payload

Returns:
    The contained message payload; returns null if no payload is present in this message

Throws:
    WebServiceException - If an error occurs when using a supplied
JAXBContext to unmarshall the payload. The cause of the WebServiceException is the original JAXBException.

```java
public void setPayload(Object payload, JAXBContext context)
```

**payload**

**context**

JAXBContext

**Throws**

UnsupportedOperationException:

**Throws**

WebServiceException: JAXBContext

WebServiceException cause JAXBException

---

**setPayload**

```java
void setPayload(Object payload,
JAXBContext context)
```

Sets the message payload

**Parameters:**

- payload - message payload
- context - The JAXBContext that should be used to marshall the payload

**Throws:**

- UnsupportedOperationException - If this operation is not supported
- WebServiceException - If an error occurs when using the supplied JAXBContext to marshall the payload. The cause of the WebServiceException is the original JAXBException.
javax.xml.ws.handler Interface LogicalMessageContext

All Superinterfaces:
   Map<String, Object>, MessageContext

public interface LogicalMessageContext
extends MessageContext

Implements: MessageContext

LogicalMessageContext MessageContext LogicalMessage
   since JAX-WS 2.0

The LogicalMessageContext interface extends MessageContext to provide access to a the contained message as a protocol neutral LogicalMessage

Since:
   JAX-WS 2.0

Nested Class Summary

Nested classes/interfaces inherited from interface javax.xml.ws.handler.MessageContext
MessageContext.Scope

Nested classes/interfaces inherited from interface java.util.Map
Map.Entry<K, V>

Field Summary

Fields inherited from interface
### Method Summary

<table>
<thead>
<tr>
<th>LogicalMessage</th>
<th>getMessage()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the message from this message context</td>
</tr>
</tbody>
</table>

### Methods inherited from interface javax.xml.ws.handler.MessageContext

- `getScope`, `setScope`

### Methods inherited from interface java.util.Map

- `clear`, `containsKey`, `containsValue`, `entrySet`, `equals`, `get`, `hashCode`, `isEmpty`, `keySet`, `put`, `putAll`, `remove`, `size`, `values`

### Method Detail

```java
public LogicalMessage getMessage()
```

```java
return null
```

### getMessage

```java
LogicalMessage getMessage()
```

Gets the message from this message context

**Returns:**

The contained message; returns null if no message is present in
this message context

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.convert Class LongConverter

java.lang.Object
   javax.faces.convert.LongConverter

All Implemented Interfaces:
   Converter

public class LongConverter
extends Object
implements Converter

Implements: Converter

java.lang.Long long value     Converter

Converter implementation for java.lang.Long (and long primitive) values.

---

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String CONVERETER_ID</td>
<td>The standard converter id for this converter.</td>
<td></td>
</tr>
<tr>
<td>static String LONG_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion to Long fails.</td>
<td></td>
</tr>
<tr>
<td>static String STRING_ID</td>
<td>The message identifier of the FacesMessage to be created if the conversion of the Long value to String fails.</td>
<td></td>
</tr>
</tbody>
</table>

Constructor Summary

LongConverter()
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAsObject</strong> (FacesContext context, UIComponent component, String value)</td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><strong>AsString</strong> (FacesContext context, UIComponent component, Object value)</td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

#### Methods inherited from class java.lang.Object
- Clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Field Detail

**CONVERTER_ID**

```java
public static final String CONVERTER_ID
```

The standard converter id for this converter.

**See Also:**
- Constant Field Values
LONG_ID

public static final String LONG_ID

The message identifier of the FacesMessage to be created if the conversion to Long fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

STRING_ID

public static final String STRING_ID

The message identifier of the FacesMessage to be created if the conversion of the Long value to String fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

Constructor Detail

public LongConverter()
LongConverter

public LongConverter()

Method Detail

public Object AsObject(FacesContext context, UIComponent component, String value)

Throws ConverterException: NullPointerException
Throws NullPointerException: NullPointerException context

getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

Description copied from interface: Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

Specified by:
getAsObject in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - String value to be converted (may be null)

Returns:
null if the value to convert is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws ConverterException: NullPointerException

Throws NullPointerException: context component null

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.rpc.holders  Class LongHolder

java.lang.Object
   ↓ javax.xml.rpc.holders.LongHolder

All Implemented Interfaces:
   Holder

public final class LongHolder
extends Object
implements Holder

Implements: Holder

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>long value</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>LongHolder()</td>
</tr>
<tr>
<td>LongHolder(long mylong)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| Methods inherited from class java.lang.Object 
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |
public long value

public LongHolder()

LongHolder

public LongHolder()

public LongHolder(long mylong)

LongHolder

public LongHolder(long mylong)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.faces.validator Class LongRangeValidator

java.lang.Object
   ↓ java.lang.Number
      ↓ javax.faces.validator.LongRangeValidator

All Implemented Interfaces:
   javax.faces.validator.Validator, javax.faces.validator.StateHolder

public class LongRangeValidator
extends java.lang.Object
implements javax.faces.validator.Validator, javax.faces.validator.StateHolder

Implements: javax.faces.validator.Validator, javax.faces.validator.StateHolder

LongRangeValidator extends Validator

- null
- long String TYPE_MESSAGE_ID
- Validator maximum minimum
  #NOT_IN_RANGE_MESSAGE_ID  ValidatorException
- Validator maximum MAXIMUM_MESSAGE_ID
  ValidatorException
- Validator minimum MINIMUM_MESSAGE_ID
  ValidatorException

ValidatorException ID javax.faces.Number Converter
Locale

LongRangeValidator is a Validator that checks the value of the corresponding component against specified minimum and maximum values. The following algorithm is implemented:

- If the passed value is null, exit immediately.
- If the current component value is not a floating point type, or a String
that is convertible to long, throw a `ValidatorException` containing a `TYPE_MESSAGE_ID` message.

- If both a `maximum` and `minimum` property has been configured on this `Validator`, check the component value against both limits. If the component value is not within this specified range, throw a `ValidatorException` containing a `NOT_IN_RANGE_MESSAGE_ID` message.
- If a `maximum` property has been configured on this `Validator`, check the component value against this limit. If the component value is greater than the specified maximum, throw a `ValidatorException` containing a `MAXIMUM_MESSAGE_ID` message.
- If a `minimum` property has been configured on this `Validator`, check the component value against this limit. If the component value is less than the specified minimum, throw a `ValidatorException` containing a `MINIMUM_MESSAGE_ID` message.

For all of the above cases that cause a `ValidatorException` to be thrown, if there are parameters to the message that match up with validator parameters, the values of these parameters must be converted using the `Converter` registered in the application under the converter id `javax.faces.Number`. This allows the values to be localized according to the current `Locale`.

### Field Summary

<table>
<thead>
<tr>
<th>Static String</th>
<th><code>MAXIMUM_MESSAGE_ID</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the maximum value check fails.</td>
</tr>
<tr>
<td>Static String</td>
<td><code>MINIMUM_MESSAGE_ID</code></td>
</tr>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the minimum value check fails.</td>
</tr>
<tr>
<td>Static String</td>
<td><code>NOT_IN_RANGE_MESSAGE_ID</code></td>
</tr>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set.</td>
</tr>
<tr>
<td>Static String</td>
<td><code>TYPE_MESSAGE_ID</code></td>
</tr>
<tr>
<td></td>
<td>The message identifier of the <code>FacesMessage</code> to be</td>
</tr>
</tbody>
</table>
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LongRangeValidator()</code></td>
<td>Construct a <code>Validator</code> with no preconfigured limits.</td>
</tr>
<tr>
<td><code>LongRangeValidator(long maximum)</code></td>
<td>Construct a <code>Validator</code> with the specified preconfigured limit.</td>
</tr>
<tr>
<td><code>LongRangeValidator(long maximum, long minimum)</code></td>
<td>Construct a <code>Validator</code> with the specified preconfigured limits.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean equals(Object otherObj)</code></td>
<td></td>
</tr>
<tr>
<td><code>long getMaximum()</code></td>
<td>Return the maximum value to be enforced by this <code>Validator</code>.</td>
</tr>
<tr>
<td><code>long getMinimum()</code></td>
<td>Return the minimum value to be enforced by this <code>Validator</code>.</td>
</tr>
<tr>
<td><code>int hashCode()</code></td>
<td></td>
</tr>
<tr>
<td><code>boolean isTransient()</code></td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td><code>void restoreState(FacesContext context, Object state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>Object saveState(FacesContext context)</code></td>
<td>Gets the state of the instance as a <code>Serializable</code> Object.</td>
</tr>
<tr>
<td><code>void setMaximum(long maximum)</code></td>
<td></td>
</tr>
</tbody>
</table>
Set the maximum value to be enforced by this Validator.

```java
void setMinimum(long minimum)
```

Set the minimum value to be enforced by this Validator.

```java
void setTransient(boolean transientValue)
```

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

```java
void validate(FacesContext context, UIComponent component, Object value)
```

Perform the correctness checks implemented by this Validator against the specified UIComponent.

### Methods inherited from class java.lang.Object

class java.lang.Object

- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
maximum value check fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the configured maximum value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

---

MINIMUM_MESSAGE_ID

public static final String MINIMUM_MESSAGE_ID

The message identifier of the FacesMessage to be created if the minimum value check fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the configured minimum value.
- `{1}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values

---

NOT_IN_RANGE_MESSAGE_ID

public static final String NOT_IN_RANGE_MESSAGE_ID

The message identifier of the FacesMessage to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set. The message format string for this message may optionally include the following placeholders:
- `{0}` replaced by the configured minimum value.
- `{1}` replaced by the configured maximum value.
- `{2}` replaced by a `String` whose value is the label of the input component that produced this message.

**See Also:**
- [Constant Field Values](#)

### TYPE_MESSAGE_ID

```java
public static final String TYPE_MESSAGE_ID
```

The message identifier of the `FacesMessage` to be created if the current value of this component is not of the correct type. The message format string for this message may optionally include a `{0}` placeholder that will be replaced by a `String` whose value is the label of the input component that produced this message.

**See Also:**
- [Constant Field Values](#)

### Constructor Detail

**public LongRangeValidator()

Validator**

### LongRangeValidator

**public LongRangeValidator()

Construct a `Validator` with no preconfigured limits.**
public LongRangeValidator(long maximum)

Validator

maximum

LongRangeValidator

public LongRangeValidator(long maximum)

Construct a Validator with the specified preconfigured limit.

Parameters:
maximum - Maximum value to allow

public LongRangeValidator(long maximum, long minimum)

Validator

maximum
minimum

LongRangeValidator

public LongRangeValidator(long maximum, long minimum)

Construct a Validator with the specified preconfigured limits.

Parameters:
maximum - Maximum value to allow
minimum - Minimum value to allow

Method Detail
public long getMaximum()

    Validator

getMaximum

public long getMaximum()

    Return the maximum value to be enforced by this Validator.

public void setMaximum(long maximum)

    Validator

    maximum

setMaximum

public void setMaximum(long maximum)

    Set the maximum value to be enforced by this Validator.

    Parameters:
    maximum - The new maximum value

public long getMinimum()

    Validator

getMinimum
public long getMinimum()

Return the minimum value to be enforced by this Validator.

public void setMinimum(long minimum)

Validator

minimum

setMinimum

public void setMinimum(long minimum)

Set the minimum value to be enforced by this Validator.

Parameters:
minimum - The new minimum value

public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException

Throws

NullPointerException: NullPointerEception context
cOMPONENT null

ThOws ValidatorException: NullPointerEception

validate

public void validate(FacesContext context, UIComponent component, Object value)

throws ValidatorException

Description copied from interface: Validator

Perform the correctness checks implemented by this Validator
against the specified UICOMPONENT. If any violations are found, a ValidatorException will be thrown containing the FacesMessage describing the failure.

**Specified by:**

`validate` in interface `Validator`

**Parameters:**

- `context` - FacesContext for the request we are processing
- `component` - UICOMPONENT we are checking for correctness
- `value` - the value to validate

**Throws:**

- `NullPointerException` - if context or component is null
- `ValidatorException` - if validation fails

---

**public boolean equals(Object otherObj)**

**equals**

**public boolean equals(Object otherObj)**

**Overrides:**

`equals` in class `Object`

---

**public int hashCode()**

**hashCode**

**public int hashCode()**

**Overrides:**

`hashCode` in class `Object`

---

**public Object saveState(FacesContext context)**
saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

   Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:
   saveState in interface StateHolder

public void restoreState(FacesContext context, Object state)

restoreState

public void restoreState(FacesContext context, Object state)
Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

Specified by:  
`restoreState` in interface `StateHolder`

---

**public boolean isTransient()**

`isTransient`

**public boolean isTransient()**

Description copied from interface: **StateHolder**

If true, the Object implementing this interface must not participate in state saving or restoring.

Specified by:  
`isTransient` in interface `StateHolder`

---

**public void setTransient(boolean transientValue)**

`setTransient`

**public void setTransient(boolean transientValue)**

Description copied from interface: **StateHolder**
Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Specified by:**

`setTransient` in interface `StateHolder`

**Parameters:**

`transientValue` - boolean pass `true` if this Object will participate in state saving or restoring, otherwise pass `false`. 
java.lang.Object
   javax.xml.rpc.holders

All Implemented Interfaces:
   Holder

public final class LongWrapperHolder
extends Object
implements Holder

Implements: Holder

Field Summary

| Long | value |

Constructor Summary

| LongWrapperHolder() |
| LongWrapperHolder(Long mylong) |

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
value

public Long value

Constructor Detail

public LongWrapperHolder()

LongWrapperHolder

public LongWrapperHolder()

----------------

public LongWrapperHolder(Long mylong)

LongWrapperHolder

public LongWrapperHolder(Long mylong)

----------------

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
</tr>
</tbody>
</table>
javax.activation  Class MailcapCommandMap

java.lang.Object
   └ javax.activation.CommandMap
      └ javax.activation.MailcapCommandMap

public class MailcapCommandMap
extends CommandMap

Extends: CommandMap

MailcapCommandMap CommandMap mailcap (1524) CommandMap MailcapCommandMap

mailcap

MailcapCommandMap mailcap MailcapCommandMap mailcap

1. MailcapCommandMap
2. .mailcap
3. <java.home>/lib/mailcap
4. META-INF/mailcap
5. META-INF/mailcap.default activation.jar

mailcap

mailcap mailcap (RFC 1524, A User Agent Configuration Mechanism For Multimedia Mail Format Information) MIME MIME MailcapCommandMap mailcap Java MIME

mailcap MailcapCommandMap x-j
MailcapCommandMap name \
x-java-<name> name content-handler
MailcapCommandMap DataContentHandler JavaBean
   x-java-view=com.foo.ViewBean
MailcapCommandMap extends the CommandMap abstract class. It implements a CommandMap whose configuration is based on mailcap files ([RFC 1524](https://www.rfc-editor.org/info/rfc1524)). The MailcapCommandMap can be configured both programmatically and via configuration files.

**Mailcap file search order:**

The MailcapCommandMap looks in various places in the user's system for mailcap file entries. When requests are made to search for commands in the MailcapCommandMap, it searches mailcap files in the following order:

1. Programatically added entries to the MailcapCommandMap instance.
2. The file `.mailcap` in the user's home directory.
3. The file `<java.home>/lib/mailcap`.
4. The file or resources named META-INF/mailcap.
5. The file or resource named META-INF/mailcap.default (usually found only in the activation.jar file).

**Mailcap file format:**
Mailcap files must conform to the mailcap file specification (RFC 1524, *A User Agent Configuration Mechanism For Multimedia Mail Format Information*). The file format consists of entries corresponding to particular MIME types. In general, the specification specifies applications for clients to use when they themselves cannot operate on the specified MIME type. The MailcapCommandMap extends this specification by using a parameter mechanism in mailcap files that allows JavaBeans(tm) components to be specified as corresponding to particular commands for a MIME type.

When a mailcap file is parsed, the MailcapCommandMap recognizes certain parameter signatures, specifically those parameter names that begin with x-java-. The MailcapCommandMap uses this signature to find command entries for inclusion into its registries. Parameter names with the form x-java-<name> are read by the MailcapCommandMap as identifying a command with the name name. When the name is content-handler the MailcapCommandMap recognizes the class signified by this parameter as a DataContentHandler. All other commands are handled generically regardless of command name. The command implementation is specified by a fully qualified class name of a JavaBean(tm) component. For example; a command for viewing some data can be specified as: x-java-view=com.foo.ViewBean.

When the command name is fallback-entry, the value of the command may be true or false. An entry for a MIME type that includes a parameter of x-java-fallback-entry=true defines fallback commands for that MIME type that will only be used if no non-fallback entry can be found. For example, an entry of the form text/*; x-java-fallback-entry=true; x-java-view=com.sun.TextViewer specifies a view command to be used for any text MIME type. This view command would only be used if a non-fallback view command for the MIME type could not be found.

MailcapCommandMap aware mailcap files have the following general form:

```
# Comments begin with a '#' and continue to the end of the line.
mime type; ; <parameter list>
# Where a parameter list consists of one or more parameters,
# where parameters look like: x-java-view=com.sun.TextViewer
# and a parameter list looks like:
```


# Note that mailcap entries that do not contain 'x-java' parameters and comply to RFC 1524 are simply ignored:
image/gif; /usr/dt/bin/sdtimage %s

**Author:**
Bart Calder, Bill Shannon

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MailcapCommandMap()</strong></td>
<td>The default Constructor.</td>
</tr>
<tr>
<td><strong>MailcapCommandMap(InputStream is)</strong></td>
<td>Constructor that allows the caller to specify an InputStream containing a mailcap file.</td>
</tr>
<tr>
<td><strong>MailcapCommandMap(String fileName)</strong></td>
<td>Constructor that allows the caller to specify the path of a <em>mailcap</em> file.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <strong>addMailcap(String mail_cap)</strong></td>
<td>Add entries to the registry.</td>
</tr>
<tr>
<td><strong>DataContentHandler createDataContentHandler(String mimeType)</strong></td>
<td>Return the DataContentHandler for the specified MIME type.</td>
</tr>
<tr>
<td><strong>CommandInfo[] getAllCommands(String mimeType)</strong></td>
<td>Get all the available commands in all mailcap files known to this instance of MailcapCommandMap for this MIME type.</td>
</tr>
<tr>
<td><strong>CommandInfo getCommand(String mimeType, String cmdName)</strong></td>
<td>Get the command corresponding to <em>cmdName</em> for the MIME type.</td>
</tr>
<tr>
<td><strong>String[] getMimeTypes()</strong></td>
<td>Get all the MIME types known to this command map.</td>
</tr>
</tbody>
</table>
getNativeCommands(String mimeType)
Get the native commands for the given MIME type.

getPreferredCommands(String mimeType)
Get the preferred command list for a MIME Type.

Methods inherited from class javax.activation.CommandMap
createDataContentHandler, getAllCommands, getCommand,
getDefaultCommandMap, getPreferredCommands, setDefaultCommandMap

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Constructor Detail

public MailcapCommandMap()

MailcapCommandMap

public MailcapCommandMap()

    The default Constructor.

public MailcapCommandMap(String fileName) throws java.io.IOException

    mailcap

    fileName

    mailcap

    Throws

    java.io.IOException:
MailcapCommandMap

public MailcapCommandMap(String fileName)
throws IOException

    Constructor that allows the caller to specify the path of a mailcap file.

    Parameters:
    fileName - The name of the mailcap file to open

    Throws:
    IOException - if the file can't be accessed

public MailcapCommandMap(java.io.InputStream is)

    Constructor that allows the caller to specify an InputStream containing a mailcap file.

    Parameters:
    is - InputStream of the mailcap file to open

Method Detail

public CommandInfo[] getPreferredCommands(String mimeType)

    MIME MailcapCommandMap
    n
    mailcap

    MailcapCommandMap mailmap
    mimeType MIME
public CommandInfo[] getPreferredCommands(String mimeType)

Get the preferred command list for a MIME Type. The MailcapCommandMap searches the mailcap files as described above under Mailcap file search order.

The result of the search is a proper subset of available commands in all mailcap files known to this instance of MailcapCommandMap. The first entry for a particular command is considered the preferred command.

Specified by:  
getPreferredCommands in class CommandMap

Parameters:  
mimeType - the MIME type

Returns:  
the CommandInfo objects representing the preferred commands.

public CommandInfo[] getAllCommands(String mimeType)

Get all the available commands in all mailcap files known to this instance of MailcapCommandMap for this MIME type.
Specified by:
getAllCommands in class CommandMap

Parameters:
mimeType - the MIME type

Returns:
the CommandInfo objects representing all the commands.

public CommandInfo getCommand(String mimeType, String cmdName)

MIME mimeType
     cmdName
     MIME
     cmdName
     return

public void addMailcap(String mail_cap)

mailcap
    mail_cap mailcap
**addMailcap**

```java
public void addMailcap(String mail_cap)
```

Add entries to the registry. Programatically added entries are searched before other entries.

The string that is passed in should be in mailcap format.

**Parameters:**
- `mail_cap` - a correctly formatted mailcap string

---

**createDataContentHandler**

```java
public DataContentHandler createDataContentHandler(String mimeType)
```

MIME `DataContentHandler`

<table>
<thead>
<tr>
<th><code>mimeType</code></th>
<th>MIME</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>return</td>
</tr>
<tr>
<td></td>
<td>DataContentHandler</td>
</tr>
</tbody>
</table>

**createDataContentHandler**

```java
public DataContentHandler createDataContentHandler(String mimeType)
```

Return the DataContentHandler for the specified MIME type.

**Specified by:**
- `createDataContentHandler` in class `CommandMap`

**Parameters:**
- `mimeType` - the MIME type

**Returns:**
- the DataContentHandler

---

**getMimeTypes**

```java
public String[] getMimeTypes()
```

MIME
getMimeTypes

public String[] getMimeTypes()

Get all the MIME types known to this command map.

Overrides:
    getMimeTypes in class CommandMap

Returns:
    array of MIME types as strings

Since:
    JAF 1.1

getNativeCommands

public String[] getNativeCommands(String mimeType)

Get the native commands for the given MIME type. Returns an array of strings where each string is an entire mailcap file entry. The application will need to parse the entry to extract the actual command as well as any attributes it needs. See RFC 1524 for details of the mailcap entry syntax. Only mailcap entries that specify a view command for the specified MIME type are returned.

Returns:
    array of native command entries
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail.internet  **Class MailDateFormat**

java.lang.Object  
  - java.text.Format  
    - java.text.DateFormat  
      - java.text.SimpleDateFormat  
        - javax.mail.internet.MailDateFormat

All Implemented Interfaces:  
  Serializable, Cloneable

```
public class MailDateFormat 
extends SimpleDateFormat

Extends: java.text.Format > java.text.DateFormat > java.text.SimpleDateFormat

2000 1 26 draft-ietf-drums-msg-fmt-08 RFC822

3.3

-- FWS  FWS

date-time = [ day-of-week "," ] date FWS time [CFWS]
day-of-week = ([FWS] day-name) / obs-day-of-week
day-name = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"
date = day month year
year = 4*DIGIT / obs-year
month = (FWS month-name FWS) / obs-month
month-name = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" / "Jul" / "Aug" /
```
"Sep" / "Oct" / "Nov" / "Dec"

day = ([FWS] 1*2DIGIT) / obs-day

time = time-of-day FWS zone

time-of-day = hour ":" minute [ "":" second ]

hour = 2DIGIT / obs-hour

minute = 2DIGIT / obs-minute

second = 2DIGIT / obs-second

zone = (( "+" / "-" ) 4DIGIT) / obs-zone

day year

time-of-day

date time-of-day

zone date time-of-day UTC""""+-"" time-of-day +hhmm +(hh * 60 + mm) -hhmm -(hh * 60 + mm) ""+0000" ""-0000"

-day-of-week day-of-month 1 time-of-day 00:00:00 23:59:60 [STD-12]zone -9959 +9959

since JavaMail 1.2 en

Formats and parses date specification based on the draft-ietf-drums-msg-fmt-08 dated January 26, 2000. This is a followup spec to RFC822.

This class does not take pattern strings. It always formats the date based on the specification below.
3.3 Date and Time Specification

Date and time occur in several header fields of a message. This section specifies the syntax for a full date and time specification. Though folding whitespace is permitted throughout the date-time specification, it is recommended that only a single space be used where FWS is required and no space be used where FWS is optional in the date-time specification; some older implementations may not interpret other occurrences of folding whitespace correctly.

date-time = [ day-of-week , ] date FWS time [CFWS]
day-of-week = ([FWS] day-name) / obs-day-of-week
day-name = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"
date = day month year
year = 4*DIGIT / obs-year
month = (FWS month-name FWS) / obs-month
day = ([FWS] 1*2DIGIT) / obs-day
time = time-of-day FWS zone
time-of-day = hour : minute [ : second ]
hour = 2DIGIT / obs-hour
minute = 2DIGIT / obs-minute
second = 2DIGIT / obs-second
zone = ("+" / "-" ) 4DIGIT) / obs-zone

The day is the numeric day of the month. The year is any numeric year in
the common era.

The time-of-day specifies the number of hours, minutes, and optionally seconds since midnight of the date indicated.

The date and time-of-day SHOULD express local time.

The zone specifies the offset from Coordinated Universal Time (UTC, formerly referred to as "Greenwich Mean Time") that the date and time-of-day represent. The "+" or "-" indicates whether the time-of-day is ahead of or behind Universal Time. The first two digits indicate the number of hours difference from Universal Time, and the last two digits indicate the number of minutes difference from Universal Time. (Hence, +hhmm means +(hh * 60 + mm) minutes, and -hhmm means -(hh * 60 + mm) minutes). The form "+0000" SHOULD be used to indicate a time zone at Universal Time. Though "-0000" also indicates Universal Time, it is used to indicate that the time was generated on a system that may be in a local time zone other than Universal Time.

A date-time specification MUST be semantically valid. That is, the day-of-the-week (if included) MUST be the day implied by the date, the numeric day-of-month MUST be between 1 and the number of days allowed for the specified month (in the specified year), the time-of-day MUST be in the range 00:00:00 through 23:59:60 (the number of seconds allowing for a leap second; see [STD-12]), and the zone MUST be within the range -9959 through +9959.

Since:
    JavaMail 1.2
Author:
    Max Spivak
See Also:
    Serialized Form

---

**Nested Class Summary**

| Nested classes/interfaces inherited from class |
java.text.**DateFormat**

**DateFormat.Field**

### Field Summary

Fields inherited from class java.text.**DateFormat**

- AM_PM_FIELD
- calendar
- DATE_FIELD
- DAY_OF_WEEK_FIELD
- DAY_OF_WEEK_IN_MONTH_FIELD
- DAY_OF_YEAR_FIELD
- DEFAULT
- ERA_FIELD
- FULL
- HOUR_OF_DAY0_FIELD
- HOUR_OF_DAY1_FIELD
- HOUR0_FIELD
- HOUR1_FIELD
- LONG
- MEDIUM
- MILLISECOND_FIELD
- MINUTE_FIELD
- MONTH_FIELD
- numberFormat
- SECOND_FIELD
- SHORT
- TIMEZONE_FIELD
- WEEK_OF_MONTH_FIELD
- WEEK_OF_YEAR_FIELD
- YEAR_FIELD

### Constructor Summary

- **MailDateFormat**()

### Method Summary

- **format** *(Date date, StringBuffer dateStrBuf, FieldPosition fieldPosition)*
  - Formats the given date in the format specified by draft-ietf-drums-msg-fmt-08 in the current TimeZone.

- **parse** *(String text, ParsePosition pos)*
  - Parses the given date in the format specified by draft-ietf-drums-msg-fmt-08 in the current TimeZone.

- **setCalendar** *(Calendar newCalendar)*
  - Don't allow setting the calendar

- **setNumberFormat** *(NumberFormat newNumberFormat)*
  - Don't allow setting the NumberFormat

Methods inherited from class java.text.**SimpleDateFormat**

- applyLocalizedPattern
- applyPattern
- clone
- equals
- formatToCharacterIterator
- get2DigitYearStart
- getDateFormatSymbols
- hashCode
- set2DigitYearStart
- setDateFormatSymbols
- toLocalizedPattern
- toPattern
Methods inherited from class java.text.\texttt{DateFormat}

\texttt{format, format, getAvailableLocales, getCalendar, getDateInstance, getDateInstance, getDateInstance, getDateTimeInstance, getDateTimeInstance, getInstance, getNumberFormat, getDateTimeInstance, getTimeInstance, getTimeInstance, getTimeZone, isLenient, parse, parseObject, setLenient, setTimeZone}

Methods inherited from class java.text.\texttt{Format}

\texttt{format, parseObject}

Methods inherited from class java.lang.\texttt{Object}

\texttt{finalize, getClass, notify, notifyAll, toString, wait, wait, wait}

**Constructor Detail**

public MailDateFormat()

**Method Detail**

public StringBuffer format(java.util.Date date, StringBuffer dateStrBuf, java.text.FieldPosition fieldPosition)

\texttt{TimeZone\ draft-ietf-drums-msg-fmt-08}

\texttt{date\ Date}

\texttt{dateStrBuf\ StringBuffer}

\texttt{fieldPosition\ String}

\texttt{return\ JavaMail 1.2}

\texttt{since\ en}
public StringBuffer format(Date date, StringBuffer dateStrBuf, FieldPosition fieldPosition)

Formats the given date in the format specified by draft-ietf-drums-msg-fmt-08 in the current TimeZone.

Overrides:
  format in class SimpleDateFormat

Parameters:
  date - the Date object
  dateStrBuf - the formatted string
  fieldPosition - the current field position

Returns:
  StringBuffer the formatted String

Since:
  JavaMail 1.2

public java.util.Date parse(String text, java.text.ParsePosition pos)
TimeZone draft-ietf-drums-msg-fmt-08
  text
  pos
  return Date (Date)
  since JavaMail 1.2

parse

public Date parse(String text, ParsePosition pos)

  Parses the given date in the format specified by draft-ietf-drums-
msg-fmt-08 in the current TimeZone.

Overrides:
parse in class SimpleDateFormat

Parameters:
text - the formatted date to be parsed
pos - the current parse position

Returns:
Date the parsed date in a Date object

Since:
JavaMail 1.2

public void setCalendar(java.util.Calendar newCalendar)

setCalendar

public void setCalendar(Calendar newCalendar)

Don't allow setting the calendar

Overrides:
setCalendar in class DateFormat

public void setNumberFormat(java.text.NumberFormat newNumberFormat)

NumberFormat

setNumberFormat

public void setNumberFormat(NumberFormat newNumberFormat)

Don't allow setting the NumberFormat
Overrides:

`setNumberFormat` in class `DateFormat`
java.mail.event Class MailEvent

java.lang.Object  
   java.util.EventObject  
      javax.mail.event.MailEvent

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   ConnectionEvent, FolderEvent, MessageChangedEvent, MessageCountEvent, StoreEvent, TransportEvent

public abstract class MailEvent
extends EventObject

Extends: java.util.EventObject
Extended by: ConnectionEvent, FolderEvent, MessageChangedEvent, MessageCountEvent, StoreEvent, TransportEvent

Common base class for mail events, defining the dispatch method.

Author:
   Bill Shannon

See Also:
   Serialized Form

Field Summary

Fields inherited from class java.util.EventObject

source
**Constructor Summary**

| Constructor | MailEvent(Object source) |

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract</td>
<td>public void dispatch(Object listener)</td>
</tr>
<tr>
<td></td>
<td>This method invokes the appropriate method on a listener for this event.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.util.EventObject**
- getSource, toString

**Methods inherited from class java.lang.Object**
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Constructor Detail**

public MailEvent(Object source)

**Method Detail**

abstract public void dispatch(Object listener)
dispatch

public abstract void dispatch(Object listener)

This method invokes the appropriate method on a listener for this event. Subclasses provide the implementation.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.resource.spi  Interface ManagedConnection

public interface ManagedConnection

ManagedConnection EIS
ManagedConnection javax.transaction.xa.XAResource
javax.resource.spi.LocalTransaction

XAResource EIS ManagedConnection
XAResource JTA

LocalTransaction

version 0.5

See also
javax.resource.spi.ManagedConnectionFactory,
javax.transaction.xa.XAResource,
javax.resource.spi.LocalTransaction

ManagedConnection instance represents a physical connection to the underlying EIS.

A ManagedConnection instance provides access to a pair of interfaces: javax.transaction.xa.XAResource and javax.resource.spi.LocalTransaction.

XAResource interface is used by the transaction manager to associate and dissociate a transaction with the underlying EIS resource manager instance and to perform two-phase commit protocol. The ManagedConnection interface is not directly used by the transaction manager. More details on the XAResource interface are described in the JTA specification.

The LocalTransaction interface is used by the application server to manage local transactions.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addConnectionEventListener(ConnectionEventListener)</code></td>
<td>Adds a connection event listener to the ManagedConnection instance.</td>
</tr>
<tr>
<td><code>void associateConnection(Object connection)</code></td>
<td>Used by the container to change the association of an application-level connection handle with a ManagedConnection instance.</td>
</tr>
<tr>
<td><code>void cleanup()</code></td>
<td>Application server calls this method to force a cleanup on the ManagedConnection instance.</td>
</tr>
<tr>
<td><code>void destroy()</code></td>
<td>Destroys the physical connection to the underlying resource manager.</td>
</tr>
<tr>
<td><code>Object getConnection(Subject subject, ConnectionRequestInfo cxRequestInfo)</code></td>
<td>Creates a new connection handle for the underlying physical connection represented by the ManagedConnection instance.</td>
</tr>
<tr>
<td><code>LocalTransaction getLocalTransaction()</code></td>
<td>Returns a <code>javax.resource.spi.LocalTransaction</code> object.</td>
</tr>
<tr>
<td><code>PrintWriter getLogWriter()</code></td>
<td>Gets the log writer for this ManagedConnection instance.</td>
</tr>
<tr>
<td><code>ManagedConnectionMetaData getMetaData()</code></td>
<td>Gets the metadata information for this connection's underlying EIS resource manager instance.</td>
</tr>
<tr>
<td><code>XAResource getXAResource()</code></td>
<td>Returns a <code>javax.transaction.xa.XAResource</code> object.</td>
</tr>
<tr>
<td><code>removeConnectionEventListener(ConnectionEventListener)</code></td>
<td>Removes a connection event listener from the ManagedConnection instance.</td>
</tr>
</tbody>
</table>
void
Removes an already registered connection event listener from the ManagedConnection instance.

void setLogWriter(PrintWriter out)
Sets the log writer for this ManagedConnection instance.

**Method Detail**

**public Object getConnection(javax.security.auth.Subject subject, ConnectionRequestInfo cxRequestInfo) throws ResourceException**

ManagedConnection Subject ConnectionRequest

subject JAAS

cxRequestInfo ConnectionRequestInfo

return Object CCIManagedConnection

javax.resource.cci.Connection

Throws ResourceException:

Throws ResourceAdapterInternalException:

Throws SecurityException:

Throws CommException: EIS

Throws EISSystemException: EIS EIS

ManagedConnection

getConnection

**Object getConnection(Subject subject, ConnectionRequestInfo cxRequestInfo) throws ResourceException**

Creates a new connection handle for the underlying physical connection represented by the ManagedConnection instance. This
connection handle is used by the application code to refer to the underlying physical connection. This connection handle is associated with its ManagedConnection instance in a resource adapter implementation specific way.

The ManagedConnection uses the Subject and additional ConnectionRequest Info (which is specific to resource adapter and opaque to application server) to set the state of the physical connection.

**Parameters:**
- `subject` - security context as JAAS subject
- `cxRequestInfo` - ConnectionRequestInfo instance

**Returns:**
- generic Object instance representing the connection handle. For CCI, the connection handle created by a ManagedConnection instance is of the type `javax.resource.cci.Connection`.

**Throws:**
- `ResourceException` - generic exception if operation fails
- `ResourceAdapterInternalException` - resource adapter internal error condition
- `SecurityException` - security related error condition
- `CommException` - failed communication with EIS instance
- `EISSystemException` - internal error condition in EIS instance - used if EIS instance is involved in setting state of ManagedConnection

```java
public void destroy() throws ResourceException

ManagedConnection.destroy
ManagedConnection

Throws ResourceException:

Throws IllegalStateException: 
```

destroy
void destroy() throws ResourceException

Destroys the physical connection to the underlying resource manager.

To manage the size of the connection pool, an application server can explicitly call ManagedConnection.destroy to destroy a physical connection. A resource adapter should destroy all allocated system resources for this ManagedConnection instance when the method destroy is called.

Throws:
  ResourceException - generic exception if operation failed
  IllegalStateException - illegal state for destroying connection

public void cleanup() throws ResourceException

ManagedConnection

ManagedConnection.cleanup ManagedConnection
ManagedConnection ManagedConnection cleanup

ManagedConnection ManagedConnectionManagedConnection.cleanup

ManagedConnection.cleanup

ManagedConnection cleanup

Throws ResourceException:
Throws ResourceAdapterInternalException:
Throws IllegalStateException: localtransaction

cleanup
void cleanup() throws ResourceException

Application server calls this method to force any cleanup on the ManagedConnection instance.

The method ManagedConnection.cleanup initiates a cleanup of the any client-specific state as maintained by a ManagedConnection instance. The cleanup should invalidate all connection handles that had been created using this ManagedConnection instance. Any attempt by an application component to use the connection handle after cleanup of the underlying ManagedConnection should result in an exception.

The cleanup of ManagedConnection is always driven by an application server. An application server should not invoke ManagedConnection.cleanup when there is an uncompleted transaction (associated with a ManagedConnection instance) in progress.

The invocation of ManagedConnection.cleanup method on an already cleaned-up connection should not throw an exception.

The cleanup of ManagedConnection instance resets its client specific state and prepares the connection to be put back in to a connection pool. The cleanup method should not cause resource adapter to close the physical pipe and reclaim system resources associated with the physical connection.

Throws:

- ResourceException - generic exception if operation fails
- ResourceAdapterInternalException - resource adapter internal error condition
- IllegalStateException - Illegal state for calling connection cleanup. Example - if a localtransaction is in progress that doesn't allow connection cleanup

public void associateConnection(Object connection)
throws ResourceException
ManagedConnection ManagedConnection
associateConnection

associateConnection ManagedConnection ManagedConnection

connection

Throws ResourceException: ManagedConnection
Throws IllegalStateException:
Throws ResourceAdapterInternalException:

associateConnection

void associateConnection(Object connection)
throws ResourceException

Used by the container to change the association of an application-level connection handle with a ManagedConnection instance. The container should find the right ManagedConnection instance and call the associateConnection method.

The resource adapter is required to implement the associateConnection method. The method implementation for a ManagedConnection should dissociate the connection handle (passed as a parameter) from its currently associated ManagedConnection and associate the new connection handle with itself.

Parameters:
connection - Application-level connection handle

Throws:
ResourceException - Failed to associate the connection handle with this ManagedConnection instance
IllegalStateException - Illegal state for invoking this method
ResourceAdapterInternalException - Resource adapter internal error condition
public void addConnectionEventListener(ConnectionEventListener listener)
ManagedConnection

ConnectionEventListener ManagedConnection

listener ConnectionEventListener

addConnectionEventListener

void addConnectionEventListener(ConnectionEventListener listener)

Adds a connection event listener to the ManagedConnection instance.

The registered ConnectionEventListener instances are notified of connection close and error events, also of local transaction related events on the Managed Connection.

Parameters:
listener - a new ConnectionEventListener to be registered

public void removeConnectionEventListener(ConnectionEventListener listener)
ManagedConnection

removeConnectionEventListener

void removeConnectionEventListener(ConnectionEventListener listener)
Removes an already registered connection event listener from the ManagedConnection instance.

**Parameters:**

- **listener** - already registered connection event listener to be removed

---

```java
public XAResource getXAResource() throws ResourceException
```

javax.transaction.xa.XAresource JTA

ManagedConnection XAResource

```java
return XAResource
```

**Throws:**

- ResourceException
- NotSupportedException
- ResourceAdapterInternalException

---

**getXAResource**

```java
XAResource getXAResource() throws ResourceException
```

Returns an javax.transaction.xa.XAresource instance. An application server enlists this XAResource instance with the Transaction Manager if the ManagedConnection instance is being used in a JTA transaction that is being coordinated by the Transaction Manager.

**Returns:**

- XAResource instance

**Throws:**

- ResourceException - generic exception if operation fails
- NotSupportedException - if the operation is not supported
- ResourceAdapterInternalException - resource adapter internal error condition
public **LocalTransaction** getLocalTransaction() throws **ResourceException**

`javax.resource.spi.LocalTransaction` LocalTransaction RM

return LocalTransaction

**Throws**: ResourceException:

**Throws**: NotSupportedException:

**Throws**: ResourceAdapterInternalException:

---

**getLocalTransaction**

**LocalTransaction** getLocalTransaction()

throws **ResourceException**

Returns an `javax.resource.spi.LocalTransaction` instance. The LocalTransaction interface is used by the container to manage local transactions for a RM instance.

**Returns**: LocalTransaction instance

**Throws**: **ResourceException** - generic exception if operation fails

NotSupportedException - if the operation is not supported

ResourceAdapterInternalException - resource adapter internal error condition

---

public **ManagedConnectionMetaData** getMetaData() throws **ResourceException**

**EIS ManagedConnectionMetaData**

ManagedConenction EIS

return ManagedConnectionMetaData

**Throws**: ResourceException:

**Throws**: NotSupportedException:

**Throws**: NotSupportedIOException:
getMetaData

ManagedConnectionMetaData getMetaData() throws ResourceException

Gets the metadata information for this connection's underlying EIS resource manager instance. The ManagedConnectionMetaData interface provides information about the underlying EIS instance associated with the ManagedConnection instance.

Returns:
ManagedConnectionMetaData instance

Throws:
ResourceException - generic exception if operation fails
NotSupportedException - if the operation is not supported

public void setLogWriter(java.io.PrintWriter out) throws ResourceException

ManagedConnection writer

writer ManagedConnection
ManagedConnection

ManagedConnection ManagedConnectionFactory
writer setLogWriter ManagedConnection
writer/

out

Throws ResourceException:

Throws ResourceAdapterInternalException:

setLogWriter
void setLogWriter(PrintWriter out)
throws ResourceException

Sets the log writer for this ManagedConnection instance.

The log writer is a character output stream to which all logging and tracing messages for this ManagedConnection instance will be printed. Application Server manages the association of output stream with the ManagedConnection instance based on the connection pooling requirements.

When a ManagedConnection object is initially created, the default log writer associated with this instance is obtained from the ManagedConnectionFactory. An application server can set a log writer specific to this ManagedConnection to log/trace this instance using setLogWriter method.

Parameters:
   out - Character Output stream to be associated

Throws:
   ResourceException - generic exception if operation fails
   ResourceAdapterInternalException - resource adapter related error condition

public java.io.PrintWriter getLogWriter() throws
ResourceException

ManagedConnection writer

ManagedConnection writer
ConnectionManager ManagedConnection

ManagedConnection writer
ManagedConnectionFactory writer

return ManagedConnection

Throws ResourceException:
getLogWriter

`PrintWriter getLogWriter()` throws `ResourceException`

Gets the log writer for this ManagedConnection instance.

The log writer is a character output stream to which all logging and tracing messages for this ManagedConnection instance will be printed. ConnectionManager manages the association of output stream with the ManagedConnection instance based on the connection pooling requirements.

The Log writer associated with a ManagedConnection instance can be one set as default from the ManagedConnectionFactory (that created this connection) or one set specifically for this instance by the application server.

**Returns:**
Character output stream associated with this Managed-Connection instance

**Throws:**
`ResourceException` - generic exception if operation fails
javax.resource.spi

**Interface ManagedConnectionFactory**

**All Superinterfaces:**

[Serializable]

```java
public interface ManagedConnectionFactory
extends Serializable

Implements: java.io.Serializable
```

**ManagedConnectionFactory ManagedConnection EIS**

ManagedConnectionFactory ManagedConnectionFactory

JavaBean

```java
version 0.6
See also javax.resource.spi.ManagedConnection
```

ManagedConnectionFactory instance is a factory of both ManagedConnection and EIS-specific connection factory instances. This interface supports connection pooling by providing methods for matching and creation of ManagedConnection instance. A ManagedConnectionFactory instance is required to be a JavaBean.

**Version:**

0.6

**Author:**
Rahul Sharma

**See Also:**
ManagedConnection

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createConnectionFactory()</code></td>
<td>Creates a Connection Factory instance.</td>
</tr>
<tr>
<td><code>createConnectionFactory(ConnectionManager cxManager)</code></td>
<td></td>
</tr>
<tr>
<td><code>Object</code></td>
<td></td>
</tr>
</tbody>
</table>


<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createManagedConnection(Subject subject, ConnectionRequestInfo cxRequestInfo)</code></td>
<td>Creates a new physical connection to the underlying EIS resource manager.</td>
</tr>
<tr>
<td><code>equals(Object other)</code></td>
<td>Check if this ManagedConnectionFactory is equal to another ManagedConnectionFactory.</td>
</tr>
<tr>
<td><code>getLogWriter()</code></td>
<td>Get the log writer for this ManagedConnectionFactory instance.</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td>Returns the hash code for the ManagedConnectionFactory.</td>
</tr>
<tr>
<td><code>matchManagedConnections(Set connectionSet, Subject subject, ConnectionRequestInfo cxRequestInfo)</code></td>
<td>Returns a matched connection from the candidate set of connections.</td>
</tr>
<tr>
<td><code>setLogWriter(PrintWriter out)</code></td>
<td>Set the log writer for this ManagedConnectionFactory instance.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public Object createConnectionFactory(ConnectionManager cxManager) throws ResourceException {
    Connection Factory Connection Factory
    return EIS ConnectionManager
    javax.resource.cci.ConnectionFactory
   Throws ResourceException:
```
createConnectionFactory

**Object** createConnectionFactory(**ConnectionManager** cxManager) throws **ResourceException**

Creates a Connection Factory instance. The Connection Factory instance gets initialized with the passed ConnectionManager. In the managed scenario, ConnectionManager is provided by the application server.

**Parameters:**

- **cxManager** - ConnectionManager to be associated with created EIS connection factory instance

**Returns:**

- EIS-specific Connection Factory instance or javax.resource.cci.ConnectionFactory instance

**Throws:**

- **ResourceException** - Generic exception
- **ResourceAdapterInternalException** - Resource adapter related error condition

---

```java
class ConnectionFactory
```

```java
public Object createConnectionFactory() throws ResourceException
```

**Connection Factory Connection Factory ConnectionManager**

```java
return EIS Connection Factory
javax.resource.cci.ConnectionFactory
```

**Throws**

- **ResourceException:**

**Throws**

- **ResourceAdapterInternalException:**

createConnectionFactory

**Object** createConnectionFactory()
Throws: ResourceException - Generic exception
        ResourceAdapterInternalException - Resource adapter related error condition

public ManagedConnection createManagedConnection(javax.security.auth.Subject subject, ConnectionRequestInfo cxRequestInfo) throws ResourceException

ManagedConnectionFactory subject
ConnectionRequestInfo cxRequestInfo ResourceAdapter

Throws: ResourceException:
        SecurityException:
        ResourceAllocationException:
        ResourceAdapterInternalException:
        EISSystemException: EIS

return ManagedConnection

createManagedConnection

ManagedConnection createManagedConnection(Subject subject,
Creates a new physical connection to the underlying EIS resource manager.

ManagedConnectionFactory uses the security information (passed as Subject) and additional ConnectionRequestInfo (which is specific to ResourceAdapter and opaque to application server) to create this new connection.

**Parameters:**
- `subject` - Caller's security information
- `cxRequestInfo` - Additional resource adapter specific connection request information

**Returns:**
- ManagedConnection instance

**Throws:**
- `ResourceException` - generic exception
- `SecurityException` - security related error
- `ResourceAllocationException` - failed to allocate system resources for connection request
- `ResourceAdapterInternalException` - resource adapter related error condition
- `EISSystemException` - internal error condition in EIS instance

```java
public ManagedConnection matchManagedConnections(java.util.Set<E> connectionSet, javax.security.auth.Subject subject, ConnectionRequestInfo cxRequestInfo) throws ResourceException
```

```
ManagedConnectionFactory Subject ConnectionRequestInfo Connector
ManagedConnection
```
return ManagedConnection null

matchManagedConnections

ManagedConnection matchManagedConnections(Set connectionSet, Subject subject, ConnectionRequestInfo cxRequestInfo) throws ResourceException

Returns a matched connection from the candidate set of connections.

ManagedConnectionFactory uses the security info (as in Subject) and information provided through ConnectionRequestInfo and additional Resource Adapter specific criteria to do matching. Note that criteria used for matching is specific to a resource adapter and is not prescribed by the Connector specification.

This method returns a ManagedConnection instance that is the best match for handling the connection allocation request.

Parameters:
- connectionSet - candidate connection set
- subject - caller's security information
- cxRequestInfo - additional resource adapter specific connection request information

Returns:
ManagedConnection if resource adapter finds an acceptable match otherwise null

Throws:
ResourceException - generic exception
**SecurityException** - security related error

**ResourceAdapterInternalException** - resource adapter related error condition

**NotSupportedException** - if operation is not supported

---

```java
public void setLogWriter(java.io.PrintWriter out) throws ResourceException
ManagedConnectionFactory writer

writer ManagedConnectionFactory

ApplicationServer ManagedConnectionFactory
ManagedConnectionFactory writer null writer
ManagedConnectionFactory ManagedConnectionFactory

ManagedConnectionFactory ManagedConnection """"
writer ApplicationServer ManagedConnectionFactory.setLogWriter
ManagedConnection
　　out PrintWriter
　　Thrown ResourceException:
　　Thrown ResourceAdapterInternalException:

setLogWriter

void setLogWriter(PrintWriter out)
　　throws ResourceException

Set the log writer for this ManagedConnectionFactory instance.

The log writer is a character output stream to which all logging and tracing messages for this ManagedConnectionFactory instance will be printed.

ApplicationServer manages the association of output stream with the ManagedConnectionFactory. When a ManagedConnectionFactory object is created the log writer is initially null, in other words, logging is disabled.
```
Once a log writer is associated with a ManagedConnectionFactory, logging and tracing for ManagedConnectionFactory instance is enabled.

The ManagedConnection instances created by ManagedConnectionFactory "inherits" the log writer, which can be overridden by ApplicationServer using ManagedConnection.setLogWriter to set ManagedConnection specific logging and tracing.

**Parameters:**
- `out` - PrintWriter - an out stream for error logging and tracing

**Throws:**
- `ResourceException` - generic exception
- `ResourceAdapterInternalException` - resource adapter related error condition

```java
public java.io.PrintWriter getLogWriter() throws ResourceException

ManagedConnectionFactory writer

ApplicationServer ManagedConnectionFactory writer null

return PrintWriter

Throws ResourceException:
```

**getLogWriter**

`PrintWriter getLogWriter()`

`throws ResourceException`

Get the log writer for this ManagedConnectionFactory instance.

The log writer is a character output stream to which all logging and tracing messages for this ManagedConnectionFactory instance will
be printed

ApplicationServer manages the association of output stream with the ManagedConnectionFactory. When a ManagedConnectionFactory object is created the log writer is initially null, in other words, logging is disabled.

**Returns:**
- PrintWriter

**Thows:**
- ResourceException - generic exception

```java
public int hashCode()
{
    return ManagedConnectionFactory
}
```

**hashCode**

```java
int hashCode()
{
    Returns the hash code for the ManagedConnectionFactory

    ** Overrides:**
    - hashCode in class Object

    ** Returns:**
    hash code for the ManagedConnectionFactory
}
```

```java
public boolean equals(Object other)
{
    return true
}
```

**equals**
boolean equals(Object other)

Check if this ManagedConnectionFactory is equal to another ManagedConnectionFactory.

Overrides:
   equals in class Object

Returns:
   true if two instances are equal
javax.resource.spi Interface ManagedConnectionMetaData

public interface ManagedConnectionMetaData

ManagedConnectionMetaData ManagedConnection EIS

ManagedConnection.getMetaData ManagedConnectionMetaData

version 0.8
See also javax.resource.spi.ManagedConnection

The ManagedConnectionMetaData interface provides information about the underlying EIS instance associated with a ManagedConnection instance. An application server uses this information to get runtime information about a connected EIS instance.

The method ManagedConnection.getMetaData returns a ManagedConnectionMetaData instance.

Version:
0.8
Author:
Rahul Sharma
See Also:
ManagedConnection

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getEISProductName()</td>
<td>Returns Product name of the underlying EIS instance</td>
</tr>
</tbody>
</table>
String `getEISProductName()`

Returns product name of the underlying EIS instance connected through the ManagedConnection.

```
public String getEISProductName() throws ResourceException
```

```
ManagedConnection EIS
```

```
return EIS
```

### getEISProductName

**Returns:**

Product name of the EIS instance.

**Throws:**

ResourceException

---

Int `getMaxConnections()`

Returns maximum limit on number of active concurrent connections that an EIS instance can support across client processes.

```
int getMaxConnections()
```

**Returns:**

Maximum limit on number of active concurrent connections.

---

String `getUserName()`

Returns name of the user associated with the ManagedConnection instance.

```
String getUserName()
```

**Returns:**

Name of the user associated with the ManagedConnection instance.

---

Method Detail

```
public String getEISProductVersion() throws ResourceException
```

```
ManagedConnection EIS
```

```
return EIS
```

**getEISProductVersion**

**Returns:**

Product version of the underlying EIS instance connected through the ManagedConnection.

**Throws:**

ResourceException
**ResourceException**

**ManagedConnection EIS**

```
ManagedConnectionManagedConnection
```

getEISProductVersion

```java
String getEISProductVersion()
```

returns `ResourceException`

Returns product version of the underlying EIS instance connected through the ManagedConnection.

**Returns:**
Product version of the EIS instance

**Throws:**
`ResourceException`

--

**public int getMaxConnections() throws ResourceException**

```
public int getMaxConnections()
```

returns `0`

```
return 0
```

getMaxConnections

```java
int getMaxConnections()
```

returns `ResourceException`

Returns maximum limit on number of active concurrent connections that an EIS instance can support across client processes. If an EIS instance does not know about (or does not have) any such limit, it returns a 0.

**Returns:**
Maximum limit for number of active concurrent connections

**Throws:**
`ResourceException`
public String getUserName() throws ResourceException
ManagedConnection EIS

return

getUserName

String getUserName()
throws ResourceException

Returns name of the user associated with the ManagedConnection instance. The name corresponds to the resource principal under whose whose security context, a connection to the EIS instance has been established.

Returns:
name of the user

Throws:
ResourceException

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature

PS:
javax.management.j2ee Interface Management

All Superinterfaces:
   EJBObjec, Remote

```java
public interface Management
extends EJBObjec
Implements: EJBObjec
```

Management API J2EE Management EJB (MEJB)

The Management interface provides the APIs to navigate and manipulate managed objects. The J2EE Management EJB component (MEJB) must implement this as its remote interface.

**Author:**
   Hans Hrasna

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ObjectName</code></td>
<td><code>getAttribute(ObjectName name, String attribute)</code></td>
<td>Gets the value of a specific attribute of a named managed object.</td>
</tr>
<tr>
<td><code>AttributeList</code></td>
<td><code>getAttributes(ObjectName name, String[] attributes)</code></td>
<td>Enables the values of several attributes of a named managed object.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getDefaultDomain()</code></td>
<td>Returns the default domain name of this MEJB.</td>
</tr>
<tr>
<td><code>ListenerRegistration</code></td>
<td><code>getListenerRegistry()</code></td>
<td>Returns the listener registry implementation for this MEJB.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getMBeanCount()</code></td>
<td>Returns the number of managed objects registered in the MEJB.</td>
</tr>
<tr>
<td><code>getMBeanInfo(ObjectName name)</code></td>
<td>This method discovers the attributes and operations that a managed object exposes for management.</td>
</tr>
<tr>
<td><code>invoke(ObjectName name, String operationName, Object[] params, String[] signature)</code></td>
<td>Invokes an operation on a managed object.</td>
</tr>
<tr>
<td><code>isRegistered(ObjectName name)</code></td>
<td>Checks whether a managed object, identified by its object name, is already registered with the MEJB.</td>
</tr>
<tr>
<td><code>getAttributeSetNames(ObjectName name, QueryExp query)</code></td>
<td>Gets the names of managed objects controlled by the MEJB.</td>
</tr>
<tr>
<td><code>setAttribute(ObjectName name, Attribute attribute)</code></td>
<td>Sets the value of a specific attribute of a named managed object.</td>
</tr>
<tr>
<td><code>setAttributes(ObjectName name, AttributeList attributes)</code></td>
<td>Sets the values of several attributes of a named managed object.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface `javax.ejb.EJBOBJECT`

- `getEJBHome`, `getHandle`, `getPrimaryKey`, `isIdentical`, `remove`

### Method Detail

```java
public java.util.Set<E>
queryNames(javax.management.ObjectName name,
javax.management.QueryExp query) throws
java.rmi.RemoteException
MEJB
```
J2EEObjectNames

```java
public ObjectName queryNames(ObjectName name, QueryExp query)
throws RemoteException
```

The `queryNames` method gets the names of managed objects controlled by the MEJB. This method enables any of the following to be obtained: The names of all managed objects, the names of a set of managed objects specified by pattern matching on the `ObjectName`, a specific managed object name (equivalent to testing whether a managed object is registered). When the object name is null or no domain and key properties are specified, all objects are selected. It returns the set of J2EEObjectNames for the managed objects selected.

**Parameters:**
- `name` - The object name pattern identifying the managed objects to be retrieved. If null or no domain and key properties are specified, all the managed objects registered will be retrieved.

**Returns:**
- A set containing the ObjectNames for the managed objects selected. If no managed object satisfies the query, an empty set is returned.

**Throws:**
- `RemoteException` - A communication exception occurred during the execution of a remote method call

---

public boolean isRegistered(javax.management.ObjectName name) throws java.rmi.RemoteException

MEJB
isRegistered

boolean isRegistered(ObjectName name) throws RemoteException

Checks whether a managed object, identified by its object name, is already registered with the MEJB.

Parameters:
   name - The object name of the managed object to be checked.

Returns:
   True if the managed object is already registered in the MEJB, false otherwise.

Throws:
   RemoteException - A communication exception occurred during the execution of a remote method call

public Integer getMBeanCount() throws java.rmi.RemoteException

MEJB

Throws java.rmi.RemoteException:

getMBeanCount

Integer getMBeanCount() throws RemoteException

   Returns the number of managed objects registered in the MEJB.

   Throws:
      RemoteException - A communication exception occurred during
public javax.management.MBeanInfo
getMBeanInfo(javax.management.ObjectName name)
throws javax.management.IntrospectionException,
javax.management.InstanceNotFoundException,
javax.management.ReflectionException,
java.rmi.RemoteException

name
return MBeanInfo
Throws javax.management.IntrospectionException: introspection
Throws javax.management.InstanceNotFoundException:
Throws javax.management.ReflectionException:
Throws java.rmi.RemoteException:

getMBeanInfo
MBeanInfo getMBeanInfo(ObjectName name)
throws IntrospectionException,
InstanceNotFoundException,
ReflectionException,
RemoteException

This method discovers the attributes and operations that a managed object exposes for management.

Parameters:
name - The name of the managed object to analyze

Returns:
An instance of MBeanInfo allowing the retrieval of all attributes and operations of this managed object.

Throws:
IntrospectionException - An exception occurs during introspection.
**InstanceNotFoundException** - The managed object specified is not found.
**ReflectionException** - An exception occurred when trying to perform reflection on a managed object
**RemoteException** - A communication exception occurred during the execution of a remote method call

---

```java
public Object getAttribute(javax.management.ObjectName name, String attribute) throws
java.management.MBeanException, java.management.AttributeNotFoundException,
java.management.InstanceNotFoundException, java.management.ReflectionException,
java.rmi.RemoteException

    name
    attribute String
    return

Throws javax.management.AttributeNotFoundException:
Throws javax.management.MBeanException:
Throws javax.management.InstanceNotFoundException: MEJB
    java.management.ReflectionException: Dynamic MBean
        getAttribute
    java.rmi.RemoteException:
```

**getAttribute**

```java
Object getAttribute(ObjectName name, String attribute) throws
MBeanException,
AttributeNotFoundException, InstanceNotFoundException, ReflectionException,
RemoteException
```

Gets the value of a specific attribute of a named managed object.
The managed object is identified by its object name.

**Parameters:**

- `name` - The object name of the managed object from which the attribute is to be retrieved.
- `attribute` - A String specifying the name of the attribute to be retrieved.

**Returns:**

The value of the retrieved attribute.

**Throws:**

- `AttributeNotFoundException` - The attribute specified is not accessible in the managed object.
- `MBeanException` - Wraps an exception thrown by the managed object's getter.
- `InstanceNotFoundException` - The managed object specified is not registered in the MEJB.
- `ReflectionException` - An exception occurred when trying to invoke the `getAttribute` method of a Dynamic MBean.
- `RemoteException` - A communication exception occurred during the execution of a remote method call.

```java

name
attributes
return

Throws  javax.management.InstanceNotFoundException: MEJB

Throws  javax.management.ReflectionException: Dynamic MBean

Throws  java.rmi.RemoteException:
```
getAttributes

AttributeList getAttributes(ObjectName name, String[] attributes)
throws InstanceNotFoundException,
ReflectionException,
RemoteException

Enables the values of several attributes of a named managed object. The managed object is identified by its object name.

Parameters:
  name - The object name of the managed object from which the attributes are retrieved.
  attributes - A list of the attributes to be retrieved.

Returns:
  The list of the retrieved attributes.

Throws:
  InstanceNotFoundException - The managed object specified is not registered in the MEJB.
  ReflectionException - An exception occurred when trying to invoke the getAttributes method of a Dynamic MBean.
  RemoteException - A communication exception occurred during the execution of a remote method call

public void setAttribute(javax.management.ObjectName name, javax.management.Attribute attribute) throws
javax.management.InstanceNotFoundException,
javax.management.AttributeNotFoundException,
javax.management.InvalidAttributeValueException,
javax.management.MBeanException,
javax.management.ReflectionException,
java.rmi.RemoteException

  name
setAttribute

void setAttribute(ObjectName name, Attribute attribute)
throws InstanceNotFoundException, AttributeNotFoundException, InvalidAttributeValueException, MBeanException, ReflectionException, RemoteException

Sets the value of a specific attribute of a named managed object. The managed object is identified by its object name.

Parameters:

name - The name of the managed object within which the attribute is to be set.
attribute - The identification of the attribute to be set and the value it is to be set to.

Throws:

InstanceNotFoundException - The managed object specified is not registered in the MEJB.
AttributeNotFoundException - The attribute specified is not accessible in the managed object.
InvalidAttributeValueException - The value specified for the attribute is not valid.
MBeanException - Wraps an exception thrown by the managed object's setter.
ReflectionException - An exception occurred when trying to invoke the setAttribute method of a Dynamic MBean.
RemoteException - A communication exception occurred during the execution of a remote method call

```java

name
attributes
return
Throws javax.management.InstanceNotFoundException: MEJB
Throwsjavax.management.ReflectionException: Dynamic MBean
Throws java.rmi.RemoteException:

setAttributes

AttributeList setAttributes(ObjectName name, AttributeList attributes) throws InstanceNotFoundException, ReflectionException, RemoteException

Sets the values of several attributes of a named managed object. The managed object is identified by its object name.

Parameters:
name - The object name of the managed object within which the attributes are to be set.
attributes - A list of attributes: The identification of the attributes to be set and the values they are to be set to.

Returns:
```
The list of attributes that were set, with their new values.

**Throws:**
- `InstanceNotFoundException` - The managed object specified is not registered in the MEJB.
- `ReflectionException` - An exception occurred when trying to invoke the `setAttributes` method of a Dynamic MBean.
- `RemoteException` - A communication exception occurred during the execution of a remote method call.

```java
public Object invoke(javax.management.ObjectName name, String operationName, Object[] params, String[] signature) throws
java.management.InstanceNotFoundException, java.management.MBeanException, java.management.ReflectionException, java.rmi.RemoteException
```

```
name
operationName
params
signature
return
```

**Throws**
- `javax.management.InstanceNotFoundException`: MEJB
- `javax.management.MBeanException`: 
- `javax.management.ReflectionException`: `java.lang.Exception`
- `java.rmi.RemoteException`: 

**invoke**

```java
Object invoke(ObjectName name, String operationName, Object[] params, String[] signature)
```
Invokes an operation on a managed object.

Parameters:
- **name** - The object name of the managed object on which the method is to be invoked.
- **operationName** - The name of the operation to be invoked.
- **params** - An array containing the parameters to be set when the operation is invoked.
- **signature** - An array containing the signature of the operation.

The class objects will be loaded using the same class loader as the one used for loading the managed object on which the operation was invoked.

Returns:
The object returned by the operation, which represents the result of invoking the operation on the managed object specified.

Throws:
- **InstanceNotFoundException** - The managed object specified is not registered in the MEJB.
- **MBeanException** - Wraps an exception thrown by the managed object's invoked method.
- **ReflectionException** - Wraps a `java.lang.Exception` thrown while trying to invoke the method.
- **RemoteException** - A communication exception occurred during the execution of a remote method call.

```java
public String getDefaultDomain() throws java.rmi.RemoteException,
                             MEJB

    Throws java.rmi.RemoteException:
```

```
getDefaultDomain
```
String getDefaultDomain() throws RemoteException

Returns the default domain name of this MEJB.

Throws:
   RemoteException - A communication exception occurred during the execution of a remote method call

public ListenerRegistration getListenerRegistry() throws java.rmi.RemoteException
MEJB

   return javax.management.j2ee.ListenerRegistration

Throws java.rmi.RemoteException:

getListenerRegistry

ListenerRegistration getListenerRegistry() throws RemoteException

Returns the listener registry implementation for this MEJB. The listener registry implements the methods that enable clients to add and remove event notification listeners managed objects

Returns:
   An implementation of javax.management.j2ee.ListenerRegistration

Throws:
   RemoteException - A communication exception occurred during the execution of a remote method call
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.management.j2ee  Interface ManagementHome

All Superinterfaces:
   EJBHome, Remote

public interface ManagementHome extends EJBHome

Implements: EJBHome

J2EE Management EJB (MEJB) home J2EE create()

The required home interface for the J2EE Management EJB component (MEJB). A J2EE client must be able to create a compliant session object using the specified create() method.

Author:
   Hans Hrasna

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
</tr>
<tr>
<td>create()</td>
</tr>
<tr>
<td>Creates an MEJB session object which provides access to the J2EE Management Model</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.ejb.EJBHome
getEJBMetaData, getHomeHandle, remove, remove

Method Detail
public Management create() throws CreateException, java.rmi.RemoteException

MEJB  J2EE Management Model

    return javax.management.j2ee.Management MEJB
    Throws CreateException: EJB
    Throws java.rmi.RemoteException:

create

Management create() throws CreateException, RemoteException

Creates an MEJB session object which provides access to the J2EE Management Model

Returns:
    An MEJB session object which implements javax.management.j2ee.Management

Throws:
    CreateException - Indicates a failure to create the EJB object
    RemoteException - A communication exception occurred during the execution of a remote method call
    CreateException

Overview  Package  Tree  Deprecated  Index  Help
javax.persistence  Annotation Type ManyToMany

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface ManyToMany

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Collection  generic

OneToMany    ManyToMany

1

Customer

@ManyToMany
@JoinTable(name="CUST_PHONES")
public Set getPhones() { return phones; }

PhoneNumber

@ManyToMany(mappedBy="phones")
public Set getCustomers() { return customers; }

2

Customer

@ManyToMany(targetEntity=com.acme.PhoneNumber.class)
public Set getPhones() { return phones; }

PhoneNumber

@ManyToMany(targetEntity=com.acme.Customer.class, mappedBy="phones")
public Set getCustomers() { return customers; }

3

Customer
@ManyToMany
@JoinTable(name="CUST_PHONE",
    joinColumns=
        @JoinColumn(name="CUST_ID", referencedColumnName="ID"),
    inverseJoinColumns=
        @JoinColumn(name="PHONE_ID", referencedColumnName="ID")
)
public Set getPhones() { return phones; }

PhoneNumberClass
@ManyToMany(mappedBy="phones")
public Set getCustomers() { return customers; }

since Java Persistence 1.0

Defines a many-valued association with many-to-many multiplicity. If the Collection is defined using generics to specify the element type, the associated target entity class does not need to be specified; otherwise it must be specified.

Every many-to-many association has two sides, the owning side and the non-owning, or inverse, side. The join table is specified on the owning side. If the association is bidirectional, either side may be designated as the owning side.

The same annotation elements for the OneToMany annotation apply to the ManyToMany annotation.

Example 1:
In Customer class:

@ManyToMany
@JoinTable(name="CUST_PHONES")
public Set getPhones() { return phones; }

In PhoneNumber class:

@ManyToMany(mappedBy="phones")
public Set getCustomers() { return customers; }

Example 2:
In Customer class:

```java
@ManyToMany(targetEntity=com.acme.PhoneNumber.class)
public Set getPhones() { return phones; }
```

In PhoneNumber class:

```java
@ManyToMany(targetEntity=com.acme.Customer.class, mappedBy="phones")
public Set getCustomers() { return customers; }
```

Example 3:

In Customer class:

```java
@ManyToMany
@JoinTable(name="CUST_PHONE",
    joinColumns=
        @JoinColumn(name="CUST_ID", referencedColumnName="ID"),
    inverseJoinColumns=
        @JoinColumn(name="PHONE_ID", referencedColumnName="ID")
)
public Set getPhones() { return phones; }
```

In PhoneNumber class:

```java
@ManyToMany(mappedBy="phones")
public Set getCustomers() { return customers; }
```

Since:
Java Persistence 1.0

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CascadeType[]</td>
<td>(Optional) The operations that must be cascaded to the target of the association.</td>
</tr>
<tr>
<td>FetchType</td>
<td>(Optional) Whether the association should be lazily loaded or must be eagerly fetched.</td>
</tr>
<tr>
<td>String mappedBy</td>
<td>The field that owns the relationship.</td>
</tr>
</tbody>
</table>
abstract public Class<T> targetEntity()
Java generic Collection
generic Collection
targetEntity

public abstract Class targetEntity

(Optional) The entity class that is the target of the association. Optional only if the collection property is defined using Java generics. Must be specified otherwise.

Defaults to the parameterized type of the collection when defined using generics.

Default:
void.class

abstract public CascadeType[] cascade()
(Optional) The operations that must be cascaded to the target of the association.

Defaults to no operations being cascaded.

Default:

```java
{}
```

---

**abstract public FetchType fetch()**

**fetch**

```java
public abstract FetchType fetch()
```

(Optional) Whether the association should be lazily loaded or must be eagerly fetched. The **EAGER** strategy is a requirement on the persistenceprovider runtime that the associatedentities must be eagerly fetched. The **LAZY** strategy is a hint to the persistence provider runtime.

Default:

```java
LAZY
```

---

**abstract public String mappedBy()**

**mappedBy**

```java
public abstract String mappedBy()
```

The field that owns the relationship. Required unless the relationship is unidirectional.

Default:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

**PreV CLASS** | **NEXT CLASS**
**SUMMARY: REQUIRED** | **OPTIONAL**

**FRAMES** | **NO FRAMES**
**DETAIL: ELEMENT**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.persistence  Annotation Type ManyToOne

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface ManyToOne

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@ManyToOne(optional=false)
@JoinColumn(name="CUST_ID", nullable=false, updatable=false)
public Customer getCustomer() { return customer; }

since Java Persistence 1.0

This annotation defines a single-valued association to another entity class that has many-to-one multiplicity. It is not normally necessary to specify the target entity explicitly since it can usually be inferred from the type of the object being referenced.

Example:

@ManyToOne(optional=false)
@JoinColumn(name="CUST_ID", nullable=false, updatable=false)
public Customer getCustomer() { return customer; }

Since: Java Persistence 1.0

Optional Element Summary
<table>
<thead>
<tr>
<th></th>
<th>cascade</th>
<th>(Optional) The operations that must be cascaded to the target of the association.</th>
</tr>
</thead>
<tbody>
<tr>
<td>FetchType</td>
<td>fetch</td>
<td>(Optional) Whether the association should be lazily loaded or must be eagerly fetched.</td>
</tr>
<tr>
<td>boolean</td>
<td>optional</td>
<td>(Optional) Whether the association is optional.</td>
</tr>
<tr>
<td>Class</td>
<td>targetEntity</td>
<td>(Optional) The entity class that is the target of the association.</td>
</tr>
</tbody>
</table>

abstract public Class<T> targetEntity()

targetEntity

public abstract Class targetEntity

(Optional) The entity class that is the target of the association.

Defaults to the type of the field or property that stores the association.

Default:

void.class

abstract public CascadeType[] cascade()
cascade

public abstract CascadeType[] cascade

(Optional) The operations that must be cascaded to the target of the association.

By default no operations are cascaded.

**Default:**

```
{}
```

---

abstract public FetchType fetch()

```
EAGER  LAZY
```

fetch

public abstract FetchType fetch

(Optional) Whether the association should be lazily loaded or must be eagerly fetched. The **EAGER** strategy is a requirement on the persistence provider runtime that the associated entity must be eagerly fetched. The **LAZY** strategy is a hint to the persistence provider runtime.

**Default:**

```
EAGER
```

---

abstract public boolean optional()

false null

optional
public abstract boolean optional

(Optional) Whether the association is optional. If set to false then a non-null relationship must always exist.

Default: true
javax.el  **Class MapELResolver**

 java.lang.Object  
  ↓  javax.el.ELResolver  
  ↓  javax.el.MapELResolver

public class MapELResolver

extends ELResolver

**Extends:** ELResolver

java.util.Map

java.util.Map  base

 #isReadOnly  true#setValue  PropertyNotWritableException

ELResolver  CompositeELResolver  ELResolver

Javadoc

**since**  JSP 2.1

**See**  javax.el.CompositeELResolver, javax.el.ELResolver, java.util.Map

**also**  java.util.Map

Defines property resolution behavior on instances of Map.

This resolver handles base objects of type java.util.Map. It accepts any object as a property and uses that object as a key in the map. The resulting value is the value in the map that is associated with that key.

This resolver can be constructed in read-only mode, which means that isReadOnly will always return true and setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always throw PropertyNotWritableException.

ELResolvers are combined together using CompositeELResolver, to
define rich semantics for evaluating an expression. See the javadocs for \texttt{ELResolver} for details.

\textbf{Since:}  
JSP 2.1

\textbf{See Also:}  
\texttt{CompositeELResolver, ELResolver, Map}

---

\section*{Field Summary}

\begin{tabular}{|l|}
\hline
Fields inherited from class \texttt{javax.el.ELResolver} \\
\texttt{RESOLVABLE_AT_DESIGN_TIME, TYPE} \\
\hline
\end{tabular}

\section*{Constructor Summary}

\begin{tabular}{|l|}
\hline
\texttt{MapELResolver()}  \\
Creates a new read/write \texttt{MapELResolver}. \\
\hline
\texttt{MapELResolver(boolean isReadOnly)}  \\
Creates a new \texttt{MapELResolver} whose read-only status is determined by the given parameter. \\
\hline
\end{tabular}

\section*{Method Summary}

\begin{tabular}{|l|}
\hline
\texttt{Class<?>}  \\
\texttt{getCommonPropertyType(ELContext context, Object base)}  \\
If the base object is a map, returns the most general type that this resolver accepts for the property argument. \\
\hline
\texttt{Iterator<FeatureDescriptor>}  \\
\texttt{getFeatureDescriptors(ELContext context, Object base)}  \\
If the base object is a map, returns an Iterator containing the set of keys available in the Map. \\
\hline
\texttt{Class<?>}  \\
\texttt{getType(ELContext context, Object base, Object property)}  \\
If the base object is a map, returns the
Object getValue(ELContext context, Object base, Object property)

If the base object is a map, returns the value associated with the given key, as specified by the property argument.

boolean isReadOnly(ELContext context, Object base, Object property)

If the base object is a map, returns whether a call to setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always fail.

void setValue(ELContext context, Object base, Object property, Object val)

If the base object is a map, attempts to set the value associated with the given key, as specified by the property argument.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public MapELResolver()
/

MapELResolver

public MapELResolver()

Creates a new read/write MapELResolver.
public MapELResolver(boolean isReadOnly)

MapELResolver

public MapELResolver(boolean isReadOnly)

Creates a new MapELResolver whose read-only status is determined by the given parameter.

Parameters:
isReadOnly - true if this resolver cannot modify maps; false otherwise.

Method Detail

public Class<T> getType(ELContext context, Object base, Object property)

base

base Map ELContext propertyResolved true
true

base Map Object.class Map

context
base Map base
property

return ELContext propertyResolved true

Throws NullPointerException: context null

Throws ELEException: cause
**getType**

```java
public Class<?> getType(ELContext context, Object base, Object property)
```

If the base object is a map, returns the most general acceptable type for a value in this map.

If the base is a `Map`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

Assuming the base is a `Map`, this method will always return `Object.class`. This is because `Maps` accept any object as the value for a given key.

**Specified by:**
- `getType` in class `ELResolver`

**Parameters:**
- `context` - The context of this evaluation.
- `base` - The map to analyze. Only bases of type `Map` are handled by this resolver.
- `property` - The key to return the acceptable type for. Ignored by this resolver.

**Returns:**
- If the `propertyResolved` property of `ELContext` was set to `true`, then the most general acceptable type; otherwise undefined.

**Throws:**
- `NullPointerException` - if context is null
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public Object getValue(ELContext context, Object base, Object property)
```

```java
base property null
```
getValue

public Object getValue(ELContext context, Object base, Object property)

If the base object is a map, returns the value associated with the given key, as specified by the property argument. If the key was not found, null is returned.

If the base is a Map, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Just as in Map.get(java.lang.Object), just because null is returned doesn’t mean there is no mapping for the key; it’s also possible that the Map explicitly maps the key to null.

Specified by:
getValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - The map to be analyzed. Only bases of type Map are
handled by this resolver.

property - The key whose associated value is to be returned.

Returns:
If the propertyResolved property of ELContext was set to true, then the value associated with the given key or null if the key was not found. Otherwise, undefined.

Throws:

ClassCastException - if the key is of an inappropriate type for this map (optionally thrown by the underlying Map).

NullPointerException - if context is null, or if the key is null and this map does not permit null keys (the latter is optionally thrown by the underlying Map).

ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

public void setValue(ELContext context, Object base, Object property, Object val)

Base                Map            ELContext    propertyResolved    true

PropertyNotWritableException

java.util.Collections#unmodifiableMap              Map

PropertyNotWritableException Collection API

Map base

collection

base

property

val

Throws ClassCastException:

Throws NullPointerException: null null null

Throws IllegalArgumentException:
Throws: `ELException`: cause

Throws: `PropertyNotWritableException`:

**setValue**

```java
public void setValue(ELContext context,
                     Object base,
                     Object property,
                     Object val)
```

If the base object is a map, attempts to set the value associated with the given key, as specified by the `property` argument.

If the base is a `Map`, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller can safely assume no value was set.

If this resolver was constructed in read-only mode, this method will always throw `PropertyNotWritableException`.

If a `Map` was created using `Collections.unmodifiableMap(java.util.Map)`, this method must throw `PropertyNotWritableException`. Unfortunately, there is no Collections API method to detect this. However, an implementation can create a prototype unmodifiable `Map` and query its runtime type to see if it matches the runtime type of the base object as a workaround.

**Specified by:**

`setValue` in class `ELResolver`

**Parameters:**

- `context` - The context of this evaluation.
- `base` - The map to be modified. Only bases of type `Map` are handled by this resolver.
- `property` - The key with which the specified value is to be associated.
- `val` - The value to be associated with the specified key.

**Throws:**
**ClassCastException** - if the class of the specified key or value prevents it from being stored in this map.

**NullPointerException** - if context is null, or if this map does not permit null keys or values, and the specified key or value is null.

**IllegalArgumentException** - if some aspect of this key or value prevents it from being stored in this map.

**ELException** - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

**PropertyNotWritableException** - if this resolver was constructed in read-only mode, or if the put operation is not supported by the underlying map.

```java
public boolean isReadOnly(ELContext context, Object base, Object property) {
    return context == null;
}
```

Throws **NullPointerException**: context null

Throws **ELException**: cause
public boolean isReadOnly(ELContext context, Object base, Object property)

If the base object is a map, returns whether a call to setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) will always fail.

If the base is a Map, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

If this resolver was constructed in read-only mode, this method will always return true.

If a Map was created using Collections.unmodifiableMap(java.util.Map), this method must return true. Unfortunately, there is no Collections API method to detect this. However, an implementation can create a prototype unmodifiable Map and query its runtime type to see if it matches the runtime type of the base object as a workaround.

Specified by:
isReadOnly in class ELResolver

Parameters:
context - The context of this evaluation.
base - The map to analyze. Only bases of type Map are handled by this resolver.
property - The key to return the read-only status for. Ignored by this resolver.

Returns:
If the propertyResolved property of ELContext was set to true, then true if calling the setValue method will always fail or false if it is possible that such a call may succeed; otherwise undefined.

Throws:
NullPointerException - if context is null
**ELException** - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

```java
public java.util.Iterator<E>
getFeatureDescriptors(ELContext context, Object base)
```

```java
Iterator 0 java.beans.FeatureDescriptor info
Map
```

- displayName - toString null "null"
- name – displayName
- shortDescription –
- expert - false
- hidden - false
- preferred - true

```java
FeatureDescriptor
```

- **ELResolver#TYPE** - getClass() null null
- **ELResolver#RESOLVABLE_AT_DESIGN_TIME** - true

```java
context
base
return 0
```

```java
getFeatureDescriptors
```

```java
public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext context, Object base)
```

If the base object is a map, returns an Iterator containing the set of
keys available in the Map. Otherwise, returns null.

The Iterator returned must contain zero or more instances of FeatureDescriptor. Each info object contains information about a key in the Map, and is initialized as follows:

- displayName - The return value of calling the toString method on this key, or "null" if the key is null.
- name - Same as displayName property.
- shortDescription - Empty string
- expert - false
- hidden - false
- preferred - true

In addition, the following named attributes must be set in the returned FeatureDescriptors:
- ELResolver.TYPE - The return value of calling the getClass() method on this key, or null if the key is null.
- ELResolver.RESOLVABLE_AT_DESIGN_TIME - true

Specified by:
getFeatureDescriptors in class ELResolver

Parameters:
context - The context of this evaluation.
base - The map whose keys are to be iterated over. Only bases of type Map are handled by this resolver.

Returns:
An Iterator containing zero or more (possibly infinitely more) FeatureDescriptor objects, each representing a key in this map, or null if the base object is not a map.

See Also:
FeatureDescriptor

public Class<T> getCommonPropertyType(ELContext context, Object base)
base       property       null
base       Map       Object.class       Map
getCommonPropertyType

```java
public Class<?>
getCommonPropertyType(ELContext context, Object base)
```

If the base object is a map, returns the most general type that this resolver accepts for the property argument. Otherwise, returns null.

Assuming the base is a Map, this method will always return Object.class. This is because Maps accept any object as a key.

Specified by:
- getCommonPropertyType in class ELResolver

Parameters:
- **context** - The context of this evaluation.
- **base** - The map to analyze. Only bases of type Map are handled by this resolver.

Returns:
- null if base is not a Map; otherwise Object.class.
javax.persistence  Annotation Type MapKey

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface MapKey

**Implements:** Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

```java
import java.util.Map

@Entity
public class Department {
    // ...
    @OneToMany(mappedBy="department")
    @MapKey(name="empId")
    public Map getEmployees() {...}
    // ...
}

@Entity
public class Employee {
    // ...
    @Id Integer getEmpid() {...}
    @ManyToOne
    @JoinColumn(name="dept_id")
    public Department getDepartment() {...}
    // ...
}
```

2

```java
@Entity
public class Department {
    // ...
    @OneToMany(mappedBy="department")
    @MapKey(name="empPK")
    public Map getEmployees() {...}
```

Is used to specify the map key for associations of type `Map`.

If a persistent field or property other than the primary key is used as a
map key then it is expected to have a uniqueness constraint associated
with it.

Example 1:

```java
@Entity
public class Department {
    ...
    @OneToMany(mappedBy="department")
    @MapKey(name="empId")
    public Map getEmployees() { ... }
    ...
}

@Entity
public class Employee {
    ...
    @Id Integer getEmpid() { ... }
    @ManyToOne
    @JoinColumn(name="dept_id")
    public Department getDepartment() { ... }
}```
Example 2:

@Entity
class Department {
...
    @OneToMany(mappedBy="department")
    @MapKey(name="empPK")
    public Map getEmployees() { ... }
...
}

@Entity
class Employee {
    @EmbeddedId
    public EmployeePK getEmpPK() { ... }
    ...
    @ManyToOne
    @JoinColumn(name="dept_id")
    public Department getDepartment() { ... }
...
}

@Embeddable
class EmployeePK {
    String name;
    Date bday;
}

Since:
Java Persistence 1.0

### Optional Element Summary

| **String** name | The name of the persistent field or property of the associated entity that is used as the map key. |

abstract public String name()
public abstract String name

The name of the persistent field or property of the associated entity that is used as the map key. If the name element is not specified, the primary key of the associated entity is used as the map key. If the primary key is a composite primary key and is mapped as IdClass, an instance of the primary key class is used as the key.

Default:

""
javax.jms Interface MapMessage

All Superinterfaces:
   Message

public interface MapMessage
extends Message

Implements: Message

MapMessage | String | Java null
MapMessage | Message | Map

MapMessage.setInt("foo", 6)
MapMessage setObject("foo", new Integer(6))

MapMessage | MessageNotWriteableException | clearBody
MapMessage | JMSException | valueOf() | String | String

<table>
<thead>
<tr>
<th></th>
<th>boolean</th>
<th>byte</th>
<th>short</th>
<th>char</th>
<th>int</th>
<th>long</th>
<th>float</th>
<th>double</th>
<th>String</th>
<th>byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>short</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>char</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>long</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>double</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>String</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>byte[]</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>

null null | valueOf(String) | char | String | null |
char | NullPointerException |
A `MapMessage` object is used to send a set of name-value pairs. The names are `String` objects, and the values are primitive data types in the Java programming language. The names must have a value that is not null, and not an empty string. The entries can be accessed sequentially or randomly by name. The order of the entries is undefined. `MapMessage` inherits from the `Message` interface and adds a message body that contains a Map.

The primitive types can be read or written explicitly using methods for each type. They may also be read or written generically as objects. For instance, a call to `MapMessage.setInt("foo", 6)` is equivalent to `MapMessage setObject("foo", new Integer(6))`. Both forms are provided, because the explicit form is convenient for static programming, and the object form is needed when types are not known at compile time.

When a client receives a `MapMessage`, it is in read-only mode. If a client attempts to write to the message at this point, a `MessageNotWriteableException` is thrown. If `clearBody` is called, the message can now be both read from and written to.

`MapMessage` objects support the following conversion table. The marked cases must be supported. The unmarked cases must throw a `JMSException`. The `String`-to-primitive conversions may throw a runtime exception if the primitive's `valueOf()` method does not accept it as a valid `String` representation of the primitive.

A value written as the row type can be read as the column type.

<table>
<thead>
<tr>
<th></th>
<th>boolean byte short char int long float double String byte</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>true</code></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td><code>false</code></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>
Attempting to read a null value as a primitive type must be treated as calling the primitive's corresponding `valueOf(String)` conversion method with a null value. Since `char` does not support a `String` conversion, attempting to read a null value as a `char` must throw a `NullPointerException`.

**Version:**
1.1 February 2, 002

**Author:**
Mark Hapner, Rich Burridge

**See Also:**
- `Session.createMapMessage()`, `BytesMessage`, `Message`, `ObjectMessage`, `StreamMessage`, `TextMessage`

---

### Field Summary

- Fields inherited from interface `javax.jms.Message`
  - `DEFAULT_DELIVERY_MODE`, `DEFAULT_PRIORITY`, `DEFAULT_TIME_TO_LIVE`

---

### Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th>getBoolean(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the boolean value with the specified name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte</th>
<th>getByte(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the byte value with the specified name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>byte[]</th>
<th>getBytes(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the byte array value with the specified name.</td>
</tr>
<tr>
<td>Type</td>
<td>Method</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------</td>
</tr>
<tr>
<td>char</td>
<td><code>getChar(String name)</code></td>
</tr>
<tr>
<td>double</td>
<td><code>getDouble(String name)</code></td>
</tr>
<tr>
<td>float</td>
<td><code>getFloat(String name)</code></td>
</tr>
<tr>
<td>int</td>
<td><code>getInt(String name)</code></td>
</tr>
<tr>
<td>long</td>
<td><code>getLong(String name)</code></td>
</tr>
<tr>
<td>Enumeration</td>
<td><code>getMapNames()</code></td>
</tr>
<tr>
<td>Object</td>
<td><code>getObject(String name)</code></td>
</tr>
<tr>
<td>short</td>
<td><code>getShort(String name)</code></td>
</tr>
<tr>
<td>String</td>
<td><code>getString(String name)</code></td>
</tr>
<tr>
<td>boolean</td>
<td><code>itemExists(String name)</code></td>
</tr>
<tr>
<td>void</td>
<td><code>setBoolean(String name, boolean value)</code></td>
</tr>
<tr>
<td>void</td>
<td><code>setByte(String name, byte value)</code></td>
</tr>
<tr>
<td>void</td>
<td><code>setBytes(String name, byte[] value)</code></td>
</tr>
<tr>
<td>void</td>
<td><code>setBytes(String name, byte[] value, int offset, int length)</code></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>setChar(String name, char value)</strong></td>
<td>Sets a Unicode character value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setDouble(String name, double value)</strong></td>
<td>Sets a double value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setFloat(String name, float value)</strong></td>
<td>Sets a float value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setInt(String name, int value)</strong></td>
<td>Sets an int value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setLong(String name, long value)</strong></td>
<td>Sets a long value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setObject(String name, Object value)</strong></td>
<td>Sets an object value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setShort(String name, short value)</strong></td>
<td>Sets a short value with the specified name into the Map.</td>
</tr>
<tr>
<td><strong>setString(String name, String value)</strong></td>
<td>Sets a String value with the specified name into the Map.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.Message

public boolean getBoolean(String name) throws JMSException

boolean

name

return

Throws JMSException: JMS

getBoolean

boolean getBoolean(String name)

throws JMSException

Returns the boolean value with the specified name.

Parameters:

name - the name of the boolean

Returns:

the boolean value with the specified name

Throws:

JMSException - if the JMS provider fails to read the message due to some internal error.

MessageFormatException - if this type conversion is invalid.

public byte getByte(String name) throws JMSException

byte

name

return

Throws JMSException: JMS

Throws MessageFormatException:
getBytes

byte getByte(String name) throws JMSException

Returns the byte value with the specified name.

Parameters:
  name - the name of the byte

Returns:
  the byte value with the specified name

Throws:
  JMSException - if the JMS provider fails to read the message due to some internal error.
  MessageFormatException - if this type conversion is invalid.

public short getShort(String name) throws JMSException

short getShort(String name) throws JMSException: JMS

Throws
  JMSException: JMS
  MessageFormatException: MessageFormatException:

goingShort

short getShort(String name) throws JMSException

Returns the short value with the specified name.

Parameters:
  name - the name of the short

Returns:
  the short value with the specified name

Throws:
  JMSException - if the JMS provider fails to read the message due to some internal error.
public char getChar(String name) throws IOException

Unicode

name
return

Throws IOException: JMS

Throws MessageFormatException:

getChar

char getChar(String name)
throws IOException

Returns the Unicode character value with the specified name.

Parameters:
name - the name of the Unicode character

Returns:
the Unicode character value with the specified name

Throws:
IOException - if the JMS provider fails to read the message due to some internal error.
MessageFormatException - if this type conversion is invalid.

public int getInt(String name) throws IOException

int

name
return

Throws IOException: JMS

Throws MessageFormatException:

getInt
int getInt(String name) throws JMSException

Returns the int value with the specified name.

Parameters:
   name - the name of the int

Returns:
   the int value with the specified name

Throws:
   JMSException - if the JMS provider fails to read the message due to some internal error.
   MessageFormatException - if this type conversion is invalid.

gelong getLong(String name) throws JMSException

long getLong(String name) throws JMSException

public long getLong(String name) throws JMSException
long

name
return
long
long

Throws
   JMSException: JMS
   MessageFormatException:

getLong

long getLong(String name) throws JMSException

Returns the long value with the specified name.

Parameters:
   name - the name of the long

Returns:
   the long value with the specified name

Throws:
   JMSException - if the JMS provider fails to read the message due to some internal error.
   MessageFormatException - if this type conversion is invalid.
public float getFloat(String name) throws JMSException

float

name float
return float

Throws JMSException: JMS

Throws MessageFormatException:

getFloat

float getFloat(String name)

throws JMSException

Returns the float value with the specified name.

Parameters:
name - the name of the float

Returns:
the float value with the specified name

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageFormatException - if this type conversion is invalid.

__________________________________________________________________________

gpublic double getDouble(String name) throws JMSException

double

double

name double
return double

Throws JMSException: JMS

Throws MessageFormatException:

gDouble

double getDouble(String name)
public String getString(String name) throws JMSException

Parameters:
    name - the name of the String

Returns:
    the String value with the specified name; if there is no item by 
    this name, a null value is returned

Throws:
    JMSException - if the JMS provider fails to read the message due 
    to some internal error.
    MessageFormatException - if this type conversion is invalid.
public byte[] getBytes(String name) throws JMSException

byte[] getBytes(String name)
    throws JMSException

Returns the byte array value with the specified name.

Parameters:
    name - the name of the byte array

Returns:
    a copy of the byte array value with the specified name; if there is
    no item by this name, a null value is returned.

Throws:
    JMSException - if the JMS provider fails to read the message due
to some internal error.
    MessageFormatException - if this type conversion is invalid.

public Object getObject(String name) throws JMSException

Object getObject(String name)
    throws JMSException

Java “Java ”
setObject settype
Map

byte byte[] Byte[]
    name Java
return Java
Throws JMSException: JMS
**getObject**

```java
Object getObject(String name)
```

throws `JMSException`

Returns the value of the object with the specified name.

This method can be used to return, in objectified format, an object in the Java programming language ("Java object") that had been stored in the Map with the equivalent `setObject` method call, or its equivalent primitive `settype` method.

Note that byte values are returned as `byte[]`, not `Byte[]`.

**Parameters:**
- `name` - the name of the Java object

**Returns:**
- a copy of the Java object value with the specified name, in objectified format (for example, if the object was set as an `int`, an `Integer` is returned); if there is no item by this name, a null value is returned

**Throws:**
- `JMSException` - if the JMS provider fails to read the message due to some internal error.

---

**public java.util.Enumeration<E> getMapNames() throws JMSException**

```java
MapMessage return Enumeration MapMessage
```

Throws `JMSException` - JMS

---

**getMapNames**

```java
Enumeration getMapNames()
```

throws `JMSException`
Returns an Enumeration of all the names in the MapMessage object.

**Returns:**
an enumeration of all the names in this MapMessage

**Throws:**
- `JMSException` - if the JMS provider fails to read the message due to some internal error.

---

```java
public void setBoolean(String name, boolean value) throws JMSException

setBoolean

void setBoolean(String name, boolean value)
throws JMSException

Sets a boolean value with the specified name into the Map.

**Parameters:**
- name - the name of the boolean
- value - the boolean value to set in the Map

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageNotWriteableException` - if the message is in read-only mode.
```
public void setByte(String name, byte value) throws JMSException

   Sets a byte value with the specified name into the Map.

   Parameters:
   name - the name of the byte
   value - the byte value to set in the Map

   Throws:
   JMSException - if the JMS provider fails to write the message due to some internal error.
   IllegalArgumentException - if the name is null or if the name is an empty string.
   MessageNotWriteableException - if the message is in read-only mode.

public void setShort(String name, short value) throws JMSException

   Sets a short value with the specified name into the Map.

   Parameters:
   name - the name of the short
   value - the short value to set in the Map

   Throws:
   JMSException - if the JMS provider fails to write the message due to some internal error.
   IllegalArgumentException - if the name is null or if the name is an empty string.
   MessageNotWriteableException - if the message is in read-only mode.
Throws  

**setMessageNotWriteableException**:  

**setShort**

```java
void setShort(String name, short value)
    throws JMSException
```

Sets a short value with the specified name into the Map.

**Parameters:**
- `name` - the name of the short value
- `value` - the short value to set in the Map

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageNotWriteableException` - if the message is in read-only mode.

---

**public void setChar(String name, char value) throws JMSException**

**Unicode Map**

<table>
<thead>
<tr>
<th>name</th>
<th>Unicode Map Unicode</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td></td>
</tr>
</tbody>
</table>

**Throws**
- `JMSException`: JMS
- `IllegalArgumentException`: null
- `MessageNotWriteableException`:  

**setChar**

```java
void setChar(String name, char value)
    throws JMSException
```
Sets a Unicode character value with the specified name into the Map.

Parameters:
- **name** - the name of the Unicode character
- **value** - the Unicode character value to set in the Map

Throws:
- **JMSException** - if the JMS provider fails to write the message due to some internal error.
- **IllegalArgumentException** - if the name is null or if the name is an empty string.
- **MessageNotWriteableException** - if the message is in read-only mode.

```java
public void setInt(String name, int value) throws JMSException
```

Sets an **int** value with the specified name into the Map.

Parameters:
- **name** - the name of the **int**
- **value** - the **int** value to set in the Map

Throws:
- **JMSException** - if the JMS provider fails to write the message due
public void setLong(String name, long value) throws JMSException

Sets a long value with the specified name into the Map.

Parameters:
name - the name of the long
value - the long value to set in the Map

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
IllegalArgumentException - if the name is null or if the name is an empty string.
MessageNotWriteableException - if the message is in read-only mode.

public void setFloat(String name, float value) throws
JMSException

float Map

name float
value Map float

Throws JMSException: JMS
Throws IllegalArgumentException: null
Throws MessageNotWriteableException:

setFloat

void setFloat(String name, float value) throws JMSException

Sets a float value with the specified name into the Map.

Parameters:
name - the name of the float
value - the float value to set in the Map

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
IllegalArgumentException - if the name is null or if the name is an empty string.
MessageNotWriteableException - if the message is in read-only mode.

public void setDouble(String name, double value) throws JMSException

double Map

name double
value Map double

Throws JMSException: JMS
Throws IllegalArgumentException: null
Throws MessageNotWriteableException:
**setDouble**

```java
void setDouble(String name, 
    double value) 
    throws JMSException
```

Sets a double value with the specified name into the Map.

**Parameters:**
- name - the name of the double
- value - the double value to set in the Map

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageNotWriteableException` - if the message is in read-only mode.

---

**public void setString(String name, String value) throws JMSException**

```
    String Map
    name String
    value Map String
```

**Throws**
- `JMSException`: JMS
- `IllegalArgumentException`: null
- `MessageNotWriteableException`

**setString**

```java
void setString(String name, 
    String value) 
    throws JMSException
```

Sets a string value with the specified name into the Map.
Parameters:
- **name** - the name of the String
- **value** - the String value to set in the Map

Throws:
- **JMSException** - if the JMS provider fails to write the message due to some internal error.
- **IllegalArgumentException** - if the name is null or if the name is an empty string.
- **MessageNotWriteableException** - if the message is in read-only mode.

```java
public void setBytes(String name, byte[] value) throws JMSException
```

**Description:**
Sets a byte array value with the specified name into the Map.

**Parameters:**
- **name** - the name of the byte array
- **value** - the byte array value to set in the Map; the array is copied so that the value for **name** will not be altered by future modifications

**Throws:**
- **JMSException** - if the JMS provider fails to write the message due to some internal error.
**NullPointerException** - if the name is null, or if the name is an empty string.

**MessageNotWriteableException** - if the message is in read-only mode.

```java
public void setBytes(String name, byte[] value, int offset, int length) throws JMSException
```

Sets a portion of the byte array value with the specified name into the Map.

**Parameters:**
- `name` - the name of the byte array
- `value` - the byte array value to set in the Map
- `offset` - the initial offset within the byte array
- `length` - the number of bytes to use

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is
public void setObject(String name, Object value) throws JMSException

Map

<table>
<thead>
<tr>
<th>Integer</th>
<th>Double</th>
<th>Long</th>
<th>String</th>
<th>byte</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td></td>
<td></td>
<td>Java</td>
<td></td>
</tr>
<tr>
<td>value</td>
<td></td>
<td></td>
<td>Map</td>
<td>Java</td>
</tr>
</tbody>
</table>

Throws: 

- JMSException: JMS
- IllegalArgumentException: null
- MessageFormatException:
- MessageNotWriteableException:

setObject

void setObject(String name, Object value) throws JMSException

Sets an object value with the specified name into the Map.

This method works only for the objectified primitive object types (Integer, Double, Long ...), String objects, and byte arrays.

**Parameters:**

- name - the name of the Java object
- value - the Java object value to set in the Map

**Throws:**

- JMSException - if the JMS provider fails to write the message due to some internal error.
- IllegalArgumentException - if the name is null or if the name is an empty string.
MessageFormatException - if the object is invalid.
MessageNotWriteableException - if the message is in read-only mode.

```java
public boolean itemExists(String name) throws JMSException
```

Returns true if the item exists

**Parameters:**
- `name` - the name of the item to test

**Returns:**
- true if the item exists

**Throws:**
- `JMSException` - if the JMS provider fails to determine if the item exists due to some internal error.
PS:
javax.resource.cci Interface MappedRecord

All Superinterfaces:
   Cloneable, Map, Record, Serializable

public interface MappedRecord
extends Record, Map, Serializable

Implements: Record, java.util.Map<K, V>, java.io.Serializable

javax.resource.cci.MappedRecord MappedRecord
java.util.Map

version 0.8

The interface javax.resource.cci.MappedRecord is used for key-value map based representation of record elements. The MappedRecord interface extends both Record and java.util.Map interfaces.

Version:
   0.8
Author:
   Rahul Sharma

Nested Class Summary

<table>
<thead>
<tr>
<th>Nested classes/interfaces inherited from interface java.util.Map</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map.Entry&lt;K, V&gt;</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from interface javax.resource.cci.Record</th>
</tr>
</thead>
</table>
clone, equals, getRecordName, getRecordShortDescription, hashCode, setRecordName, setRecordShortDescription

Methods inherited from interface java.util.Map

clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values
javax.persistence Annotation Type MappedSuperclass

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface MappedSuperclass

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

MappedSuperclass

AttributeOverride
AssociationOverride XML

@MappedSuperclass
public class Employee {

    @Id protected Integer empId;
    @Version protected Integer version;
    @ManyToOne @JoinColumn(name="ADDR")
    protected Address address;

    public Integer getEmpId() { ... }
    public void setEmpId(Integer id) { ... }
    public Address getAddress() { ... }
    public void setAddress(Address addr) { ... }
}

// Default table is FTEMPLOYEE table
@Entity
public class FTEmployee extends Employee {

    // Inherited empId field mapped to FTEMPLOYEE.EMPID
    // Inherited version field mapped to FTEMPLOYEE.VERSION
    // Inherited address field mapped to FTEMPLOYEE.ADDR fk

    // Defaults to FTEMPLOYEE.SALARY

    protected Integer salary;
public FTEmployee() {}

public Integer getSalary() { ... }
public void setSalary(Integer salary) { ... }

@Entity @Table(name="PT_EMP")
@AssociationOverride(name="address",
joinColumns=@JoinColumn(name="ADDR_ID"))
public class PartTimeEmployee extends Employee {

    // Inherited empId field mapped to PT_EMP.EMPID
    // Inherited version field mapped to PT_EMP.VERSION
    // address field mapping overridden to PT_EMP.ADDR_ID fk
    @Column(name="WAGE")
    protected Float hourlyWage;

    public PartTimeEmployee() {}

    public Float getHourlyWage() { ... }
    public void setHourlyWage(Float wage) { ... }
}

Example: Non-entity superclass

public class Cart {

    // This state is transient
    Integer operationCount;

    public Cart() { operationCount = 0; }
    public Integer getOperationCount() { return operationCount;
    public void incrementOperationCount() { operationCount++;
}

@Entity
public class ShoppingCart extends Cart {

    Collection items = new Vector();

    public ShoppingCart() { super(); }

    ...

Designates a class whose mapping information is applied to the entities that inherit from it. A mapped superclass has no separate table defined for it.

A class designated with the `@MappedSuperclass` annotation can be mapped in the same way as an entity except that the mappings will apply only to its subclasses since no table exists for the mapped superclass itself. When applied to the subclasses the inherited mappings will apply in the context of the subclass tables. Mapping information may be overridden in such subclasses by using the `@AttributeOverride` and `@AssociationOverride` annotations or corresponding XML elements.

Example: Concrete class as a mapped superclass

```java
@MappedSuperclass
public class Employee {

    @Id protected Integer empId;
    @Version protected Integer version;
    @ManyToOne @JoinColumn(name="ADDR")
    protected Address address;

    public Integer getEmpId() { ... }
    public void setEmpId(Integer id) { ... }
    public Address getAddress() { ... }
    public void setAddress(Address addr) { ... }
}
```

// Default table is FTEMPLOYEE table
```java
@Entity
public class FTEmployee extends Employee {

    // Inherited empId field mapped to FTEMPLOYEE.EMPID
```
// Inherited version field mapped to FTEMPLOYEE.VERSION
// Inherited address field mapped to FTEMPLOYEE.ADDR fk

// Defaults to FTEMPLOYEE.SALARY
protected Integer salary;

public FTEmployee() {}
Collection items = new Vector();

public ShoppingCart() { super(); }

...

@OneToMany
public Collection getItems() { return items; }
public void addItem(Item item) {
  items.add(item);
  incrementOperationCount();
}

Since:
Java Persistence 1.0

Overview Package Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAME NO FRAMES
SUMMARY: REQUIRED | OPTIONAL DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind  **Class MarshalException**

```java
class MarshalException extends JAXBException
```

**Extends:** Throwable > Exception > JAXBException

All Implemented Interfaces:
- Serializable

This exception indicates that an error has occurred while performing a marshal operation that the provider is unable to recover from.

The `ValidationEventHandler` can cause this exception to be thrown during the marshal operations. See `ValidationEventHandler.handleEvent(ValidationEvent)`.

Since: JAXB1.0

Version:
Author:
- Ryan Shoemaker, Sun Microsystems, Inc.

See Also:
 JAXBException, Marshaller, Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MarshalException(String message)</code></td>
<td>Construct a MarshalException with the specified detail message.</td>
</tr>
<tr>
<td><code>MarshalException(String message, String errorCode)</code></td>
<td>Construct a MarshalException with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td><code>MarshalException(String message, String errorCode, Throwable exception)</code></td>
<td>Construct a MarshalException with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td><code>MarshalException(String message, Throwable exception)</code></td>
<td>Construct a MarshalException with the specified detail message and linkedException.</td>
</tr>
<tr>
<td><code>MarshalException(Throwable exception)</code></td>
<td>Construct a MarshalException with a linkedException.</td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class javax.xml.bind.JAXBException

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getCause</code>, <code>getErrorCode</code>, <code>getLinkedException</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>printStackTrace</code>, <code>setLinkedException</code>, <code>toString</code></td>
<td></td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fillInStackTrace</code>, <code>getLocalizedMessage</code>, <code>getMessage</code>, <code>getStackTrace</code>, <code>initCause</code>, <code>setStackTrace</code></td>
<td></td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
<td></td>
</tr>
</tbody>
</table>


public MarshalException(String message)  
MarshalException errorCode  linkedException  null  

message

MarshalAsException

public MarshalException(String message)

Construct a MarshalException with the specified detail message. The errorCode and linkedException will default to null.

Parameters:
message - a description of the exception

public MarshalException(String message, String errorCode)

errorCode MarshalException linkedException null  

message  
errorCode

MarshalAsException

public MarshalException(String message, String errorCode)

Construct a MarshalException with the specified detail message and
vendor specific errorCode. The linkedException will default to null.

Parameters:
message - a description of the exception
errorCode - a string specifying the vendor specific error code

public MarshalException(Throwables exception)
linkedException MarshalException errorCode
null

exception

MarshalException

public MarshalException(Throwables exception)

Construct a MarshalException with a linkedException. The detail message and vendor specific errorCode will default to null.

Parameters:
exception - the linked exception

public MarshalException(String message, Throwables exception)
linkedException MarshalException errorCode
null

message

exception

MarshalException

public MarshalException(String message, Throwables exception)

Construct a MarshalException with the specified detail message and
linkedException. The errorCode will default to null.

Parameters:
message - a description of the exception
exception - the linked exception

public MarshalException(String message, String errorCode, Throwable exception)

errorCode linkedException MarshalException
  message
code
exception

MarshalException

public MarshalException(String message, String errorCode, Throwable exception)

Construct a MarshalException with the specified detail message, vendor specific errorCode, and linkedException.

Parameters:
message - a description of the exception
errorCode - a string specifying the vendor specific error code
exception - the linked exception
PS:
javax.xml.bind  **Interface Marshaller**

**All Known Implementing Classes:**  
[AbstractMarshallerImpl](#)

---

```java
public interface Marshaller

**Inner classes:** Marshaller.Listener
**Implemented by:** AbstractMarshallerImpl

Marshallers  Java  XML

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
Object element = u.unmarshal( new File( "foo.xml" ) );
Marshaller m = jc.createMarshaller();

File

OutputStream os = new FileOutputStream( "nosferatu.xml" );
m.marshal( element, os );

SAX ContentHandler

// assume MyContentHandler instanceof ContentHandler
m.marshal( element, new MyContentHandler() );

DOM Node

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.newDocument();
m.marshal( element, doc );```
java.io.OutputStream

m.marshal( element, System.out );

java.io.Writer

m.marshal( element, new PrintWriter( System.out ) );

javax.xml.transform.SAXResult

// assume MyContentHandler instanceof ContentHandler
SAXResult result = new SAXResult( new MyContentHandler() );
m.marshal( element, result );

javax.xml.transform.DOMResult

DOMResult result = new DOMResult();
m.marshal( element, result );

javax.xml.transform.StreamResult

StreamResult result = new StreamResult( System.out );
m.marshal( element, result );

javax.xml.stream.XMLStreamWriter

XMLStreamWriter xmlStreamWriter =
XMLOutputFactory.newInstance().createXMLStreamWriter( ... );
m.marshal( element, xmlStreamWriter );

javax.xml.stream.XMLEventWriter

XMLEventWriter xmlEventWriter =
XMLOutputFactory.newInstance().createXMLEventWriter( ... );
m.marshal( element, xmlEventWriter );
JAXB

marshaller.marshal(java.lang.Object, ...)

isElement(java.lang.Object) JAXB marshaller.marshal

MarshalException JAXB JAXBEelement

Marshaller.marshal java @ XmlRootElement

XML java.io.OutputStream java.io.Writer

Marshaller UTF-8 setProperty API W3C XML 1.

Recommendation Java

API Java XML JAXB Java

JAXB XML JAXB JAXB

JAXB JAXB

API JAXB 2.0

javax.xml.bind.helpers.DefaultValidationEventHandler

JAXB

jaxb.encoding java.lang.String

XML Marshaller "UTF-8"

jaxb.formatted.output java.lang.Boolean

Marshaller XML true xml false xml

Marshaller false

jaxb.schemaLocation java.lang.String

XML xsi:schemaLocation schemaLocation

W3C XML Schema Part 0:Primer 5.6 W3C XML

Schema Part 1:Structures 2.6

jaxb.noNamespaceSchemaLocation java.lang.String

XML xsi:noNamespaceSchemaLocation schemaLocation

W3C XML Sc
5.6 W3C XML Schema Part 1: Structures 2.6

jaxb.fragment java.lang.Boolean
Marshaller false true API

- `marshal(Object,ContentHandler)` Marshaller
  `startDocument()` `endDocument()`
- `marshal(Object,Node)` API
- `marshal(Object,OutputStream)` Marshaller xml
- `marshal(Object,Writer)` Marshaller xml
- `marshal(Object,Result)` Result Node
  ContentHandler Stream API
- `marshal(Object,XMLEventWriter)` Marshaller
  `javax.xml.stream.events.XMLEvent#START_DOCUMENT`
  `javax.xml.stream.events.XMLEvent#END_DOCUMENT`
- `marshal(Object,XMLStreamWriter)` Marshaller
  `javax.xml.stream.events.XMLEvent#START_DOCUMENT`
  `javax.xml.stream.events.XMLEvent#END_DOCUMENT`

Marshaller """ JAXB """

JAXB

// Invoked by Marshaller after it has created an instance of this
// boolean beforeMarshal(Marshaller, Object parent);

// Invoked by Marshaller after it has marshalled all properties
void afterMarshal(Marshaller, Object parent);

/

Listener setListener(Listener)

"""

The **Marshaller** class is responsible for governing the process of serializing Java content trees back into XML data. It provides the basic marshalling methods:

**Assume the following setup code for all following code fragments:**

```java
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
Object element = u.unmarshal( new File( "foo.xml" ) );
Marshaller m = jc.createMarshaller();
```

**Marshalling to a File:**

```java
OutputStream os = new FileOutputStream( "nosferatu.xml" );
m.marshal( element, os );
```

**Marshalling to a SAX ContentHandler:**

```java
// assume MyContentHandler instanceof ContentHandler
m.marshal( element, new MyContentHandler() );
```

**Marshalling to a DOM Node:**

```java
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.newDocument();

m.marshal( element, doc );
```

**Marshalling to a java.io.OutputStream:**

```java
m.marshal( element, System.out );
```

**Marshalling to a java.io.Writer:**

```java
```

---

*Note: The above code snippets are simplified examples for illustration purposes and may require additional setup or configuration in a real-world application.*
m.marshal( element, new PrintWriter( System.out ) );

Marshalling to a javax.xml.transform.SAXResult:

// assume MyContentHandler instanceof ContentHandler
SAXResult result = new SAXResult( new MyContentHandler() );
m.marshal( element, result );

Marshalling to a javax.xml.transform.DOMResult:

DOMResult result = new DOMResult();
m.marshal( element, result );

Marshalling to a javax.xml.transform.StreamResult:

StreamResult result = new StreamResult( System.out );
m.marshal( element, result );

Marshalling to a javax.xml.stream.XMLStreamWriter:

XMLStreamWriter xmlStreamWriter =
    XMLOutputFactory.newInstance().createXMLStreamWriter( .
m.marshal( element, xmlStreamWriter );

Marshalling to a javax.xml.stream.XMLEventWriter:

XMLEventWriter xmlEventWriter =
    XMLOutputFactory.newInstance().createXMLEventWriter( ..
m.marshal( element, xmlEventWriter );

Marshalling content tree rooted by a JAXB element

The first parameter of the overloaded
Marshaller.marshal(java.lang.Object, ...) methods must be a
JAXB element as computed by
JAXBIntrospector.isElement(java.lang.Object); otherwise, a Marshaller.marshal method must throw a MarshalException. There exist two mechanisms to enable marshalling an instance that is not a JAXB element. One method is to wrap the instance as a value of a JAXBElememt, and pass the wrapper element as the first parameter to a Marshaller.marshal method. For java to schema binding, it is also possible to simply annotate the instance's class with @XmlRootElement.

Encoding

By default, the Marshaller will use UTF-8 encoding when generating XML data to a java.io.OutputStream, or a java.io.Writer. Use the setProperty API to change the output encoding used during these marshal operations. Client applications are expected to supply a valid character encoding name as defined in the W3C XML 1.0 Recommendation and supported by your Java Platform.

Validation and Well-Formedness

Client applications are not required to validate the Java content tree prior to calling any of the marshal API's. Furthermore, there is no requirement that the Java content tree be valid with respect to its original schema in order to marshal it back into XML data. Different JAXB Providers will support marshalling invalid Java content trees at varying levels, however all JAXB Providers must be able to marshal a valid content tree back to XML data. A JAXB Provider must throw a MarshalException when it is unable to complete the marshal operation due to invalid content. Some JAXB Providers will fully allow marshalling invalid content, others will fail on the first validation error.

Even when schema validation is not explicitly enabled for the marshal operation, it is possible that certain types of validation events will be detected during the operation. Validation events will be reported to the registered event handler. If the client application has not registered an event handler prior to invoking one of the marshal API's, then events will be delivered to a default event handler which will terminate the marshal operation after encountering the first error.
or fatal error. Note that for JAXB 2.0 and later versions, DefaultValidationEventHandler is no longer used.

Supported Properties

All JAXB Providers are required to support the following set of properties. Some providers may support additional properties.

jaxb.encoding - value must be a java.lang.String
  The output encoding to use when marshalling the XML data. The Marshaller will use "UTF-8" by default if this property is not specified.

jaxb.formatted.output - value must be a java.lang.Boolean
  This property controls whether or not the Marshaller will format the resulting XML data with line breaks and indentation. A true value for this property indicates human readable indented xml data, while a false value indicates unformatted xml data. The Marshaller will default to false (unformatted) if this property is not specified.

jaxb.schemaLocation - value must be a java.lang.String
  This property allows the client application to specify an xsi:schemaLocation attribute in the generated XML data. The format of the schemaLocation attribute value is discussed in an easy to understand, non-normative form in Section 5.6 of the W3C XML Schema Part 0: Primer and specified in Section 2.6 of the W3C XML Schema Part 1: Structures.

jaxb.noNamespaceSchemaLocation - value must be a java.lang.String
  This property allows the client application to specify an xsi:noNamespaceSchemaLocation attribute in the generated XML data. The format of the schemaLocation attribute value is discussed in an easy to understand, non-normative form in Section 5.6 of the W3C XML Schema Part 0: Primer and specified in Section 2.6 of the W3C XML Schema Part 1: Structures.

jaxb.fragment - value must be a java.lang.Boolean
  This property determines whether or not document level events will be generated by the Marshaller. If the property is not specified, the default is false. This property has different implications depending on which marshal api you are using -
when this property is set to true:

- **marshal(Object,ContentHandler)** - the Marshaller won't invoke `ContentHandler.startDocument()` and `ContentHandler.endDocument()`.
- **marshal(Object,Node)** - the property has no effect on this API.
- **marshal(Object,OutputStream)** - the Marshaller won't generate an xml declaration.
- **marshal(Object,Writer)** - the Marshaller won't generate an xml declaration.
- **marshal(Object,Result)** - depends on the kind of Result object, see semantics for Node, ContentHandler, and Stream APIs
- **marshal(Object,XMLStreamWriter)** - the Marshaller will not generate `XMLStreamConstants.START_DOCUMENT` and `XMLStreamConstants.END_DOCUMENT` events.
- **marshal(Object,XMLStreamWriter)** - the Marshaller will not generate `XMLStreamConstants.START_DOCUMENT` and `XMLStreamConstants.END_DOCUMENT` events.

**Marshal Event Callbacks**

"The Marshaller provides two styles of callback mechanisms that allow application specific processing during key points in the unmarshalling process. In 'class defined' event callbacks, application specific code placed in JAXB mapped classes is triggered during marshalling. 'External listeners' allow for centralized processing of marshal events in one callback method rather than by type event callbacks.

Class defined event callback methods allow any JAXB mapped class to specify its own specific callback methods by defining methods with the following method signatures:

```java
// Invoked by Marshaller after it has created an instance of boolean beforeMarshal(Marshaller, Object parent);

// Invoked by Marshaller after it has marshalled all properties void afterMmarshal(Marshaller, Object parent);
```
The class defined event callback methods should be used when the callback method requires access to non-public methods and/or fields of the class.

The external listener callback mechanism enables the registration of a `Marshaller.Listener` instance with a `setListener(Listener)` method. The external listener receives all callback events, allowing for more centralized processing than per class defined callback methods.

The 'class defined' and external listener event callback methods are independent of each other, both can be called for one event. The invocation ordering when both listener callback methods exist is defined in `Marshaller.Listener.beforeMarshal(Object)` and `Marshaller.Listener.afterMarshal(Object)`.

An event callback method throwing an exception terminates the current marshal process.

Since:  
JAXB1.0  

Version:  
$Revision: 1.19 $ $Date: 2006/03/08 16:54:42 $

Author:  
- Kohsuke Kawaguchi, Sun Microsystems, Inc.  
- Ryan Shoemaker, Sun Microsystems, Inc.  
- Joe Fialli, Sun Microsystems, Inc.

See Also:  
JAXBContext, Validator, Unmarshaller

---

**Nested Class Summary**

<table>
<thead>
<tr>
<th>static class</th>
<th><code>Marshaller.Listener</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Register an instance of an implementation of this class with a <code>Marshaller</code> to externally listen for marshal events.</td>
</tr>
</tbody>
</table>
Field Summary

| static String | JAXB_ENCODING | The name of the property used to specify the output encoding in the marshalled XML data. |
| static String | JAXB_FORMATTED_OUTPUT | The name of the property used to specify whether or not the marshalled XML data is formatted with linefeeds and indentation. |
| static String | JAXB_FRAGMENT | The name of the property used to specify whether or not the marshaller will generate document level events (ie calling startDocument or endDocument). |
| static String | JAXB_NO_NAMESPACE_SCHEMA_LOCATION | The name of the property used to specify the xsi:noNamespaceSchemaLocation attribute value to place in the marshalled XML output. |
| static String | JAXB_SCHEMA_LOCATION | The name of the property used to specify the xsi:schemaLocation attribute value to place in the marshalled XML output. |

Method Summary

getAdapter(Class<A> type)

- Gets the adapter associated with the specified type.

AttachmentMarshaller getAttachmentMarshaller()

- Return the current event handler or the default event handler if one hasn't been set.

ValidationEventHandler getEventHandler()

- Return Marshaller.Listener registered with this Marshaller.

getNode(Object contentTree)

- Get a DOM tree view of the content tree(Optional).

getProperty(String name)

- Get the particular property in the underlying implementation of Marshaller.
Get the JAXP 1.3 Schema object being used to perform marshal-time validation.

```java
void marshal(Object jaxbElement, ContentHandler handler)
```

Marshal the content tree rooted at jaxbElement into SAX2 events.

```java
void marshal(Object jaxbElement, Node node)
```

Marshal the content tree rooted at jaxbElement into a DOM tree.

```java
void marshal(Object jaxbElement, OutputStream os)
```

Marshal the content tree rooted at jaxbElement into an output stream.

```java
void marshal(Object jaxbElement, Result result)
```

Marshal the content tree rooted at jaxbElement into the specified javax.xml.transform.Result.

```java
void marshal(Object jaxbElement, Writer writer)
```

Marshal the content tree rooted at jaxbElement into a Writer.

```java
void marshal(Object jaxbElement, XMLEventWriter writer)
```

Marshal the content tree rooted at jaxbElement into an XMLEventWriter.

```java
void marshal(Object jaxbElement, XMLStreamWriter writer)
```

Marshal the content tree rooted at jaxbElement into a XMLStreamWriter.

```java
<A extends XmlAdapter>
void setAdapter(Class<A> type, A adapter)
```

Associates a configured instance of XmlAdapter with this marshaller.

```java
void setAdapter(XmlAdapter adapter)
```

Associates a configured instance of XmlAdapter with this marshaller.

```java
void setAttachmentMarshaller(AttachmentMarshaller am)
```

Associate a context that enables binary data within an XML document to be transmitted as XML-binary optimized attachment.

```java
void setEventHandler(ValidationEventHandler handler)
```

Allow an application to register a validation event handler.

```java
void setListener(Marshaller.Listener listener)
```

Register marshal event callback Marshaller.Listener with this Marshaller.

```java
void setProperty(String name, Object value)
```

Set the particular property in the underlying implementation of Marshaller.

```java
void setSchema(Schema schema)
```

Specify the JAXP 1.3 Schema object that should be used to validate subsequent marshal operations against.
Field Detail

JAXB_ENCODING

static final String JAXB_ENCODING

The name of the property used to specify the output encoding in the marshalled XML data.

See Also:
Constant Field Values

JAXB_FORMATTED_OUTPUT

static final String JAXB_FORMATTED_OUTPUT

The name of the property used to specify whether or not the marshalled XML data is formatted with linefeeds and indentation.

See Also:
Constant Field Values

JAXB_SCHEMA_LOCATION

static final String JAXB_SCHEMA_LOCATION

The name of the property used to specify the xsi:schemaLocation attribute value to place in the marshalled XML output.

See Also:
Constant Field Values
JAXB_NO_NAMESPACE_SCHEMA_LOCATION

static final String JAXB_NO_NAMESPACE_SCHEMA_LOCATION

The name of the property used to specify the xsi:noNamespaceSchemaLocation attribute value to place in the marshalled XML output.

See Also:
Constant Field Values

JAXB_FRAGMENT

static final String JAXB_FRAGMENT

The name of the property used to specify whether or not the marshaller will generate document level events (ie calling startDocument or endDocument).

See Also:
Constant Field Values

Method Detail

public void marshal(Object jaxbElement, javax.xml.transform.Result result) throws JAXBException

jaxbElement javax.xml.transform.Result

JAXB
javax.xml.transform.dom.DOMResultjavax.xml.transform.sax.SAXResult
javax.xml.transform.stream.StreamResult Result
marshal

```java
public void marshal(Object jaxbElement, java.io.OutputStream os) throws JAXBException
```

**Parameters:**
- `jaxbElement` - The root of content tree to be marshalled.
- `result` - XML will be sent to this Result

**Throws:**
- `JAXBException` - If any unexpected problem occurs during the marshalling.
- `MarshalException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Marshaller` is unable to marshal `obj` (or any object reachable from `obj`). See [Marshalling a JAXB element](https://docs.oracle.com/javase/8/docs/api/javax/xml/bind/Marshaller.html#marshal-javax.xml.bind.Element;javax.xml.transform.Result;). See [Marshalling a JAXB element](https://docs.oracle.com/javase/8/docs/api/javax/xml/bind/Marshaller.html#marshal-javax.xml.bind.Element;javax.xml.transform.Result;).
- `IllegalArgumentException` - If any of the method parameters are null

This method marshall the content tree rooted at `jaxbElement` into the specified `javax.xml.transform.Result`. All JAXB Providers must at least support `DOMResult`, `SAXResult`, and `StreamResult`. It can support other derived classes of `Result` as well.
void marshal(Object jaxbElement, OutputStream os)
    throws JAXBException

Marshal the content tree rooted at jaxbElement into an output stream.

Parameters:
  jaxbElement - The root of content tree to be marshalled.
  os - XML will be added to this stream.

Throws:
  JAXBException - If any unexpected problem occurs during the marshalling.
  MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.
  IllegalArgument Exception - If any of the method parameters are null
public void marshal(Object jaxbElement, org.xml.sax.ContentHandler handler) throws JAXBException

jaxbElement  SAX2

handler  XML  SAX2

Throws  JAXBException:

MarshalException:  ValidationEventHandler

Throws  handleEvent  false  Marshaller  obj  obj  .  JAXB

Throws  IllegalArgumentException:  null
marshal

void marshal(Object jaxbElement, ContentHandler handler) throws JAXBException

Marshal the content tree rooted at jaxbElement into SAX2 events.

Parameters:
  jaxbElement - The root of content tree to be marshalled.
  handler - XML will be sent to this handler as SAX2 events.

Throws:
  JAXBException - If any unexpected problem occurs during the marshalling.
  MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.
  IllegalArgumentException - If any of the method parameters are null

domain

public void marshal(Object jaxbElement, org.w3c.dom.Node node) throws JAXBException
  jaxbElement DOM
  node org.w3c.dom.Documentorg.w3c.dom.DocumentFragment
  org.w3c.dom.Element Node

Throws
  JAXBException:
  MarshalException: ValidationEventHandler

Throws
  handleEvent false Marshaller jaxbElement

Throws
  IllegalArgumentException: null
marshal

```java
void marshal(Object jaxbElement,
             Node node)
  throws JAXBException
```

Marshal the content tree rooted at jaxbElement into a DOM tree.

**Parameters:**
- `jaxbElement` - The content tree to be marshalled.
- `node` - DOM nodes will be added as children of this node. This parameter must be a Node that accepts children (Document, DocumentFragment, or Element)

**Throws:**
- `JAXBException` - If any unexpected problem occurs during the marshalling.
- `MarshalException` - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal jaxbElement (or any object reachable from jaxbElement). See Marshalling a JAXB element.
- `IllegalArgumentException` - If any of the method parameters are null

```java
public void marshal(Object jaxbElement, XMLStreamWriter writer) throws JAXBException
```

```java
jaxbElement javax.xml.stream.XMLStreamWriter
jaxbElement
writer XML
```

**Throws**
- `JAXBException`
- `MarshalException`: ValidationEventHandler
- `IllegalArgumentException`: null

since JAXB 2.0
void marshal(Object jaxbElement,
             XMLStreamWriter writer)
throws JAXBException

Marshal the content tree rooted at jaxbElement into a
XMLStreamWriter.

Parameters:
  jaxbElement - The content tree to be marshalled.
  writer - XML will be sent to this writer.

Throws:
  JAXBException - If any unexpected problem occurs during the
  marshalling.
  MarshalException - If the ValidationEventHandler returns false
  from its handleEvent method or the Marshaller is unable to
  marshal obj (or any object reachable from obj). See Marshalling
  a JAXB element.
  IllegalArgumentException - If any of the method parameters are
  null

Since:
  JAXB 2.0

public void marshal(Object jaxbElement, XMLEventWriter
writer) throws JAXBException

jaxbElement   : javax.xml.stream.XMLEventWriter
jaxbElement   : jaxbElement
writer        : XML

Throws
  JAXBException:
  MarshalException: ValidationEventHandler

Throws
  handleEvent false Marshaller obj obj .
  JAXB

Throws
  IllegalArgumentException: null

since
  JAXB 2.0

marshal
void marshal(Object jaxbElement, XMLEventWriter writer) throws JAXBException

Marshal the content tree rooted at jaxbElement into a XMLEventWriter.

Parameters:
jaxbElement - The content tree rooted at jaxbElement to be marshalled.
writer - XML will be sent to this writer.

Throws:
JAXBException - If any unexpected problem occurs during the marshalling.
MarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Marshaller is unable to marshal obj (or any object reachable from obj). See Marshalling a JAXB element.
IllegalArgumentException - If any of the method parameters are null

Since:
JAXB 2.0

public org.w3c.dom.Node getNode(Object contentTree) throws JAXBException

DOM DOM
org.w3c.dom.Node DOM
contentTree XML JAXB Java
return contentTree DOM

Throws UnsupportedOperationException: JAXB DOM
Throws IllegalArgumentException: null
Throws JAXBException:

getNode

Node getNode(Object contentTree)
Get a DOM tree view of the content tree(Optional). If the returned DOM tree is updated, these changes are also visible in the content tree. Use `marshal(Object, org.w3c.dom.Node)` to force a deep copy of the content tree to a DOM representation.

**Parameters:**
- `contentTree` - JAXB Java representation of XML content

**Returns:**
- the DOM tree view of the content tree

**Throws:**
- `UnsupportedOperationException` - If the JAXB provider implementation does not support a DOM view of the content tree
- `IllegalArgumentException` - If any of the method parameters are null
- `JAXBException` - If any unexpected problem occurs

---

```java
def setProperty(String name, Object value) throws PropertyException
```

**Thrown by:**
- `JAXB PropertyException`

**Throws:**
- `PropertyException`: 
- `IllegalArgumentException`: name null

---

Set the particular property in the underlying implementation of the `Marshaller`. This method can only be used to set one of the standard JAXB defined properties above or a provider specific property. Attempting to set an undefined property will result in an
PropertyException being thrown. See Supported Properties.

Parameters:
- `name` - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
- `value` - the value of the property to be set

Throws:
- `PropertyException` - when there is an error processing the given property or value
- `IllegalArgumentException` - If the name parameter is null

```java
public Object getProperty(String name) throws PropertyException
```

Get the particular property in the underlying implementation of Marshaller. This method can only be used to get one of the standard JAXB defined properties above or a provider specific property. Attempting to get an undefined property will result in a PropertyException being thrown. See Supported Properties.

Parameters:
- `name` - the name of the property to retrieve

Returns:
- the value of the requested property

Throws:
public void setEventHandler(ValidationEventHandler handler) throws JAXBException

API JAXB marshal

null Marshaller

handler

Throws JAXBException:

setEventHandler

void setEventHandler(ValidationEventHandler handler) throws JAXBException

Allow an application to register a validation event handler.

The validation event handler will be called by the JAXB Provider if any validation errors are encountered during calls to any of the marshal API's. If the client application does not register a validation event handler before invoking one of the marshal methods, then validation events will be handled by the default event handler which will terminate the marshal operation after the first error or fatal error is encountered.

Calling this method with a null parameter will cause the Marshaller to revert back to the default default event handler.

Parameters:

handler - the validation event handler

Throws:
public ValidationEventHandler getEventHandler() throws JAXBException

return ValidationEventHandler

Throws JAXBException: 

getEventHandler

ValidationEventHandler getEventHandler() throws JAXBException

Return the current event handler or the default event handler if one hasn't been set.

Returns: 
the current ValidationEventHandler or the default event handler if it hasn't been set

Throws: 
JAXBException - if an error was encountered while getting the current event handler

public void setAdapter("XmlAdapter adapter)

Xm1Adapter marshaller

setAdapter(adapter.getClass(),adapter)

Throws IllegalArgumentException: adapter null

Throws UnsupportedOperationException: JAXB 1.0 since JAXB 2.0

See also setAdapter(Class,XmlAdapter)
setAdapter
void setAdapter(XmlAdapter adapter)

Associates a configured instance of XmlAdapter with this marshaller.

This is a convenience method that invokes
setAdapter(adapter.getClass(), adapter);

Throws:
IllegalArgumentException - if the adapter parameter is null.
UnsupportedOperationException - if invoked against a JAXB 1.0 implementation.

Since:
JAXB 2.0

See Also:
setAdapter(Class, XmlAdapter)

setAdapter
<A extends XmlAdapter> void setAdapter(Class<A> type,
A adapter)

Associates a configured instance of XmlAdapter with this marshaller.

Every marshaller internally maintains a Map<Class, XmlAdapter>,
which it uses for marshalling classes whose fields/methods are
annotated with XmlJavaTypeAdapter.

This method allows applications to use a configured instance of
XmlAdapter. When an instance of an adapter is not given, a
marshaller will create one by invoking its default constructor.

Parameters:
type - The type of the adapter. The specified instance will be used when XmlJavaTypeAdapter.value() refers to this type.
adapter - The instance of the adapter to be used. If null, it will
un-register the current adapter set for this type.

**Throws:**
- `IllegalArgumentException` - if the type parameter is null.
- `UnsupportedOperationException` - if invoked agains a JAXB 1.0 implementation.

**Since:**
JAXB 2.0

---

**getAdapter**

```java
<A extends XmlAdapter> A getAdapter(Class<A> type)
```

Gets the adapter associated with the specified type. This is the reverse operation of the method.

**Throws:**
- `IllegalArgumentException` - if the type parameter is null.
- `UnsupportedOperationException` - if invoked agains a JAXB 1.0 implementation.

**Since:**
JAXB 2.0

---

**public void setAttachmentMarshaller(AttachmentMarshaller am)**

**XML XML xml id URI (cid) XML**

**Throws**
IllegalStateException:

---

**setAttachmentMarshaller**

```java
void setAttachmentMarshaller(AttachmentMarshaller am)
```
Associate a context that enables binary data within an XML document to be transmitted as XML-binary optimized attachment. The attachment is referenced from the XML document content model by content-id URIs(cid) references stored within the xml document.

**Throws:**

*IllegalStateException* - if attempt to concurrently call this method during a marshal operation.

```java
public AttachmentMarshaller getAttachmentMarshaller()
```

**getAttachmentMarshaller**

```java
AttachmentMarshaller getAttachmentMarshaller()
```

```java
public void setSchema(javax.xml.validation.Schema schema)
```

**setSchema**

```java
void setSchema(Schema schema)
```

Specify the JAXP 1.3 Schema object that should be used to validate
subsequent marshal operations against. Passing null into this method will disable validation.

This method allows the caller to validate the marshalled XML as it's marshalled.

Initially this property is set to null.

Parameters:

- `schema` - Schema object to validate marshal operations against or null to disable validation

Throws:

- `UnsupportedOperationException` - could be thrown if this method is invoked on an Marshaller created from a JAXBContext referencing JAXB 1.0 mapped classes

Since:

JAXB2.0

---

```java
public javax.xml.validation.Schema getSchema()
```

JAXP 1.3

```java
getSchema()
```

Get the JAXP 1.3 `Schema` object being used to perform marshal-time validation. If there is no Schema set on the marshaller, then this method will return null indicating that marshal-time validation will not be performed.

Returns:
the Schema object being used to perform marshal-time validation or null if not present.

Throws:

UnsupportedOperationException - could be thrown if this method is invoked on an Marshaller created from a JAXBContext referencing JAXB 1.0 mapped classes

Since:
JAXB2.0

public void setListener(Marshaller.Listener listener)

Marshaller Listener Listener Listener Listener
listener null Listener

setListener

void setListener(Marshaller.Listener listener)

Register marshal event callback Marshaller.Listener with this Marshaller.

There is only one Listener per Marshaller. Setting a Listener replaces the previous set Listener. One can unregister current Listener by setting listener to null.

Parameters:

listener - an instance of a class that implements Marshaller.Listener

Since:
JAXB2.0
public Marshaller.Listener getListener()

getterListener

Marshaller.Listener getListener()

Return Marshaller.Listener registered with this Marshaller.

Returns:
 registered Marshaller.Listener or null if no Listener is registered with this Marshaller.

Since:
JAXB2.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.bind  **Class MarshallListener**

**java.lang.Object**  
  `javax.xml.bind.MarshallListener`

**Enclosing interface:**  
  **Marshall**

```java
public abstract static class MarshallListener
extends Object
```

**Contained within:**  
  **Marshall**

Marshaller

xml  Java

**Marshal Event Callback**

**since**  JAXB2.0

**See also**  ```setListener(Listener), getListener()```  

Register an instance of an implementation of this class with a **Marshall** to externally listen for marshal events.

This class enables pre and post processing of each marshalled object. The event callbacks are called when marshalling from an instance that maps to an xml element or complex type definition. The event callbacks are not called when marshalling from an instance of a Java datatype that represents a simple type definition.

External listener is one of two different mechanisms for defining marshal event callbacks. See **Marshal Event Callbacks** for an overview.

**Since:**
JAXB2.0

See Also:
Marshaller.setListener(Listener), Marshaller.getListener()

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marshaller.Listener()</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void afterMarshal(Object source)</td>
<td>Callback method invoked after marshalling source to XML.</td>
</tr>
<tr>
<td>void beforeMarshal(Object source)</td>
<td>Callback method invoked before marshalling from source to XML.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

public Marshaller.Listener()
Method Detail

public void beforeMarshal(Object source)

beforeMarshal

public void before Marshal( Object source )

Callback method invoked before marshalling from source to XML.

This method is invoked just before marshalling process starts to marshal source. Note that if the class of source defines its own beforeMarshal method, the class specific callback method is invoked just before this method is invoked.

Parameters:

source - instance of JAXB mapped class prior to marshalling from it.

public void afterMarshal(Object source)

afterMarshal

public void after Marshal( Object source )

source XML

source source afterMarshal

source source JAXB
public void afterMarshal(Object source)

Callback method invoked after marshalling source to XML.

This method is invoked after source and all its descendants have been marshalled. Note that if the class of source defines its own afterMarshal method, the class specific callback method is invoked just before this method is invoked.

Parameters:

source - instance of JAXB mapped class after marshalling it.
javax.jms Interface Message

All Known Subinterfaces:
- BytesMessage, MapMessage, ObjectMessage, StreamMessage, TextMessage

public interface Message

Implemented by: BytesMessage, MapMessage, ObjectMessage, StreamMessage, TextMessage

Message JMS acknowledge

(MOM)

JMS API

JMS

- API
- API -
- 
- Java “Java ”
- (Extensible Markup Language, XML)

JMS

- Header-
- Property-
- Body- JMS API

JMS API

- Stream- StreamMessage Java “Java ”
- Map- MapMessage - String Java
### JMS Correlation ID

- `JMSCorrelationID ID String - byte[]`

### Message

- `null null IllegalArgumentException`
- `boolean byte short int long float double String`
- `MessageNotWriteableException clearProperties`

### JMS Exception

- `JMSException` `valueOf` `String` `String`

<table>
<thead>
<tr>
<th></th>
<th>boolean</th>
<th>byte</th>
<th>short</th>
<th>int</th>
<th>long</th>
<th>float</th>
<th>double</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>short</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>long</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>double</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>
setObjectProperty Boolean Byte Short Integer Long Float Double String JMSException
getObjectProperty Boolean Byte Short Integer Long Float Double String
getPropertyNames
clearProperties
null getStringProperty getObjectProperty null null
valueOf(String) null

JMS API JMS JMSX Java JMS JMS API
String[] ConnectionMetaData.getJMSXPropertyNames JMSX
JMSX
"" JMSX "" JMSX
JMSXGroupID JMSXGroupSeq JMSX

JMS API JMS_vendor_name vendor_name JMS JMS (per-message services)
JMS JMS - JMS

JMS Message

JMS API JMS
JMS Session

JMSReplyTo
true

String SQL92 null

- 'literal' 'literal''s' Java Unicode
- 57-957 +62 long Java
- 7E3 -57.9E2 7.-95.7 +6.2 double Java
- TRUE FALSE

- Character.isJavaLetter true ' ' '$'
- Character.isJavaLetterOrDigit true
- NULL TRUE FALSE
- NOT AND OR BETWEEN LIKE IN IS ESCAPE
- NULL

myMessage.setStringProperty("NumberOfOrders", "2");

false

"NumberOfOrders > 1"

JMSDeliveryModeJMSPriorityJMSMessageIDJMSTimestampJMSCorrelationID
null
  o 'JMSX' JMS
  o 'JMS_'
  o 'JMS'

Java
  o true false
  o
  o

() NOT AND OR
  o Java (numeric promotion) false
  o
  o += - / +
  o += /
  o += -

Java arithmetic-expr1 [NOT] BETWEEN arithmetic-expr2 AND arithmetic-expr3
  o "age BETWEEN 15 AND 19" "age >= 15 AND age <= 19"
  o "age NOT BETWEEN 15 AND 19" "age < 15 OR age > 19"

identifier [NOT] IN (string-literal1, string-literal2,...)
  identifier String NULL
  o "Country IN ('UK', 'US', 'France')" 'UK' true 'Peru' false
  (Country = 'UK') OR (Country = 'US') OR (Country = 'France')
  o "Country NOT IN ('UK', 'US', 'France')" 'UK' false 'Peru' true
  "NOT ((Country = 'UK') OR (Country = 'US') OR (Country = 'France'))
  o IN NOT IN NULL

identifier [NOT] LIKE pattern-value [ESCAPE escape-character]
  identifier String pattern-value _
  'pattern-value' escape-character pattern-value _
  'pattern-value'
  o "phone LIKE '12%3'" '123' '12993' true '1234'
false
- "word LIKE 'l_se'"  'lose' true  'loose' false
- "underscored LIKE '_\%' ESCAPE '\'"  '_foo' true  'bar' false
- "phone NOT LIKE '12%3'"  '123'  '12993' false  '1234' true
- LIKE NOT LIKE identifier NULL
- identifier IS NULL null
- "prop_name IS NULL"
- identifier IS NOT NULL null
- "prop_name IS NOT NULL"

JMS     JMSExceptionJMS

car blue 2500
"JMSType = 'car' AND color = 'blue' AND weight > 2500"

Null

NULL NULL SQL92 NULL

SQL  NULL ""

IS NULL IS NOT NULL TRUE FALSE

AND

<table>
<thead>
<tr>
<th>AND</th>
<th>T</th>
<th>F</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>F</td>
<td>U</td>
</tr>
<tr>
<td>F</td>
<td>F</td>
<td>F</td>
<td>F</td>
</tr>
<tr>
<td>U</td>
<td>U</td>
<td>F</td>
<td>U</td>
</tr>
</tbody>
</table>

OR

<table>
<thead>
<tr>
<th>OR</th>
<th>T</th>
<th>F</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>F</td>
<td>U</td>
</tr>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>T</td>
</tr>
<tr>
<td>U</td>
<td>T</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

+---------------------------+
<table>
<thead>
<tr>
<th>NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>U</td>
</tr>
</tbody>
</table>
+---------------------------+

JMSDeliveryMode 'PERSISTENT' 'NON_PERSISTENT'

long java.util.Calendar

SQL JMS

SQL

<table>
<thead>
<tr>
<th>version</th>
<th>1.1 April 2, 2002</th>
</tr>
</thead>
</table>

See also

receive(), receive(long), receiveNoWait(),

onMessage(Message), javax.jms.BytesMessage,

javax.jms.MapMessage, javax.jms.ObjectMessage,

javax.jms.StreamMessage, javax.jms.TextMessage

The Message interface is the root interface of all JMS messages. It defines the message header and the acknowledge method used for all messages.

Most message-oriented middleware (MOM) products treat messages as lightweight entities that consist of a header and a payload. The header contains fields used for message routing and identification; the payload contains the application data being sent.
Within this general form, the definition of a message varies significantly across products. It would be quite difficult for the JMS API to support all of these message models.

With this in mind, the JMS message model has the following goals:

- Provide a single, unified message API
- Provide an API suitable for creating messages that match the format used by provider-native messaging applications
- Support the development of heterogeneous applications that span operating systems, machine architectures, and computer languages
- Support messages containing objects in the Java programming language ("Java objects")
- Support messages containing Extensible Markup Language (XML) pages

JMS messages are composed of the following parts:

- Header - All messages support the same set of header fields. Header fields contain values used by both clients and providers to identify and route messages.
- Properties - Each message contains a built-in facility for supporting application-defined property values. Properties provide an efficient mechanism for supporting application-defined message filtering.
- Body - The JMS API defines several types of message body, which cover the majority of messaging styles currently in use.

**Message Bodies**

The JMS API defines five types of message body:

- **Stream** - A StreamMessage object's message body contains a stream of primitive values in the Java programming language ("Java primitives"). It is filled and read sequentially.
- **Map** - A MapMessage object's message body contains a set of name-value pairs, where names are String objects, and values are Java primitives. The entries can be accessed sequentially or
randomly by name. The order of the entries is undefined.

- **Text** - A `TextMessage` object's message body contains a `java.lang.String` object. This message type can be used to transport plain-text messages, and XML messages.
- **Object** - An `ObjectMessage` object's message body contains a `Serializable` Java object.
- **Bytes** - A `BytesMessage` object's message body contains a stream of uninterpreted bytes. This message type is for literally encoding a body to match an existing message format. In many cases, it is possible to use one of the other body types, which are easier to use. Although the JMS API allows the use of message properties with byte messages, they are typically not used, since the inclusion of properties may affect the format.

### Message Headers

The `JMSCorrelationID` header field is used for linking one message with another. It typically links a reply message with its requesting message.

`JMSCorrelationID` can hold a provider-specific message ID, an application-specific `String` object, or a provider-native `byte[]` value.

### Message Properties

A `Message` object contains a built-in facility for supporting application-defined property values. In effect, this provides a mechanism for adding application-specific header fields to a message.

Properties allow an application, via message selectors, to have a JMS provider select, or filter, messages on its behalf using application-specific criteria.

Property names must obey the rules for a message selector identifier. Property names must not be null, and must not be empty strings. If a property name is set and it is either null or an empty string, an `IllegalArgumentException` must be thrown.

Property values can be `boolean`, `byte`, `short`, `int`, `long`, `float`, `double`,
and String.

Property values are set prior to sending a message. When a client receives a message, its properties are in read-only mode. If a client attempts to set properties at this point, a MessageNotAllowedWriteableException is thrown. If clearProperties is called, the properties can now be both read from and written to. Note that header fields are distinct from properties. Header fields are never in read-only mode.

A property value may duplicate a value in a message's body, or it may not. Although JMS does not define a policy for what should or should not be made a property, application developers should note that JMS providers will likely handle data in a message's body more efficiently than data in a message's properties. For best performance, applications should use message properties only when they need to customize a message's header. The primary reason for doing this is to support customized message selection.

Message properties support the following conversion table. The marked cases must be supported. The unmarked cases must throw a JMSException. The String-to-primitive conversions may throw a runtime exception if the primitive's valueOf method does not accept the String as a valid representation of the primitive.

A value written as the row type can be read as the column type.

<table>
<thead>
<tr>
<th></th>
<th>boolean</th>
<th>byte</th>
<th>short</th>
<th>int</th>
<th>long</th>
<th>float</th>
<th>double</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>short</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>long</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>double</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

In addition to the type-specific set/get methods for properties, JMS provides the setObjectProperty and getObjectProperty methods.
These support the same set of property types using the objectified primitive values. Their purpose is to allow the decision of property type to be made at execution time rather than at compile time. They support the same property value conversions.

The `setObjectProperty` method accepts values of class `Boolean`, `Byte`, `Short`, `Integer`, `Long`, `Float`, `Double`, and `String`. An attempt to use any other class must throw a `JMSException`.

The `getObjectProperty` method only returns values of class `Boolean`, `Byte`, `Short`, `Integer`, `Long`, `Float`, `Double`, and `String`.

The order of property values is not defined. To iterate through a message's property values, use `getPropertyNames` to retrieve a property name enumeration and then use the various property get methods to retrieve their values.

A message's properties are deleted by the `clearProperties` method. This leaves the message with an empty set of properties.

Getting a property value for a name which has not been set returns a null value. Only the `getStringProperty` and `getObjectProperty` methods can return a null value. Attempting to read a null value as a primitive type must be treated as calling the primitive's corresponding `valueOf(String)` conversion method with a null value.

The JMS API reserves the `JMSX` property name prefix for JMS defined properties. The full set of these properties is defined in the Java Message Service specification. New JMS defined properties may be added in later versions of the JMS API. Support for these properties is optional. The `String[] ConnectionMetaData.getJMSXPropertyNames` method returns the names of the JMSX properties supported by a connection.

JMSX properties may be referenced in message selectors whether or not they are supported by a connection. If they are not present in a message, they are treated like any other absent property.

JMSX properties defined in the specification as "set by provider on send" are available to both the producer and the consumers of the
message. JMSX properties defined in the specification as "set by provider on receive" are available only to the consumers.

JMSXGroupID and JMSXGroupSeq are standard properties that clients should use if they want to group messages. All providers must support them. Unless specifically noted, the values and semantics of the JMSX properties are undefined.

The JMS API reserves the JMS_vendor_name property name prefix for provider-specific properties. Each provider defines its own value for vendor_name. This is the mechanism a JMS provider uses to make its special per-message services available to a JMS client.

The purpose of provider-specific properties is to provide special features needed to integrate JMS clients with provider-native clients in a single JMS application. They should not be used for messaging between JMS clients.

Provider Implementations of JMS Message Interfaces

The JMS API provides a set of message interfaces that define the JMS message model. It does not provide implementations of these interfaces.

Each JMS provider supplies a set of message factories with its Session object for creating instances of messages. This allows a provider to use message implementations tailored to its specific needs.

A provider must be prepared to accept message implementations that are not its own. They may not be handled as efficiently as its own implementation; however, they must be handled.

Note the following exception case when a provider is handling a foreign message implementation. If the foreign message implementation contains a JMSReplyTo header field that is set to a foreign destination implementation, the provider is not required to handle or preserve the value of this header field.
Message Selectors

A JMS message selector allows a client to specify, by header field references and property references, the messages it is interested in. Only messages whose header and property values match the selector are delivered. What it means for a message not to be delivered depends on the MessageConsumer being used (see QueueReceiver and TopicSubscriber).

Message selectors cannot reference message body values.

A message selector matches a message if the selector evaluates to true when the message's header field values and property values are substituted for their corresponding identifiers in the selector.

A message selector is a String whose syntax is based on a subset of the SQL92 conditional expression syntax. If the value of a message selector is an empty string, the value is treated as a null and indicates that there is no message selector for the message consumer.

The order of evaluation of a message selector is from left to right within precedence level. Parentheses can be used to change this order.

Predefined selector literals and operator names are shown here in uppercase; however, they are case insensitive.

A selector can contain:

- Literals:
  - A string literal is enclosed in single quotes, with a single quote represented by doubled single quote; for example, 'literal' and 'literal''s'. Like string literals in the Java programming language, these use the Unicode character encoding.
  - An exact numeric literal is a numeric value without a decimal point, such as 57, -957, and +62; numbers in the range of long are supported. Exact numeric literals use the
integer literal syntax of the Java programming language.
- An approximate numeric literal is a numeric value in scientific notation, such as 7E3 and -57.9E2, or a numeric value with a decimal, such as 7., -95.7, and +6.2; numbers in the range of double are supported. Approximate literals use the floating-point literal syntax of the Java programming language.
- The boolean literals true and false.

Identifiers:
- An identifier is an unlimited-length sequence of letters and digits, the first of which must be a letter. A letter is any character for which the method Character.isJavaLetter returns true. This includes '_.' and '$'. A letter or digit is any character for which the method Character.isJavaLetterOrDigit returns true.
- Identifiers cannot be the names NULL, TRUE, and FALSE.
- Identifiers cannot be NOT, AND, OR, BETWEEN, LIKE, IN, IS, OR ESCAPE.
- Identifiers are either header field references or property references. The type of a property value in a message selector corresponds to the type used to set the property. If a property that does not exist in a message is referenced, its value is NULL.
- The conversions that apply to the get methods for properties do not apply when a property is used in a message selector expression. For example, suppose you set a property as a string value, as in the following:

```
myMessage.setStringProperty("NumberOfOrders", "2");
```

The following expression in a message selector would evaluate to false, because a string cannot be used in an arithmetic expression:

"NumberOfOrders > 1"

- Identifiers are case-sensitive.
- Message header field references are restricted to JMSDeliveryMode, JMSPriority, JMSMessageID, JMSTimestamp, JMSCorrelationID, and JMSType. JMSMessageID,
JMSCorrelationID, and JMSType values may be null and if so are treated as a NULL value.

- Any name beginning with 'JMSX' is a JMS defined property name.
- Any name beginning with 'JMS_' is a provider-specific property name.
- Any name that does not begin with 'JMS' is an application-specific property name.

White space is the same as that defined for the Java programming language: space, horizontal tab, form feed, and line terminator.

- Expressions:
  - A selector is a conditional expression; a selector that evaluates to true matches; a selector that evaluates to false or unknown does not match.
  - Arithmetic expressions are composed of themselves, arithmetic operations, identifiers (whose value is treated as a numeric literal), and numeric literals.
  - Conditional expressions are composed of themselves, comparison operations, and logical operations.
- Standard bracketing () for ordering expression evaluation is supported.
- Logical operators in precedence order: NOT, AND, OR
- Comparison operators: =, >, >=, <, <=, (not equal)
  - Only like type values can be compared. One exception is that it is valid to compare exact numeric values and approximate numeric values; the type conversion required is defined by the rules of numeric promotion in the Java programming language. If the comparison of non-like type values is attempted, the value of the operation is false. If either of the type values evaluates to NULL, the value of the expression is unknown.
  - String and boolean comparison is restricted to = and !=. Two strings are equal if and only if they contain the same sequence of characters.
- Arithmetic operators in precedence order:
  - +, - (unary)
  - *, / (multiplication and division)
  - +, - (addition and subtraction)
- Arithmetic operations must use numeric promotion in the Java programming language.

- `arithmetic-expr1 [NOT] BETWEEN arithmetic-expr2 AND arithmetic-expr3` (comparison operator)
  - "age BETWEEN 15 AND 19" is equivalent to "age >= 15 AND age <= 19"
  - "age NOT BETWEEN 15 AND 19" is equivalent to "age < 15 OR age > 19"

- `identifier [NOT] IN (string-literal1, string-literal2,...)` (comparison operator where `identifier` has a String or NULL value)
  - "Country IN ('UK', 'US', 'France')" is true for 'UK' and false for 'Peru'; it is equivalent to the expression "(Country = 'UK') OR (Country = 'US') OR (Country = 'France')"
  - "Country NOT IN ('UK', 'US', 'France')" is false for 'UK' and true for 'Peru'; it is equivalent to the expression "NOT ((Country = 'UK') OR (Country = 'US') OR (Country = 'France'))"
  - If `identifier` of an IN or NOT IN operation is NULL, the value of the operation is unknown.

- `identifier [NOT] LIKE pattern-value [ESCAPE escape-character]` (comparison operator, where `identifier` has a String value; `pattern-value` is a string literal where '_' stands for any single character; '%' stands for any sequence of characters, including the empty sequence; and all other characters stand for themselves. The optional `escape-character` is a single-character string literal whose character is used to escape the special meaning of the '_' and '%' in `pattern-value`.)
  - "phone LIKE '12%3'" is true for '123' or '12993' and false for '1234'
  - "word LIKE 'l_se'" is true for 'lose' and false for 'loose'
  - "underscored LIKE '_%' ESCAPE '\'" is true for '_foo' and false for 'bar'
  - "phone NOT LIKE '12%3'" is false for '123' or '12993' and true for '1234'
  - If `identifier` of a LIKE or NOT LIKE operation is NULL, the value of the operation is unknown.

- `identifier IS NULL` (comparison operator that tests for a null header field value or a missing property value)
- "prop_name IS NULL"
- identifier IS NOT NULL (comparison operator that tests for the existence of a non-null header field value or a property value)
- "prop_name IS NOT NULL"

JMS providers are required to verify the syntactic correctness of a message selector at the time it is presented. A method that provides a syntactically incorrect selector must result in a JMSException. JMS providers may also optionally provide some semantic checking at the time the selector is presented. Not all semantic checking can be performed at the time a message selector is presented, because property types are not known.

The following message selector selects messages with a message type of car and color of blue and weight greater than 2500 pounds:

"JMSType = 'car' AND color = 'blue' AND weight > 2500"

Null Values

As noted above, property values may be NULL. The evaluation of selector expressions containing NULL values is defined by SQL92 NULL semantics. A brief description of these semantics is provided here.

SQL treats a NULL value as unknown. Comparison or arithmetic with an unknown value always yields an unknown value.

The IS NULL and IS NOT NULL operators convert an unknown value into the respective TRUE and FALSE values.

The boolean operators use three-valued logic as defined by the following tables:

**The definition of the AND operator**

<table>
<thead>
<tr>
<th></th>
<th>AND</th>
<th>T</th>
<th>F</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T</td>
<td>T</td>
<td>F</td>
<td>U</td>
</tr>
</tbody>
</table>
The definition of the **OR** operator

<table>
<thead>
<tr>
<th>OR</th>
<th>T</th>
<th>F</th>
<th>U</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
<td>T</td>
<td>T</td>
<td>T</td>
</tr>
<tr>
<td>F</td>
<td>T</td>
<td>F</td>
<td>U</td>
</tr>
<tr>
<td>U</td>
<td>T</td>
<td>U</td>
<td>U</td>
</tr>
</tbody>
</table>

The definition of the **NOT** operator

<table>
<thead>
<tr>
<th>NOT</th>
</tr>
</thead>
<tbody>
<tr>
<td>T</td>
</tr>
<tr>
<td>F</td>
</tr>
<tr>
<td>T</td>
</tr>
<tr>
<td>U</td>
</tr>
</tbody>
</table>

Special Notes

When used in a message selector, the JMSDeliveryMode header field is treated as having the values 'PERSISTENT' and 'NON_PERSISTENT'.

Date and time values should use the standard long millisecond value. When a date or time literal is included in a message selector, it should be an integer literal for a millisecond value. The standard way to produce millisecond values is to use java.util.Calendar.

Although SQL supports fixed decimal comparison and arithmetic, JMS message selectors do not. This is the reason for restricting exact numeric literals to those without a decimal (and the addition of numerics with a decimal as an alternate representation for approximate numeric values).
SQL comments are not supported.

Version:
1.1 April 2, 2002

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
MessageConsumer.receive(), MessageConsumer.receive(long), MessageConsumer.receiveNoWait(), MessageListener.onMessage(Message), BytesMessage, MapMessage, ObjectMessage, StreamMessage, TextMessage

### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>DEFAULT_DELIVERY_MODE</td>
<td>The message producer's default delivery mode is PERSISTENT.</td>
</tr>
<tr>
<td>static int</td>
<td>DEFAULT_PRIORITY</td>
<td>The message producer's default priority is 4.</td>
</tr>
<tr>
<td>static long</td>
<td>DEFAULT_TIME_TO_LIVE</td>
<td>The message producer's default time to live is unlimited; the message never expires.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>acknowledge()</td>
<td>Acknowledges all consumed messages of the session of this consumed message.</td>
</tr>
<tr>
<td>void</td>
<td>clearBody()</td>
<td>Clears out the message body.</td>
</tr>
<tr>
<td>void</td>
<td>clearProperties()</td>
<td>Clears a message's properties.</td>
</tr>
<tr>
<td>boolean</td>
<td>getBooleanProperty(String name)</td>
<td>Returns the value of the boolean property with the specified name.</td>
</tr>
<tr>
<td>Type</td>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>byte</td>
<td><code>getByteProperty(String name)</code></td>
<td>Returns the value of the byte property with the specified name.</td>
</tr>
<tr>
<td>double</td>
<td><code>getDoubleProperty(String name)</code></td>
<td>Returns the value of the double property with the specified name.</td>
</tr>
<tr>
<td>float</td>
<td><code>getFloatProperty(String name)</code></td>
<td>Returns the value of the float property with the specified name.</td>
</tr>
<tr>
<td>int</td>
<td><code>getIntProperty(String name)</code></td>
<td>Returns the value of the int property with the specified name.</td>
</tr>
<tr>
<td>String</td>
<td><code>getJMSCorrelationID()</code></td>
<td>Gets the correlation ID for the message.</td>
</tr>
<tr>
<td>byte[]</td>
<td><code>getJMSCorrelationIDsAsBytes()</code></td>
<td>Gets the correlation ID as an array of bytes for the message.</td>
</tr>
<tr>
<td>int</td>
<td><code>getJMSDeliveryMode()</code></td>
<td>Gets the DeliveryMode value specified for this message.</td>
</tr>
<tr>
<td>Destination</td>
<td><code>getJMSDestination()</code></td>
<td>Gets the Destination object for this message.</td>
</tr>
<tr>
<td>long</td>
<td><code>getJMSExpiration()</code></td>
<td>Gets the message's expiration value.</td>
</tr>
<tr>
<td>String</td>
<td><code>getJMSMessageID()</code></td>
<td>Gets the message ID.</td>
</tr>
<tr>
<td>int</td>
<td><code>getJMSPriority()</code></td>
<td>Gets the message priority level.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>getJMSRedelivered()</code></td>
<td>Gets an indication of whether this message is being redelivered.</td>
</tr>
<tr>
<td>Destination</td>
<td><code>getJMSReplyTo()</code></td>
<td>Gets the Destination object to which a reply to this message should be sent.</td>
</tr>
<tr>
<td>long</td>
<td><code>getJMSTimestamp()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><code>String getJMSType()</code></td>
<td>Gets the message type identifier supplied by the client when the message was sent.</td>
<td></td>
</tr>
<tr>
<td><code>long getLongProperty(String name)</code></td>
<td>Returns the value of the long property with the specified name.</td>
<td></td>
</tr>
<tr>
<td><code>Object getObjectProperty(String name)</code></td>
<td>Returns the value of the Java object property with the specified name.</td>
<td></td>
</tr>
<tr>
<td><code>Enumeration getPropertyNames()</code></td>
<td>Returns an Enumeration of all the property names.</td>
<td></td>
</tr>
<tr>
<td><code>short getShortProperty(String name)</code></td>
<td>Returns the value of the short property with the specified name.</td>
<td></td>
</tr>
<tr>
<td><code>String getStringProperty(String name)</code></td>
<td>Returns the value of the String property with the specified name.</td>
<td></td>
</tr>
<tr>
<td><code>boolean propertyExists(String name)</code></td>
<td>Indicates whether a property value exists.</td>
<td></td>
</tr>
<tr>
<td><code>void setBooleanProperty(String name, boolean value)</code></td>
<td>Sets a boolean property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>void setByteProperty(String name, byte value)</code></td>
<td>Sets a byte property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>void setDoubleProperty(String name, double value)</code></td>
<td>Sets a double property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>void setFloatProperty(String name, float value)</code></td>
<td>Sets a float property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>void setIntProperty(String name, int value)</code></td>
<td>Sets an int property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>setJMSCorrelationID(String correlationID)</code></td>
<td>Sets the correlation ID for the message.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSCorrelationIDAsBytes(byte[] correlationID)</code></td>
<td>Sets the correlation ID as an array of bytes for the message.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSDeliveryMode(int deliveryMode)</code></td>
<td>Sets the DeliveryMode value for this message.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSDestination(Destination destination)</code></td>
<td>Sets the Destination object for this message.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSExpiration(long expiration)</code></td>
<td>Sets the message's expiration value.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSMessageID(String id)</code></td>
<td>Sets the message ID.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSPriority(int priority)</code></td>
<td>Sets the priority level for this message.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSRedelivered(boolean redelivered)</code></td>
<td>Specifies whether this message is being redelivered.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSReplyTo(Destination replyTo)</code></td>
<td>Sets the Destination object to which a reply to this message should be sent.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSTimestamp(long timestamp)</code></td>
<td>Sets the message timestamp.</td>
<td></td>
</tr>
<tr>
<td><code>setJMSType(String type)</code></td>
<td>Sets the message type.</td>
<td></td>
</tr>
<tr>
<td><code>setLongProperty(String name, long value)</code></td>
<td>Sets a long property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>setObjectProperty(String name, Object value)</code></td>
<td>Sets a Java object property value with the specified name into the message.</td>
<td></td>
</tr>
<tr>
<td><code>setShortProperty(String name, short value)</code></td>
<td>Sets a short property value with the specified name into the message.</td>
<td></td>
</tr>
</tbody>
</table>
**Field Detail**

**DEFAULT_DELIVERY_MODE**

```java
static final int DEFAULT_DELIVERY_MODE
```

The message producer's default delivery mode is PERSISTENT.

**See Also:**

[DeliveryMode.PERSISTENT, Constant Field Values](#)

**DEFAULT_PRIORITY**

```java
static final int DEFAULT_PRIORITY
```

The message producer's default priority is 4.

**See Also:**

[Constant Field Values](#)

**DEFAULT_TIME_TO_LIVE**

```java
static final long DEFAULT_TIME_TO_LIVE
```
The message producer's default time to live is unlimited; the message never expires.

See Also:  
Constant Field Values

### Method Detail

```java
public String getJMSMessageID() throws JMSException

ID

JMSMessageID

JMSMessageID send publish

JMSMessageID String

JMSMessageID 'ID:' ID

ID ID JMS
MessageProducer.setDisableMessageID JMS
JMS ID null ID

return ID

Throws JMSException: JMS ID

See setJMSMessageID(String),
also setDisableMessageID(boolean)
```

**getJMSMessageID**

```java
String getJMSMessageID() throws JMSException

Gets the message ID.
```
The JMSMessageID header field contains a value that uniquely identifies each message sent by a provider.

When a message is sent, JMSMessageID can be ignored. When the send or publish method returns, it contains a provider-assigned value.

A JMSMessageID is a String value that should function as a unique key for identifying messages in a historical repository. The exact scope of uniqueness is provider-defined. It should at least cover all messages for a specific installation of a provider, where an installation is some connected set of message routers.

All JMSMessageID values must start with the prefix 'ID:'. Uniqueness of message ID values across different providers is not required.

Since message IDs take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that the message ID is not used by an application. By calling the MessageProducer.setDisableMessageID method, a JMS client enables this potential optimization for all messages sent by that message producer. If the JMS provider accepts this hint, these messages must have the message ID set to null; if the provider ignores the hint, the message ID must be set to its normal unique value.

Returns:
  the message ID

Throws:
  JMSException - if the JMS provider fails to get the message ID due to some internal error.

See Also:
  setJMSMessageID(String),
  MessageProducer.setDisableMessageID(boolean)

public void setJMSMessageID(String id) throws
**setJMSMessageID**

```java
void setJMSMessageID(String id)
```

Sets the message ID.

JMS providers set this field when a message is sent. This method can be used to change the value for a message that has been received.

**Parameters:**

- `id` - the ID of the message

**Throws:**

- `JMSException` - if the JMS provider fails to set the message ID due to some internal error.

**See Also:**

- `getJMSMessageID()`

---

**public long getJMSTimestamp() throws JMSException**
getJMSTimestamp

long getJMSTimestamp() throws JMSException

Gets the message timestamp.

The JMSTimestamp header field contains the time a message was handed off to a provider to be sent. It is not the time the message was actually transmitted, because the actual send may occur later due to transactions or other client-side queueing of messages.

When a message is sent, JMSTimestamp is ignored. When the send or publish method returns, it contains a time value somewhere in the interval between the call and the return. The value is in the format of a normal millis time value in the Java programming language.

Since timestamps take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that the timestamp is not used by an application. By calling the MessageProducer.setDisableMessageTimestamp method, a JMS client enables this potential optimization for all messages sent by that message producer. If the JMS provider accepts this hint, these messages must have the timestamp set to zero; if the provider ignores the hint, the timestamp must be set to its normal value.
public void setJMSTimestamp(long timestamp)
throws JMSException

Sets the message timestamp.

JMS providers set this field when a message is sent. This method can be used to change the value for a message that has been received.

Parameters:

timestamp - the timestamp for this message

Throws:

JMSException - if the JMS provider fails to set the timestamp due to some internal error.

See Also:

getJMSTimestamp()
public byte[] getJMSCorrelationIDAsBytes() throws JMSException
byte ID

JMSCorrelationID byte[]
return byte ID
Throws JMSException: JMS ID
See setJMSCorrelationID(String), getJMSCorrelationID(),
also setJMSCorrelationIDAsBytes(byte[])

getJMSCorrelationIDAsBytes

byte[] getJMSCorrelationIDAsBytes() throws JMSException

Gets the correlation ID as an array of bytes for the message.

The use of a byte[] value for JMSCorrelationID is non-portable.

Returns: the correlation ID of a message as an array of bytes
Throws: JMSException - if the JMS provider fails to get the correlation ID due to some internal error.
See Also: setJMSCorrelationID(String), getJMSCorrelationID(),
setJMSCorrelationIDAsBytes(byte[])

public void setJMSCorrelationIDAsBytes(byte[] correlationID) throws JMSException
byte ID
setJMSCorrelationIDAsBytes

void setJMSCorrelationIDAsBytes(byte[] correlationID) throws JMSException

Sets the correlation ID as an array of bytes for the message.

The array is copied before the method returns, so future modifications to the array will not alter this message header.

If a provider supports the native concept of correlation ID, a JMS client may need to assign specific JMSCorrelationID values to match those expected by native messaging clients. JMS providers without native correlation ID values are not required to support this method and its corresponding get method; their implementation may throw a java.lang.UnsupportedOperationException.

The use of a byte[] value for JMSCorrelationID is non-portable.

Parameters:
- correlationID - the correlation ID value as an array of bytes

Throws:
- JMSException - if the JMS provider fails to set the
setJMSCorrelationID

void setJMSCorrelationID(String correlationID)
Sets the correlation ID for the message.

A client can use the JMSCorrelationID header field to link one message with another. A typical use is to link a response message with its request message.

JMSCorrelationID can hold one of the following:

- A provider-specific message ID
- An application-specific String
- A provider-native byte[] value

Since each message sent by a JMS provider is assigned a message ID value, it is convenient to link messages via message ID. All message ID values must start with the 'ID:' prefix.

In some cases, an application (made up of several clients) needs to use an application-specific value for linking messages. For instance, an application may use JMSCorrelationID to hold a value referencing some external information. Application-specified values must not start with the 'ID:' prefix; this is reserved for provider-generated message ID values.

If a provider supports the native concept of correlation ID, a JMS client may need to assign specific JMSCorrelationID values to match those expected by clients that do not use the JMS API. A byte[] value is used for this purpose. JMS providers without native correlation ID values are not required to support byte[] values. The use of a byte[] value for JMSCorrelationID is non-portable.

**Parameters:**

- correlationID - the message ID of a message being referred to

**Throws:**

JMSException - if the JMS provider fails to set the
correlation ID due to some internal error.

See Also:
getJMSCorrelationID(), getJMSCorrelationIDAsBytes(), setJMSCorrelationIDAsBytes(byte[])

public String getJMSCorrelationID() throws JMException

ID

String ID

return String ID

Throws JMSException: JMS ID

See also setJMSCorrelationID(String), getJMSCorrelationIDAsBytes(), setJMSCorrelationIDAsBytes(byte[])

getJMSCorrelationID

String getJMSCorrelationID() throws JMSException

Gets the correlation ID for the message.

This method is used to return correlation ID values that are either provider-specific message IDs or application-specific String values.

Returns:
the correlation ID of a message as a String

Throws:
JMException - if the JMS provider fails to get the correlation ID due to some internal error.

See Also:
setJMSCorrelationID(String), getJMSCorrelationIDAsBytes(), setJMSCorrelationIDAsBytes(byte[])
public **Destination** getJMSReplyTo() throws **JMSException**

    Destination
    return
    Throws **JMSException**: JMS
    See also **setJMSReplyTo(Destination)**

getJMSReplyTo

**Destination** getJMSReplyTo() throws **JMSException**

    Gets the **destination** object to which a reply to this message should be sent.

    **Returns**: Destination to which to send a response to this message

    **Throws**: **JMSException** - if the JMS provider fails to get the JMSReplyTo destination due to some internal error.

    **See Also**: **setJMSReplyTo(Destination)**

-------------------

public void setJMSReplyTo(**Destination** replyTo) throws **JMSException**

    Destination
    JMSReplyTo **null** Queue Topic
    **null** JMSReplyTo

    JMSReplyTo

    **JMSCorrelationID**
    **replyTo** Destination
Throws  

See also

getJMSReplyTo()

setJMSReplyTo

void setJMSReplyTo(Destination replyTo)

sets the Destination object to which a reply to this message should be sent.

The JMSReplyTo header field contains the destination where a reply to the current message should be sent. If it is null, no reply is expected. The destination may be either a Queue object or a Topic object.

Messages sent with a null JMSReplyTo value may be a notification of some event, or they may just be some data the sender thinks is of interest.

Messages with a JMSReplyTo value typically expect a response. A response is optional; it is up to the client to decide. These messages are called requests. A message sent in response to a request is called a reply.

In some cases a client may wish to match a request it sent earlier with a reply it has just received. The client can use the JMSCorrelationID header field for this purpose.

Parameters:

replyTo - Destination to which to send a response to this message

Throws:

JMSException - if the JMS provider fails to set the JMSReplyTo destination due to some internal error.

See Also:

getJMSReplyTo()
public Destination getJMSDestination() throws JMSException

  Destination

JMSDestination

  send    publish

  JMSDestination
  return

  Throws   JMSException: JMS
  See also    setJMSDestination(Destination)

getJMSDestination

  Destination getJMSDestination()    throws JMSException

  Gets the Destination object for this message.

  The JMSDestination header field contains the destination to which the message is being sent.

  When a message is sent, this field is ignored. After completion of the send or publish method, the field holds the destination specified by the method.

  When a message is received, its JMSDestination value must be equivalent to the value assigned when it was sent.

  Returns: the destination of this message

  Throws:   JMSException - if the JMS provider fails to get the destination due to some internal error.

  See Also: setJMSDestination(Destination)
public void setJMSDestination(Destination destination) throws JMSException

Sets the Destination object for this message.

JMS providers set this field when a message is sent. This method can be used to change the value for a message that has been received.

Parameters:
- destination - the destination for this message

Throws:
- JMSException - if the JMS provider fails to set the destination due to some internal error.

See Also:
- getJMSDestination()

public int getJMSDeliveryMode() throws JMSException

DeliveryMode

return

Throws JMSException: JMS

See also setJMSDeliveryMode(int), javax.jms.DeliveryMode
getJMSDeliveryMode

```java
int getJMSDeliveryMode() throws JMSException
```

Gets the DeliveryMode value specified for this message.

**Returns:**
the delivery mode for this message

**Throws:**
JMSException - if the JMS provider fails to get the delivery mode due to some internal error.

**See Also:**
setJMSDeliveryMode(int), javax.jms.DeliveryMode

---

public void setJMSDeliveryMode(int deliveryMode)
throws JMSException

JMS

```java
deliveryMode
```

**Throws**
JMSException: JMS

**See also**
getJMSDeliveryMode(), javax.jms.DeliveryMode

---

setJMSDeliveryMode

```java
void setJMSDeliveryMode(int deliveryMode) throws JMSException
```

Sets the DeliveryMode value for this message.

JMS providers set this field when a message is sent. This method can be used to change the value for a message that has been received.
Parameters:
- deliveryMode - the delivery mode for this message

Throws:
- JMSException - if the JMS provider fails to set the delivery mode due to some internal error.

See Also:
- getJMSDeliveryMode(), DeliveryMode

public boolean getJMSRedelivered() throws JMSException

JMSRedelivered

    return true

Throws
- JMSException: JMS

See also
- setJMSRedelivered(boolean)

getJMSRedelivered

boolean getJMSRedelivered() throws JMSException

Gets an indication of whether this message is being redelivered.

If a client receives a message with the JMSRedelivered field set, it is likely, but not guaranteed, that this message was delivered earlier but that its receipt was not acknowledged at that time.

Returns:
- true if this message is being redelivered

Throws:
- JMSException - if the JMS provider fails to get the redelivered state due to some internal error.

See Also:
setJMSRedelivered( boolean )

public void setJMSRedelivered( boolean redelivered ) throws JMSException

  redelivered
  Throws JMSException: JMS
  See also getJMSRedelivered()

setJMSRedelivered

void setJMSRedelivered( boolean redelivered )
  throws JMSException

  Specifies whether this message is being redelivered.

  This field is set at the time the message is delivered. This method can be used to change the value for a message that has been received.

Parameters:
  redelivered - an indication of whether this message is being redelivered

Throws:
  JMSException - if the JMS provider fails to set the redelivered state due to some internal error.

See Also:
  getJMSRedelivered()

public String getJMSType() throws JMSException

  return
  Throws JMSException: JMS
getJMSType

`String getJMSType()`

Throws: `JMSException` - if the JMS provider fails to get the message type due to some internal error.

See Also: `setJMSType(String)`

public void setJMSType(String type) throws `JMSException`

setJMSType
void setJMSType(String type) throws JMSException

Sets the message type.

Some JMS providers use a message repository that contains the definitions of messages sent by applications. The JMSType header field may reference a message's definition in the provider's repository.

The JMS API does not define a standard message definition repository, nor does it define a naming policy for the definitions it contains.

Some messaging systems require that a message type definition for each application message be created and that each message specify its type. In order to work with such JMS providers, JMS clients should assign a value to JMSType, whether the application makes use of it or not. This ensures that the field is properly set for those providers that require it.

To ensure portability, JMS clients should use symbolic values for JMSType that can be configured at installation time to the values defined in the current provider's message repository. If string literals are used, they may not be valid type names for some JMS providers.

Parameters:
  type - the message type

Throws:
  JMSException - if the JMS provider fails to set the message type due to some internal error.

See Also:
  getJMSType()

public long getJMSExpiration() throws JMSException
JMSExpiration

send	publish

JMSExpiration

GMT

0

JMSAPI

getJMSExpiration

long getJMSExpiration()

return GMT

Throws JMSException: JMS

See also setJMSExpiration(long)

getJMSExpiration

long getJMSExpiration()

Throws JMSException

Gets the message's expiration value.

When a message is sent, the JMSExpiration header field is left unassigned. After completion of the send or publish method, it holds the expiration time of the message. This is the sum of the time-to-live value specified by the client and the GMT at the time of the send or publish.

If the time-to-live is specified as zero, JMSExpiration is set to zero to indicate that the message does not expire.

When a message's expiration time is reached, a provider should discard it. The JMS API does not define any form of notification of message expiration.

Clients should not receive messages that have expired; however, the JMS API does not guarantee that this will not happen.

Returns:
the time the message expires, which is the sum of the
time-to-live value specified by the client and the GMT
at the time of the send

Throws:

JMSException - if the JMS provider fails to get the
message expiration due to some internal error.

See Also:

setJMSExpiration(long)

public void setJMSExpiration(long expiration)
throws JMSException

JMS

expiration

Throws JMSException: JMS

See also getJMSExpiration()

setJMSExpiration

void setJMSExpiration(long expiration)
throws JMSException

Sets the message's expiration value.

JMS providers set this field when a message is sent. This
method can be used to change the value for a message
that has been received.

Parameters:

expiration - the message's expiration time

Throws:

JMSException - if the JMS provider fails to set the
message expiration due to some internal error.

See Also:

getJMSExpiration()
public int getJMSPriority() throws JMSException

JMS API 10 0 9 0-4 5-9

JMS API

    return

    Throws JMSException: JMS

    See also setJMSPriority(int)

getJMSPriority

int getJMSPriority() throws JMSException

Gets the message priority level.

The JMS API defines ten levels of priority value, with 0 as the lowest priority and 9 as the highest. In addition, clients should consider priorities 0-4 as gradations of normal priority and priorities 5-9 as gradations of expedited priority.

The JMS API does not require that a provider strictly implement priority ordering of messages; however, it should do its best to deliver expedited messages ahead of normal messages.

Returns:
the default message priority

Throws:
JMSException - if the JMS provider fails to get the message priority due to some internal error.

See Also:
setJMSPriority(int)
public void setJMSPriority(int priority) throws JMSException

JMS

    priority

Throws JMSException: JMS
See also getJMSPriority()

setJMSPriority

void setJMSPriority(int priority)
    throws JMSException

Sets the priority level for this message.

JMS providers set this field when a message is sent. This method can be used to change the value for a message that has been received.

Parameters:
    priority - the priority of this message

Throws:
    JMSException - if the JMS provider fails to set the message priority due to some internal error.

See Also:
    getJMSPriority()

public void clearProperties() throws JMSException

Throws JMSException: JMS
clearProperties

void clearProperties() throws JMSException

Clears a message's properties.

The message's header fields and body are not cleared.

Throws:

  JMSException - if the JMS provider fails to clear the message properties due to some internal error.

public boolean propertyExists(String name) throws JMSException

  name
  return true

  Throws JMSException: JMS

propertyExists

boolean propertyExists(String name) throws JMSException

Indicates whether a property value exists.

Parameters:

  name - the name of the property to test

Returns:

  true if the property exists

Throws:

  JMSException - if the JMS provider fails to determine if the property exists due to some internal error.

public boolean getBooleanProperty(String name) throws JMSException
boolean

getBooleanProperty

boolean getBooleanProperty(String name)
throws JMSException

Returns the value of the boolean property with the specified name.

Parameters:
name - the name of the boolean property

Returns:
the boolean property value for the specified name

Throws:
JMSException - if the JMS provider fails to get the property value due to some internal error.
MessageFormatException - if this type conversion is invalid.

public byte getByteProperty(String name) throws JMSException

getByteProperty

byte getByteProperty(String name)
throws **JMSException**

Returns the value of the *byte* property with the specified name.

**Parameters:**
name - the name of the *byte* property

**Returns:**
the *byte* property value for the specified name

**Throws:**
**JMSException** - if the JMS provider fails to get the property value due to some internal error.
**MessageFormatException** - if this type conversion is invalid.

### public short getShortProperty(String name) throws **JMSException**

```
short getShortProperty(String name)
```

short

*name*

short

*return*

short

**Throws**
**JMSException**: JMS

**Throws**
**MessageFormatException**: 

### getShortProperty

```
short getShortProperty(String name)
```

short

*getShortProperty*(String *name*)

throws **JMSException**

Returns the value of the *short* property with the specified name.

**Parameters:**
name - the name of the *short* property

**Returns:**
the *short* property value for the specified name

**Throws:**
**JMSException** - if the JMS provider fails to get the
public int getIntProperty(String name) throws JMSException

int

name int
return int

Throws: JMSException: JMS
Throws: MessageFormatException: 

getIntProperty

int getIntProperty(String name) throws JMSException

Returns the value of the int property with the specified name.

Parameters:
name - the name of the int property

Returns:
the int property value for the specified name

Throws:
JMSException - if the JMS provider fails to get the property value due to some internal error.
MessageFormatException - if this type conversion is invalid.

public long getLongProperty(String name) throws JMSException

long

name long
return long
Throws: JMSException: JMS
Throws: MessageFormatException:

getLongProperty

long getLongProperty(String name) throws JMSException

Returns the value of the long property with the specified name.

Parameters:
name - the name of the long property

Returns:
the long property value for the specified name

Throws:
JMSException - if the JMS provider fails to get the property value due to some internal error.
MessageFormatException - if this type conversion is invalid.

public float getFloatProperty(String name) throws JMSException

float

name float
return float

Throws: JMSException: JMS
Throws: MessageFormatException:

getFloatProperty

float getFloatProperty(String name) throws JMSException

Returns the value of the float property with the specified name.
public double getDoubleProperty(String name) throws JMSException

Parameters:
name - the name of the double property

Returns:
the double property value for the specified name

Throws:
JMSException - if the JMS provider fails to get the property value due to some internal error.
MessageFormatException - if this type conversion is invalid.

getDoubleProperty

double getDoubleProperty(String name) throws JMSException

Parameters:
name - the name of the double property

Returns:
the double property value for the specified name

Throws:
JMSException - if the JMS provider fails to get the property value due to some internal error.
MessageFormatException - if this type conversion is invalid.
public String getStringProperty(String name) throws JMSException

    String
    name
    return

Throws: JMSException: JMS

getStringProperty

String getStringProperty(String name) throws JMSException

    String
    name
    return

Returns the value of the String property with the specified name.

Parameters:
    name - the name of the String property

Returns:
    the String property value for the specified name; if there is no property by this name, a null value is returned

Throws:
    JMSException - if the JMS provider fails to get the property value due to some internal error.
    MessageFormatException - if this type conversion is invalid.

public Object getObjectProperty(String name) throws JMSException

Java

    setObjectProperty
    name
    return

Throws: JMSException: JMS
**getObjectProperty**

```java
Object getObjectProperty(String name) throws JMSException
```

Returns the value of the Java object property with the specified name.

This method can be used to return, in objectified format, an object that has been stored as a property in the message with the equivalent `setObjectProperty` method call, or its equivalent primitive `setProperty` method.

**Parameters:**

- name - the name of the Java object property

**Returns:**

- the Java object property value with the specified name, in objectified format (for example, if the property was set as an `int`, an `Integer` is returned); if there is no property by this name, a null value is returned

**Throws:**

- `JMSException` - if the JMS provider fails to get the property value due to some internal error.

---

```java
public java.util.Enumeration<E> getPropertyNames() throws JMSException
```

**JMS**

```java
return
```

**Throws**

- `JMSException`: JMS

---

**getPropertyNames**
**Enumeration** `getPropertyNames()`

```java
public Enumeration getPropertyNames() throws JMSException {
    // Implementation...
    return null; // Example return value
}
```

Returns an `Enumeration` of all the property names.

Note that JMS standard header fields are not considered properties and are not returned in this enumeration.

**Returns:**
- an `Enumeration` of all the names of property values

**Throws:**
- `JMSException` - if the JMS provider fails to get the property names due to some internal error.

---

**public void setBooleanProperty(String name, boolean value)**

```java
public void setBooleanProperty(String name, boolean value) throws JMSException {
    // Implementation...
}
```

Sets a boolean property value with the specified name into the message.

**Parameters:**
- `name` - the name of the boolean property
- `value` - the boolean property value to set

**Throws:**
- `JMSException` - if the JMS provider fails to set the property due to some internal error.
IllegalArgumentException - if the name is null or if the name is an empty string.
MessageNotWriteableException - if properties are read-only

public void setByteProperty(String name, byte value) throws JMSException
  byte
  name	byte
  value	byte
  Throws JMSException: JMS
  Throws IllegalArgumentException: null
  Throws MessageNotWriteableException:

setByteProperty

void setByteProperty(String name, byte value)
  throws JMSException

Sets a byte property value with the specified name into the message.

Parameters:
  name - the name of the byte property
  value - the byte property value to set

Throws:
  JMSException - if the JMS provider fails to set the property due to some internal error.
  IllegalArgumentException - if the name is null or if the name is an empty string.
  MessageNotWriteableException - if properties are read-only

public void setShortProperty(String name, short value) throws JMSException
**setShortProperty**

```java
void setShortProperty(String name, short value)
throws JMSException
```

Sets a short property value with the specified name into the message.

**Parameters:**
- `name` - the name of the short property
- `value` - the short property value to set

**Throws:**
- `JMSException` - if the JMS provider fails to set the property due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageNotWriteableException` - if properties are read-only

---

**public void setIntProperty(String name, int value)**

```java
int
```

```java
void setIntProperty(String name, int value)
throws JMSException
```

Throws `JMSException`: JMS

Throws `IllegalArgumentException`: null

Throws `MessageNotWriteableException`: null
**setIntProperty**

```java
void setIntProperty(String name, int value)
    throws JMSException
```

Sets an `int` property value with the specified name into the message.

**Parameters:**
- `name` - the name of the `int` property
- `value` - the `int` property value to set

**Throws:**
- `JMSException` - if the JMS provider fails to set the property due to some internal error.
- `IllegalArgumentException` - if the name is `null` or if the name is an empty string.
- `MessageNotWriteableException` - if properties are read-only

---

**public void setLongProperty(String name, long value) throws JMSException**

```java
long name
long value
```

**Throws**
- `JMSException`: JMS
- `IllegalArgumentException`: `null`
- `MessageNotWriteableException`: 

---

**setLongProperty**

```java
void setLongProperty(String name, long value)
    throws JMSException
```

Sets a `long` property value with the specified name into the message.
public void setFloatProperty(String name, float value) throws JMSException

    float
    name  float
    value  float

    Throws  JMSException: JMS
    Throws  IllegalArgumentException: null
    Throws  MessageNotWriteableException:
public void setDoubleProperty(String name, double value) throws JMSException

Sets a double property value with the specified name into the message.

Parameters:
name - the name of the double property
double value - the double property value to set

Throws:
JMSException - if the JMS provider fails to set the property due to some internal error.
IllegalArgumentException - if the name is null or if the name is an empty string.
MessageNotWriteableException - if properties are read-only

public void setStringProperty(String name, String value) throws JMSException

String value
String
setStringProperty

```java
void setStringProperty(String name,
                       String value)
throws JMSException
```

Sets a string property value with the specified name into the message.

**Parameters:**
- `name` - the name of the string property
- `value` - the string property value to set

**Throws:**
- `JMSException` - if the JMS provider fails to set the property due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageNotWriteableException` - if properties are read-only
setObjectProperty

```java
void setObjectProperty(String name, Object value)
throws JMSException
```

Sets a Java object property value with the specified name into the message.

Note that this method works only for the objectified primitive object types (Integer, Double, Long ...) and String objects.

**Parameters:**
- `name` - the name of the Java object property
- `value` - the Java object property value to set

**Throws:**
- `JMSException` - if the JMS provider fails to set the property due to some internal error.
- `IllegalArgumentException` - if the name is null or if the name is an empty string.
- `MessageFormatException` - if the object is invalid
- `MessageNotWriteableException` - if properties are read-only

---

public void acknowledge() throws JMSException

```java
JMS acknowledge
```

**Throws** `JMSException`: JMS
acknowledge

void acknowledge() throws JMSException

Acknowledges all consumed messages of the session of this consumed message.

All consumed JMS messages support the acknowledge method for use when a client has specified that its JMS session's consumed messages are to be explicitly acknowledged. By invoking acknowledge on a consumed message, a client acknowledges all messages consumed by the session that the message was delivered to.

Calls to acknowledge are ignored for both transacted sessions and sessions specified to use implicit acknowledgement modes.

A client may individually acknowledge each message as it is consumed, or it may choose to acknowledge messages as an application-defined group (which is done by calling acknowledge on the last received message of the group, thereby acknowledging all messages consumed by the session.)

Messages that have been received but not acknowledged may be redelivered.

Throws:

  JMSException - if the JMS provider fails to acknowledge the messages due to some internal error.
  IllegalStateException - if this method is called on a closed session.

See Also:
public void clearBody() throws JMSException

Throws JMSException: JMS

clearBody

void clearBody()
throws JMSException

Clears out the message body. Clearing a message's body does not clear its header values or property entries.

If this message body was read-only, calling this method leaves the message body in the same state as an empty body in a newly created message.

Throws:

JMSException - if the JMS provider fails to clear the message body due to some internal error.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.mail  **Class Message**

**java.lang.Object**
  - javax.mail.Message

**All Implemented Interfaces:**
  - Part

**Direct Known Subclasses:**
  - MimeMessage

```java
public abstract class Message
    extends Object
    implements Part
```

**Implements:** Part

**Inner classes:** Message.RecipientType

**Extended by:** MimeMessage

This class models an email message. This is an abstract class. Subclasses provide actual implementations.

**See also**  
javax.mail.Part
Message implements the Part interface. Message contains a set of attributes and a "content". Messages within a folder also have a set of flags that describe its state within the folder.

Message defines some new attributes in addition to those defined in the Part interface. These attributes specify meta-data for the message - i.e., addressing and descriptive information about the message.

Message objects are obtained either from a Folder or by constructing a new Message object of the appropriate subclass. Messages that have been received are normally retrieved from a folder named "INBOX".

A Message object obtained from a folder is just a lightweight reference to the actual message. The Message is 'lazily' filled up (on demand) when each item is requested from the message. Note that certain folder implementations may return Message objects that are pre-filled with certain user-specified items. To send a message, an appropriate subclass of Message (e.g., MimeMessage) is instantiated, the attributes and content are filled in, and the message is sent using the Transport.send method.

**Author:**
John Mani, Bill Shannon, Max Spivak

**See Also:**
Part

---

### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>Message.RecipientType</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This inner class defines the types of recipients allowed by the Message class.</td>
</tr>
</tbody>
</table>

### Field Summary

<table>
<thead>
<tr>
<th>protected boolean</th>
<th>expunged</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>True if this message has been expunged.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected</th>
<th>folder</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The containing folder, if this message is obtained from a</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>Folder</code></td>
<td><code>folder</code></td>
</tr>
<tr>
<td><code>msgnum</code></td>
<td>The number of this message within its folder, or zero if the message was not retrieved from a folder.</td>
</tr>
<tr>
<td><code>Session</code></td>
<td><code>session</code></td>
</tr>
<tr>
<td><code>session</code></td>
<td>The Session object for this Message</td>
</tr>
</tbody>
</table>

**Fields inherited from interface `javax.mail.Part`**

- `ATTACHMENT`  
- `INLINE`

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Message()</code></td>
<td>No-arg version of the constructor.</td>
</tr>
<tr>
<td><code>Message(Folder folder, int msgnum)</code></td>
<td>Constructor that takes a Folder and a message number.</td>
</tr>
<tr>
<td><code>Message(Session session)</code></td>
<td>Constructor that takes a Session.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addFrom(Address[] addresses)</code></td>
<td>Add these addresses to the existing &quot;From&quot; attribute</td>
</tr>
<tr>
<td><code>addRecipient(Message.RecipientType type, Address address)</code></td>
<td>Add this recipient address to the existing ones of the given type.</td>
</tr>
<tr>
<td><code>addRecipients(Message.RecipientType type, Address[] addresses)</code></td>
<td>Add these recipient addresses to the existing ones of the given type.</td>
</tr>
<tr>
<td><code>getAllRecipients()</code></td>
<td>Get all the recipient addresses for the message.</td>
</tr>
<tr>
<td><code>getFlags()</code></td>
<td>Returns a Flags object containing the flags for this message.</td>
</tr>
</tbody>
</table>
Folder

getFolder()
Get the folder from which this message was obtained.

abstract Address[]

getFrom()
Returns the "From" attribute.

int getMessageNumber()
Get the Message number for this Message.

abstract Date

getReceivedDate()
Get the date this message was received.

abstract Address[]

getRecipients(Message.RecipientType type)
Get all the recipient addresses of the given type.

Address[]

getReplyTo()
Get the addresses to which replies should be directed.

abstract Date

getSentDate()
Get the date this message was sent.

abstract String

getSubject()
Get the subject of this message.

boolean isExpunged()
Checks whether this message is expunged.

boolean isSet(Flags.Flag flag)
Check whether the flag specified in the flag argument is set in this message.

boolean match(SearchTerm term)
Apply the specified Search criterion to this message.

abstract Message

reply(boolean replyToAll)
Get a new Message suitable for a reply to this message.

abstract void

saveChanges()
Save any changes made to this message into the message-store when the containing folder is closed, if the message is contained in a folder.

protected void

setExpunged(boolean expunged)
Sets the expunged flag for this Message.

void setFlag(Flags.Flag flag, boolean set)
Set the specified flag on this message to the specified value.

void setFlags(Flags flag, boolean set)
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract void setFrom()</code></td>
<td>Set the &quot;From&quot; attribute in this Message.</td>
</tr>
<tr>
<td><code>abstract void setFrom(Address address)</code></td>
<td>Set the &quot;From&quot; attribute in this Message.</td>
</tr>
<tr>
<td><code>protected void setMessageNumber(int msgnum)</code></td>
<td>Set the Message number for this Message.</td>
</tr>
<tr>
<td><code>void setRecipient(Message.RecipientType type, Address address)</code></td>
<td>Set the recipient address.</td>
</tr>
<tr>
<td><code>abstract void setRecipients(Message.RecipientType type, Address[] addresses)</code></td>
<td>Set the recipient addresses.</td>
</tr>
<tr>
<td><code>void setReplyTo(Address[] addresses)</code></td>
<td>Set the addresses to which replies should be directed.</td>
</tr>
<tr>
<td><code>abstract void setSentDate(Date date)</code></td>
<td>Set the sent date of this message.</td>
</tr>
<tr>
<td><code>abstract void setSubject(String subject)</code></td>
<td>Set the subject of this message.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface javax.mail.**Part**

- `addHeader`, `getAllHeaders`, `getContent`, `contentType`, `getDataHandler`, `getDescription`, `getDisposition`, `getFileName`, `getHeader`, `getInputStream`, `getLineCount`, `getMatchingHeaders`, `getNonMatchingHeaders`, `getSize`, `isMime`, `removeHeader`, `setContent`, `setContent`, `setDataHandler`, `setDescription`, `setDisposition`, `setFileName`, `setHeader`, `setText`, `writeTo`

Field Detail
msgnum

protected int msgnum

   The number of this message within its folder, or zero if the message was not retrieved from a folder.

-------------------------------------------------------

expunged

protected boolean expunged

   True if this message has been expunged.

-------------------------------------------------------

folder

protected Folder folder

   The containing folder, if this message is obtained from a folder

-------------------------------------------------------

session

protected Session session

   The Session object for this Message

-------------------------------------------------------

**Constructor Detail**

protected Message()
**Message**

```java
protected Message()
```

No-arg version of the constructor.

---

**Message**

```java
protected Message(Folder folder, int msgnum)
```

Folder  Folder

    folder
    msgnum

Constructor that takes a Folder and a message number. Used by Folder implementations.

**Parameters:**

- `folder` - containing folder
- `msgnum` - this message's sequence number within this folder

---

**Message**

```java
protected Message(Session session)
```

Session  Message

    session
    Session

---

**Message**

```java
protected Message(Session session)
```
Constructor that takes a Session. Used for client created Message objects.

**Parameters:**

session - A Session object

**Method Detail**

```java
abstract public Address[] getFrom() throws MessagingException

"From" "From"
```

Returns the "From" attribute. The "From" attribute contains the identity of the person(s) who wished this message to be sent.

In certain implementations, this may be different from the entity that actually sent the message.

This method returns `null` if this attribute is not present in this message. Returns an empty array if this attribute is present, but contains no addresses.

**Returns:**

array of Address objects
abstract public void setFrom() throws MessagingException
Message "From" "mail.user"
"user.name"

Throws

Throws MessagingException: 
Throws IllegalWriteException: 
Throws IllegalStateException: READ_ONLY 

setFrom

public abstract void setFrom()

throws MessagingException

Set the "From" attribute in this Message. The value of this attribute is obtained from the property "mail.user". If this property is absent, the system property "user.name" is used.

Throws:

MessagingException
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ ONLY folder.

abstract public void setFrom(Address address) throws MessagingException
Message "From"

address

Throws MessagingException:

Throws IllegalWriteException:

Throws IllegalStateException: READ_ONLY
setFrom

public abstract void setFrom(Address address) throws MessagingException

Set the "From" attribute in this Message.

Parameters:
address - the sender

Throws:
MessagingException
IllegalArgumentException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.

abstract public void addFrom(Address[] addresses) throws MessagingException
"From"

addresses

Throws

IllegalWriteException:

IllegalStateException: READ_ONLY

MessagingException:

addFrom

public abstract void addFrom(Address[] addresses) throws MessagingException

Add these addresses to the existing "From" attribute

Parameters:
addresses - the senders

Throws:

IllegalArgumentException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a
abstract public Address[] getRecipients(Message.RecipientType type) throws MessagingException

null

type
return Address

Throws MessagingException:

See also

getRecipients

public abstract Address[] getRecipients(Message.RecipientType type) throws MessagingException

Get all the recipient addresses of the given type.

This method returns null if no recipients of the given type are present in this message. It may return an empty array if the header is present, but contains no addresses.

Parameters:

type - the recipient type

Returns:

array of Address objects

Throws: MessagingException

See Also:

Message.RecipientType.TO, Message.RecipientType.CC, Message.RecipientType.BCC
public Address[] getAllRecipients() throws MessagingException

getRecipients TOCC BCC

public Address[] getAllRecipients()

throws MessagingException

null

return Address

Throws MessagingException:

See also TO, CC, BCC, getRecipients

getAllRecipients

Get all the recipient addresses for the message. The default implementation extracts the TO, CC, and BCC recipients using the getRecipients method.

This method returns null if none of the recipient headers are present in this message. It may return an empty array if any recipient header is present, but contains no addresses.

Returns:
array of Address objects

Throws:
MessagingException

See Also:
Message.RecipientType.TO, Message.RecipientType.CC,
Message.RecipientType.BCC,
getRecipients(javax.mail.Message.RecipientType)

abstract public void setRecipients(Message.RecipientType type, Address[] addresses) throws MessagingException

addresses

type
addresses

Throws MessagingException:
 Throws IllegalArgumentException:
 Throws IllegalStateException: READ_ONLY

setRecipients

public abstract void setRecipients(Message.RecipientType type, Address[] addresses) throws MessagingException

Set the recipient addresses. All addresses of the specified type are replaced by the addresses parameter.

Parameters:
 type - the recipient type
 addresses - the addresses

Throws:
 MessagingException
 IllegalArgumentException - if the underlying implementation does not support modification of existing values
 IllegalStateException - if this message is obtained from a READ_ONLY folder.

public void setRecipient(Message.RecipientType type, Address address) throws MessagingException

setRecipients
 type
 address

Throws MessagingException:
 Throws IllegalArgumentException:
 Throws IllegalStateException:
**setRecipient**

```java
public void setRecipient(Message.RecipientType type, Address address)
        throws MessagingException
```

Set the recipient address. All addresses of the specified type are replaced by the address parameter.

The default implementation uses the `setRecipients` method.

**Parameters:**
- `type` - the recipient type
- `address` - the address

**Throws:**
- `MessagingException`
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values

---

**abstract public void addRecipients**

```java
abstract public void addRecipients(Message.RecipientType type, Address[] addresses) throws MessagingException
```

**Parameters:**
- `type` - the recipient type
- `addresses` - the recipient addresses

**Throws:**
- `MessagingException`
- `IllegalWriteException`
- `IllegalStateException`: `READ_ONLY`

---

**addRecipients**

```java
public abstract void addRecipients(Message.RecipientType type, Address[] addresses)
        throws MessagingException
```

Add these recipient addresses to the existing ones of the given type.

**Parameters:**
public void addRecipient(Message.RecipientType type, Address address) throws MessagingException

Add this recipient address to the existing ones of the given type.

The default implementation uses the addRecipients method.

Parameters:

type - the recipient type
address - the address

Throws:

MessageException
IllegalWriteException - if the underlying implementation does not support modification of existing values

IllegalStateException - if this message is obtained from a READ_ONLY folder.
public Address[] getReplyTo() throws MessagingException

getFrom

null

return

Throws MessagingException: getFrom

See also getFrom

getReplyTo

public Address[] getReplyTo()

throws MessagingException

Get the addresses to which replies should be directed. This will usually be the sender of the message, but some messages may direct replies to a different address.

The default implementation simply calls the getFrom method.

This method returns null if the corresponding header is not present. Returns an empty array if the header is present, but contains no addresses.

Returns:

addresses to which replies should be directed

Throws:

MessagingException

See Also:

getFrom()

public void setReplyTo(Address[] addresses) throws MessagingException

MessagingException
MethodNotSupportedException

addresses

Throws MessagingException:

Throws IllegalWriteException:

Throws IllegalStateException: READ_ONLY

Throws MethodNotSupportedException:

**setReplyTo**

public void setReplyTo(Address[] addresses) throws MessagingException

Set the addresses to which replies should be directed. (Normally only a single address will be specified.) Not all message types allow this to be specified separately from the sender of the message.

The default implementation provided here just throws the MethodNotSupportedException.

**Parameters:**
- addresses - addresses to which replies should be directed

**Throws:**
- MessagingException
- IllegalWriteException - if the underlying implementation does not support modification of existing values
- IllegalStateException - if this message is obtained from a READ_ONLY folder.
- MethodNotSupportedException - if the underlying implementation does not support setting this attribute

abstract public String getSubject() throws MessagingException
getSubject

public abstract String getSubject() throws MessagingException

Get the subject of this message.

Returns:
the subject

Throws:
MessagingException

abstract public void setSubject(String subject) throws MessagingException

subject

Throws MessagingException:
Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY

setSubject

public abstract void setSubject(String subject) throws MessagingException

Set the subject of this message.

Parameters:
subject - the subject

Throws:
MessagingException
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.

abstract public java.util.Date getSentDate() throws MessagingException

    return

    Throws MessagingException:

getSentDate

public abstract Date getSentDate() throws MessagingException

    Get the date this message was sent.

    Returns: the date this message was sent

    Throws: MessagingException

abstract public void setSentDate(java.util.Date date) throws MessagingException

    date

    Throws MessagingException:
    Throws IllegalArgumentException:
    Throws IllegalStateException: READ_ONLY

setSentDate

public abstract void setSentDate(Date date) throws MessagingException
Set the sent date of this message.

**Parameters:**
- `date` - the sent date of this message

**Throws:**
- `MessagingException`
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this message is obtained from a `READ_ONLY` folder.

```java
abstract public java.util.Date getReceivedDate() throws MessagingException

return

getReceivedDate

public abstract Date getReceivedDate() throws MessagingException

Get the date this message was received.

**Returns:**
- the date this message was received

**Throws:**
- `MessagingException`

```java
abstract public Flags getFlags() throws MessagingException

Flags

Flags

setF1
```
getFlags

```java
public abstract Flags getFlags() throws MessagingException {
    return new Flags();
}
```

Returns a Flags object containing the flags for this message.

Modifying any of the flags in this returned Flags object will not affect the flags of this message. Use `setFlags()` to do that.

Returns:
- Flags object containing the flags for this message

Throws:
- MessagingException

See Also:
- Flags, `setFlags(javax.mail.Flags, boolean)`

---

public boolean isSet(Flags.Flag flag) throws MessagingException

```java
getFlags

flag

return

Throws
- MessagingException:

See also
- javax.mail.Flags.Flag, ANSWERED, DELETED, DRAFT, FLAGGED, RECENT, SEEN

isSet
```
public boolean isset(Flags.Flag flag)
    throws MessagingException

Check whether the flag specified in the flag argument is set in this
message.

The default implementation uses getFlags.

Parameters:
    flag - the flag

Returns:
    value of the specified flag for this message

Throws:
    MessagingException

See Also:
    Flags.Flag, Flags.Flag.ANSWERED, Flags.Flag.DELETED,
    Flags.Flag.DRAFT, Flags.Flag.FLAGGED, Flags.Flag.RECENT,
    Flags.Flag.SEEN

abstract public void setFlags(Flags flag, boolean set)
    throws MessagingException

Flags

Message MessageChangedListener
MessageChangedEvent

flag                Flag
set

Throws
    MessagingException:
    IllegalWriteException:
    IllegalStateException: READ_ONLY

See also
    javax.mail.event.MessageChangedEvent

setFlags

public abstract void setFlags(Flags flag,
    boolean set)
throws MessagingException

Set the specified flags on this message to the specified value. Note that any flags in this message that are not specified in the given Flags object are unaffected.

This will result in a MessageChangedEvent being delivered to any MessageChangedListener registered on this Message's containing folder.

Parameters:
flag - Flags object containing the flags to be set
set - the value to be set

Throws:
MessagingException
IllegalWriteException - if the underlying implementation does not support modification of existing values.
IllegalStateException - if this message is obtained from a READ_ONLY folder.

See Also:
MessageChangedEvent
setFlag

public void setFlag(Flags.Flag flag, boolean set)
        throws MessagingException

Set the specified flag on this message to the specified value. This will result in a MessageChangedEvent being delivered to any MessageChangedListener registered on this Message's containing folder.

The default implementation uses the setFlags method.

Parameters:
   flag - Flags.Flag object containing the flag to be set
   set - the value to be set

Throws:
   MessagingException
   IllegalWriteException - if the underlying implementation does not support modification of existing values.
   IllegalStateException - if this message is obtained from a READ_ONLY folder.

See Also:
   MessageChangedEvent

public int getMessageNumber()
        Message Message Message Folder
Message Folder Message

    1 0

    return

gMessageNumber

public int getMessageNumber()
Get the Message number for this Message. A Message object's message number is the relative position of this Message in its Folder. Note that the message number for a particular Message can change during a session if other messages in the Folder are deleted and expunged.

Valid message numbers start at 1. Messages that do not belong to any folder (like newly composed or derived messages) have 0 as their message number.

**Returns:**
the message number

---

```java
protected void setMessageNumber(int msgnum)
Message Message

setMessageNumber
protected void setMessageNumber(int msgnum)

Set the Message number for this Message. This method is invoked only by the implementation classes.

---

```java
public Folder getFolder()
null

getFolder
public Folder getFolder()

Get the folder from which this message was obtained. If this is a new message or nested message, this method returns null.
public boolean isExpunged()

getMessageNumber()  Message

Folder  expunge()  Message  Folder
Message  "expunged"  isExpunged()  true
Folder  expunge()  Folder

expunge()

See also  expunge

isExpunged

public boolean isExpunged()

Checks whether this message is expunged. All other methods except getMessageNumber() are invalid on an expunged Message object.

Messages that are expunged due to an explicit expunge() request on the containing Folder are removed from the Folder immediately. Messages that are externally expunged by another source are marked "expunged" and return true for the isExpunged() method, but they are not removed from the Folder until an explicit expunge() is done on the Folder.

See the description of expunge() for more details on expunge handling.

See Also:
  Folder.expunge()
protected void setExpunged(boolean expunged)
Message expunged
expunged
expunged

setExpunged

protected void setExpunged(boolean expunged)

Sets the expunged flag for this Message. This method is to be used only by the implementation classes.

Parameters:
expunged - the expunged flag

abstract public Message reply(boolean replyToAll) throws MessagingException
Message Message

replyToAll Message
getReplyTo

"Subject" "Re:" "Re:"

replyToAll
return Message
Throws MessagingException:

reply

public abstract Message reply(boolean replyToAll)
throws MessagingException

Get a new Message suitable for a reply to this message. The new
Message will have its attributes and headers set up appropriately. Note that this new message object will be empty, that is, it will **not** have a "content". These will have to be suitably filled in by the client.

If `replyToAll` is set, the new Message will be addressed to all recipients of this message. Otherwise, the reply will be addressed to only the sender of this message (using the value of the `getReplyTo` method).

The "Subject" field is filled in with the original subject prefixed with "Re:" (unless it already starts with "Re:").

The reply message will use the same session as this message.

**Parameters:**
- `replyToAll` - reply should be sent to all recipients of this message  

**Returns:**
- the reply Message

**Throws:**
- `MessagingException`

```java
abstract public void saveChanges() throws MessagingException
```

**saveChanges**

**saveChanges** **Folder**

**READ ONLY** **saveChanges**

**Throws**
- `MessagingException`:
- `IllegalStateException`: **READ ONLY**
- `IllegalWriteException`:

```java
saveChanges
```

**saveChanges**

```java
public abstract void saveChanges()
```
Save any changes made to this message into the message-store when the containing folder is closed, if the message is contained in a folder. (Some implementations may save the changes immediately.) Update any header fields to be consistent with the changed message contents. If any part of a message's headers or contents are changed, saveChanges must be called to ensure that those changes are permanent. If saveChanges is not called, any such modifications may or may not be saved, depending on the message store and folder implementation.

Messages obtained from folders opened READ_ONLY should not be modified and saveChanges should not be called on such messages.

Throws:
- MessagingException
- IllegalStateException - if this message is obtained from a READ_ONLY folder.
- IllegalWriteException - if the underlying implementation does not support modification of existing values.

public boolean match(SearchTerm term) throws MessagingException

Apply the specified Search criterion to this message.
Parameters:
  term - the Search criterion

Returns:
  true if the Message matches this search criterion, false otherwise.

Throws:
  MessagingException

See Also:
  searchTerm
javax.mail Class Message.RecipientType

java.lang.Object
   javax.mail.Message.RecipientType

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   MimeMessage.RecipientType

Enclosing class:
   Message

This inner class defines the types of recipients allowed by the Message class. The currently defined types are TO, CC and BCC. Note that this class only has a protected constructor, thereby restricting new Recipient types to either this class or subclasses. This effectively implements an
enumeration of the allowed Recipient types. The following code sample shows how to use this class to obtain the "TO" recipients from a message.

```java
Message msg = folder.getMessages(1);
Address[] a = m.getRecipients(Message.RecipientType.TO);
```

See Also:
- `Message.getRecipients(javax.mail.Message.RecipientType)`, `Message.setRecipients(javax.mail.Message.RecipientType, javax.mail.Address[])`, `Message.addRecipients(javax.mail.Message.RecipientType, javax.mail.Address[])`, Serialized Form

## Field Summary

<table>
<thead>
<tr>
<th>Static Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Message.RecipientType.BCC</code></td>
<td>The &quot;Bcc&quot; (blind carbon copy) recipients.</td>
</tr>
<tr>
<td><code>Message.RecipientType.CC</code></td>
<td>The &quot;Cc&quot; (carbon copy) recipients.</td>
</tr>
<tr>
<td><code>Message.RecipientType.TO</code></td>
<td>The &quot;To&quot; (primary) recipients.</td>
</tr>
</tbody>
</table>

**protected string type**

The type of recipient, usually the name of a corresponding Internet standard header.

## Constructor Summary

**protected Message.RecipientType(String type)**

Constructor for use by subclasses.

## Method Summary

**protected Object readResolve()**

When deserializing a RecipientType, we need to make sure to return only one of the known static final instances
defined in this class.

Methods inherited from class java.lang. Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

TO

public static final Message.RecipientType TO

The "To" (primary) recipients.

CC

public static final Message.RecipientType CC

The "Cc" (carbon copy) recipients.

BCC

public static final Message.RecipientType BCC

The "Bcc" (blind carbon copy) recipients.
type

protected String type

The type of recipient, usually the name of a corresponding Internet standard header.

### Constructor Detail

protected Message.RecipientType(String type)

### Message.RecipientType

protected Message.RecipientType(String type)

Constructor for use by subclasses.

### Method Detail

protected Object readResolve() throws java.io.ObjectStreamException

RecipientType

**readResolve**

protected Object readResolve()

throws ObjectStreamException

When deserializing a RecipientType, we need to make sure to return only one of the known static final instances defined in this class. Subclasses must implement their own readResolve method that
checks for their known instances before calling this super method.

**Throws:**

ObjectStreamException

---

**public String toString()**

**toString**

public String toString()

**Overrides:**

toString in class Object

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail Interface MessageAware

All Known Implementing Classes:  
   MimePartDataSource

```
public interface MessageAware
```

Implemented by: MimePartDataSource

DataSources DataContentHandler

since JavaMail 1.1

See javax.mail.MessageContext, javax.activation.DataSource,
also javax.activation.DataContentHandler

An interface optionally implemented by DataSources to supply information to a DataContentHandler about the message context in which the data content object is operating.

Since:  
JavaMail 1.1

See Also:  
MessageContext, DataSource, DataContentHandler

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getMessageContext()</code></td>
<td>Return the message context.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public MessageContext getMessageContext()
```
getMessageContext

MessageContext  getMessageContext()

    Return the message context.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.event Class MessageChangedEvent

java.lang.Object
  └ java.util.EventObject
      └ javax.mail.event.MailEvent
          └ javax.mail.event.MessageChangedEvent

All Implemented Interfaces:
  Serializable

public class MessageChangedEvent
  extends MailEvent

Extends: java.util.EventObject > MailEvent

Message

This class models Message change events.

Author:
  John Mani

See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>static int ENVELOPE_CHANGED</th>
<th>The message's envelope (headers, but not body) changed.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int FLAGS_CHANGED</td>
<td>The message's flags changed.</td>
</tr>
<tr>
<td>protected Message msg</td>
<td>The message that changed.</td>
</tr>
<tr>
<td>protected type</td>
<td></td>
</tr>
</tbody>
</table>
### Constructor Summary

**MessageChangedEvent**(Object source, int type, Message msg)
- Constructor.

### Method Summary

- **dispatch**(Object listener)
  - Invokes the appropriate MessageChangedListener method.

- **getMessage**()
  - Return the changed Message.

- **getMessageChangeType**()
  - Return the type of this event.

### Methods inherited from class java.util.EventObject

- getSource, toString

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

### Field Detail

**FLAGS_CHANGED**

- public static final int **FLAGS_CHANGED**
The message's flags changed.

See Also:
   Constant Field Values

---

ENVELOPE_CHANGED

public static final int ENVELOPE_CHANGED

   The message's envelope (headers, but not body) changed.

See Also:
   Constant Field Values

---

type

protected int type

   The event type.

---

msg

protected transient Message msg

   The message that changed.

---

Constructor Detail

public MessageChangedEvent(Object source, int type,
Message msg)

source
type
msg

MessageChangedEvent

public MessageChangedEvent(Object source,
                          int type,
                          Message msg)

Constructor.

Parameters:
  source - The folder that owns the message
  type - The change type
  msg - The changed message

Method Detail

public int getMessageChangeType()

  return

getMessageChangeType

public int getMessageChangeType()

  Return the type of this event.

Returns:
  type
public Message getMessage()

    return

getMessage

public Message getMessage()

    Return the changed Message.

    Returns: the message

dispatch

public void dispatch(Object listener)
MessageChangedListener

dispatch

public void dispatch(Object listener)

    Invokes the appropriate MessageChangedListener method.

    Specified by:
    dispatch in class MailEvent
PS:
javax.mail.event  Interface MessageChangedListener

All Superinterfaces:
   EventListener

public interface MessageChangedListener
   extends EventListener

Implements: java.util.EventListener

   MessageChanged Listener

This is the Listener interface for MessageChanged events

Author:  
   John Mani

Method Summary

<table>
<thead>
<tr>
<th>void messageChanged(MessageChangedEvent e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoked when a message is changed.</td>
</tr>
</tbody>
</table>

Method Detail

public void messageChanged(MessageChangedEvent e)

   See also  FLAGS_CHANGED, ENVELOPE_CHANGED

messageChanged
void messageChanged(MessageChangedEvent e)

Invoked when a message is changed. The change-type specifies what changed.

See Also:
MessageChangedEvent.FLAGS_CHANGED, MessageChangedEvent.ENVELOPE_CHANGED
javax.jms Interface MessageConsumer

All Known Subinterfaces:
   QueueReceiver, TopicSubscriber

public interface MessageConsumer

Implemented by: QueueReceiver, TopicSubscriber

MessageConsumer Destination - MessageConsumer

MessageConsumer

receive receive

MessageListener MessageListener onMessage

MessageListener

version 1.0 - 13 March 1998 en

See javax.jms.QueueReceiver, javax.jms.TopicSubscriber, javax.jms.Session

A client uses a MessageConsumer object to receive messages from a destination. A MessageConsumer object is created by passing a Destination object to a message-consumer creation method supplied by a session.

MessageConsumer is the parent interface for all message consumers.

A message consumer can be created with a message selector. A message selector allows the client to restrict the messages delivered to the message consumer to those that match the selector.
A client may either synchronously receive a message consumer's messages or have the consumer asynchronously deliver them as they arrive.

For synchronous receipt, a client can request the next message from a message consumer using one of its `receive` methods. There are several variations of `receive` that allow a client to poll or wait for the next message.

For asynchronous delivery, a client can register a `MessageListener` object with a message consumer. As messages arrive at the message consumer, it delivers them by calling the `MessageListener`'s `onMessage` method.

It is a client programming error for a `MessageListener` to throw an exception.

**Version:**
1.0 - 13 March 1998

**Author:**
Mark Hapner, Rich Burridge

**See Also:**
QueueReceiver, TopicSubscriber, Session

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void close()</code></td>
<td>Closes the message consumer.</td>
</tr>
<tr>
<td><code>MessageListener getMessageListener()</code></td>
<td>Gets the message consumer's <code>MessageListener</code>.</td>
</tr>
<tr>
<td><code>String getMessageSelector()</code></td>
<td>Gets this message consumer's message selector expression.</td>
</tr>
<tr>
<td><code>Message receive()</code></td>
<td>Receives the next message produced for this message consumer.</td>
</tr>
<tr>
<td><code>receive(long timeout)</code></td>
<td></td>
</tr>
</tbody>
</table>
**Message**

Receives the next message that arrives within the specified timeout interval.

**receiveNoWait()**

Receives the next message if one is immediately available.

**setMessageListener(MessageListener listener)**

Sets the message consumer's MessageListener.

---

**Method Detail**

**public String getMessageSelector() throws JMSException**

```
return null
```

Throws: **JMSException**: JMS

---

**getMessagingSelector**

```
String getMessageSelector() throws JMSException
```

Gets this message consumer's message selector expression.

**Returns:**

- this message consumer's message selector, or null if no message selector exists for the message consumer (that is, if the message selector was not set or was set to null or the empty string)

**Throws:**

- **JMSException** - if the JMS provider fails to get the message selector due to some internal error.

---

**public MessageListener getMessageListener() throws JMSException**
getMessageListener

MessageListener getMessageListener()

throws JMSException

 Gets the message consumer's MessageListener.

 Returns:
the listener for the message consumer, or null if no listener is set

 Throws:
JMSException - if the JMS provider fails to get the message listener due to some internal error.

 See Also:
setMessageListener(javax.jms.MessageListener)

public void setMessageListener(MessageListener listener)

throws JMSException

 Sets the message consumer's MessageListener.

 Returns:
null

 Throws:
JMSException: JMS

 See also:
getMessageListener
void setMessageListener(MessageListener listener) throws JMSEception

Sets the message consumer's MessageListener.

Setting the message listener to null is the equivalent of unsetting the message listener for the message consumer.

The effect of calling MessageConsumer.setMessageListener While messages are being consumed by an existing listener or the consumer is being used to consume messages synchronously is undefined.

Parameters:
  listener - the listener to which the messages are to be delivered

Throws:
  JMSEception - if the JMS provider fails to set the message listener due to some internal error.

See Also:
  getMessageListener()

public Message receive() throws JMSEception

receive
  return null

Throws JMSEception: JMS

receive

Message receive() throws JMSEception
Receives the next message produced for this message consumer.

This call blocks indefinitely until a message is produced or until this message consumer is closed.

If this `receive` is done within a transaction, the consumer retains the message until the transaction commits.

**Returns:**
the next message produced for this message consumer, or null if this message consumer is concurrently closed

**Throws:**
`JMSException` - if the JMS provider fails to receive the next message due to some internal error.

```java
public Message receive(long timeout) throws JMSException
```

```
    timeout 0
    return null
```

`timeout` of zero never expires, and the call blocks indefinitely.

Receives the next message that arrives within the specified timeout interval.

This call blocks until a message arrives, the timeout expires, or this message consumer is closed. A `timeout` of zero never expires, and the call blocks indefinitely.
Parameters:
- timeout - the timeout value (in milliseconds)

Returns:
- the next message produced for this message consumer, or null if the timeout expires or this message consumer is concurrently closed

Throws:
- JMSException - if the JMS provider fails to receive the next message due to some internal error.

public Message receiveNoWait() throws JMSException

return null

Throws JMSException: JMS

receiveNoWait

Message receiveNoWait()
throws JMSException

Receives the next message if one is immediately available.

Returns:
- the next message produced for this message consumer, or null if one is not available

Throws:
- JMSException - if the JMS provider fails to receive the next message due to some internal error.

public void close() throws JMSException

Java MessageConsumer

receive receive null
Throws  JMSException: JMS

close

void close() throws JMSException

Closes the message consumer.

Since a provider may allocate some resources on behalf of a MessageConsumer outside the Java virtual machine, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

This call blocks until a receive or message listener in progress has completed. A blocked message consumer receive call returns null when this message consumer is closed.

Throws:
  JMSException - if the JMS provider fails to close the consumer due to some internal error.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.mail Class MessageContext

javax.lang.Object
   ^javax.mail.MessageContext

public class MessageContext
extends Object

Message MessageAware getMessageContext Me
MessageAware DataSources DataContentHandler

since JavaMail 1.1

See javax.mail.MessageAware, javax.activation.DataSource,
also javax.activation.DataContentHandler

The context in which a piece of Message content is contained. A
MessageContext object is returned by the getMessageContext method of the
MessageAware interface. MessageAware is typically implemented by
DataSources to allow a DataContentHandler to pass on information about
the context in which a data content object is operating.

Since: JavaMail 1.1
See Also: MessageAware, DataSource, DataContentHandler

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>MessageContext(Part part)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a MessageContext object describing the context of the given Part.</td>
</tr>
</tbody>
</table>

---

**Method Summary**
<table>
<thead>
<tr>
<th>Message</th>
<th>getMessage()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the Message that contains the content.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Part</th>
<th>getPart()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the Part that contains the content.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Session</th>
<th>getSession()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Return the Session we're operating in.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

public MessageContext(Part part)
Part MessageContext

**MessageContext**

public MessageContext(Part part)
Create a MessageContext object describing the context of the given Part.

**Method Detail**

public Part getPart()
Part

    return Part null

getPart

public Part getPart()
Return the Part that contains the content.

**Returns:**
the containing Part, or null if not known

```java
public Message getMessage()
Message Message Message Multipart Message null
    return Message null
```

**getMessage**

```java
public Message getMessage()
    Return the Message that contains the content. Follows the parent chain up through containing Multipart objects until it comes to a Message object, or null.

    **Returns:**
    the containing Message, or null if not known
```

```java
public Session getSession()
Session getMethod
    return Session null
```

**getSession**

```java
public Session getSession()
    Return the Session we're operating in.

    **Returns:**
    the Session, or null if not known
```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.handler Interface MessageContext

All Known Subinterfaces:
   SOAPMessageContext

public interface MessageContext

Implemented by: SOAPMessageContext

    MessageContext handle

    MessageContext MessageContext

    version MessageContext

    1.1

See also javax.xml.rpc.handler.Handler

The interface MessageContext abstracts the message context that is processed by a handler in the handle method.

The MessageContext interface provides methods to manage a property set. MessageContext properties enable handlers in a handler chain to share processing related state.

Version:

   1.1

Author:

    Rahul Sharma, Roberto Chinnici

See Also:

    Handler

---

Method Summary

<table>
<thead>
<tr>
<th>boolean containsProperty(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns true if the MessageContext contains a property with the specified name.</td>
</tr>
</tbody>
</table>
**getProperty** (String name)

Gets the value of a specific property from the MessageContext

**getPropertyNames**()

Returns an Iterator view of the names of the properties in this MessageContext

**removeProperty** (String name)

Removes a property (name-value pair) from the MessageContext

**setProperty** (String name, Object value)

Sets the name and value of a property associated with the MessageContext.

### Method Detail

**public void setProperty(String name, Object value)**

<table>
<thead>
<tr>
<th>MessageContext</th>
<th>MessageContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>MessageContext</td>
</tr>
<tr>
<td>value</td>
<td></td>
</tr>
</tbody>
</table>

Throws

- IllegalArgumentException:
- UnsupportedOperationException:

**setProperty**

void **setProperty**(String name, Object value)

Sets the name and value of a property associated with the MessageContext. If the MessageContext contains a value of the same property, the old value is replaced.

**Parameters:**

- name - Name of the property associated with the MessageContext
- value - Value of the property

**Throws:**
**IllegalArgumentException** - If some aspect of the property is prevents it from being stored in the context

**UnsupportedOperationException** - If this method is not supported.

---

```java
public Object getProperty(String name)
```

**MessageContext**

```java
    name
    return
    Throws IllegalArgumentException:
```  

**getProperty**

```java
Object getProperty(String name)
```

Gets the value of a specific property from the `MessageContext`

**Parameters:**

```java
    name - Name of the property whose value is to be retrieved
```  

**Returns:**

```java
    Value of the property
```  

**Throws:**

```java
    IllegalArgumentException - if an illegal property name is specified
```  

---

```java
public void removeProperty(String name)
```

**MessageContext**

```java
    name
    Throws IllegalArgumentException:
```  

**removeProperty**

```java
void removeProperty(String name)
```

Removes a property (name-value pair) from the `MessageContext`
public boolean containsProperty(String name)
    MessageContext true
    name
    return MessageContext true false

containsProperty

boolean containsProperty(String name)
    Returns true if the MessageContext contains a property with the specified name.

Parameters:
    name - Name of the property whose presence is to be tested

Returns:
    Returns true if the MessageContext contains the property; otherwise false

public java.util.Iterator<E> getPropertyNames()
    MessageContext Iterator
    return Iterator

getPropertyNames

Iterator getPropertyNames()
    Returns an Iterator view of the names of the properties in this
MessageContext

Returns:
Iterator for the property names

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Interface MessageContext

All Superinterfaces:
   Map<String, Object>

All Known Subinterfaces:
   LogicalMessageContext, SOAPMessageContext

public interface MessageContext
   extends Map<String, Object>

Implemented by: SOAPMessageContext

   MessageContext handle

   MessageContext version 1.1

See also javax.xml.rpc.handler.Handler

The interface MessageContext abstracts the message context that is processed by a handler in the handle method.

The MessageContext interface provides methods to manage a property set. MessageContext properties enable handlers in a handler chain to share processing related state.

Since:
   JAX-WS 2.0

---

### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>MessageContext.Scope</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Property scope.</td>
</tr>
<tr>
<td>Field Summary</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>HTTP_REQUEST_HEADERS</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: HTTP request headers.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>HTTP_REQUEST_METHOD</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: HTTP request method.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>HTTP_RESPONSE_CODE</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: HTTP response status code.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>HTTP_RESPONSE_HEADERS</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: HTTP response headers.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>INBOUND_MESSAGE_ATTACHMENTS</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: Map of attachments to a message for the inbound message, key is the MIME Content-ID, value is a DataHandler.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>MESSAGE_OUTBOUND_PROPERTY</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: message direction, true for outbound messages, false for inbound.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>OUTBOUND_MESSAGE_ATTACHMENTS</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: Map of attachments to a message for the outbound message, key is the MIME Content-ID, value is a DataHandler.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>PATH_INFO</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: Request Path Info Type: String</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>QUERY_STRING</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: Query string for request.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SERVLET_CONTEXT</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: servlet context object.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SERVLET_REQUEST</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: servlet request object.</td>
<td></td>
</tr>
<tr>
<td><strong>static String</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SERVLET_RESPONSE</strong></td>
<td></td>
</tr>
<tr>
<td>Standard property: servlet response object.</td>
<td></td>
</tr>
</tbody>
</table>
static String WSDL_DESCRIPTION

static String WSDL_INTERFACE
   Standard property: name of wsdl interface (2.0) or port type (1.1).

static String WSDL_OPERATION
   Standard property: name of WSDL operation.

static String WSDL_PORT
   Standard property: name of WSDL port.

static String WSDL_SERVICE
   Standard property: name of WSDL service.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageContext.Scope.getScope(String name)</td>
<td>Gets the scope of a property.</td>
</tr>
<tr>
<td>void setScope(String name, MessageContext.Scope scope)</td>
<td>Sets the scope of a property.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface java.util.Map
- clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values

### Field Detail

MESSAGE_OUTBOUND_PROPERTY

static final String MESSAGE_OUTBOUND_PROPERTY

   Standard property: message direction, true for outbound messages, false for inbound.

   Type: boolean
See Also:
Constant Field Values

INBOUND_MESSAGE_ATTACHMENTS

static final String INBOUND_MESSAGE_ATTACHMENTS

Standard property: Map of attachments to a message for the inbound message, key is the MIME Content-ID, value is a DataHandler.

Type: java.util.Map

See Also:
Constant Field Values

OUTBOUND_MESSAGE_ATTACHMENTS

static final String OUTBOUND_MESSAGE_ATTACHMENTS

Standard property: Map of attachments to a message for the outbound message, key is the MIME Content-ID, value is a DataHandler.

Type: java.util.Map

See Also:
Constant Field Values

WSDL_DESCRIPTION
static final String WSDL_DESCRIPTION


Type: org.xml.sax.InputSource

See Also:
Constant Field Values

WSDL_SERVICE

static final String WSDL_SERVICE

Standard property: name of WSDL service.

Type: javax.xml.namespace.QName

See Also:
Constant Field Values

WSDL_PORT

static final String WSDL_PORT

Standard property: name of WSDL port.

Type: javax.xml.namespace.QName

See Also:
Constant Field Values
WSDL_INTERFACE

static final String WSDL_INTERFACE

Standard property: name of wsdl interface (2.0) or port type (1.1).

Type: javax.xml.namespace.QName

See Also:
   Constant Field Values

--------------------------------------------------------

WSDL_OPERATION

static final String WSDL_OPERATION

Standard property: name of WSDL operation.

Type: javax.xml.namespace.QName

See Also:
   Constant Field Values

--------------------------------------------------------

HTTP_RESPONSE_CODE

static final String HTTP_RESPONSE_CODE

Standard property: HTTP response status code.

Type: java.lang.Integer

See Also:
   Constant Field Values

--------------------------------------------------------
HTTP_REQUEST_HEADERS

static final String HTTP_REQUEST_HEADERS

Standard property: HTTP request headers.

Type: java.util.Map

See Also: Constant Field Values

HTTP_RESPONSE_HEADERS

static final String HTTP_RESPONSE_HEADERS

Standard property: HTTP response headers.

Type: java.util.Map

See Also: Constant Field Values

HTTP_REQUEST_METHOD

static final String HTTP_REQUEST_METHOD

Standard property: HTTP request method.

Type: java.lang.String

See Also: Constant Field Values
SERVLET_REQUEST

static final String SERVLET_REQUEST

    Standard property: servlet request object.

    Type: javax.servlet.http.HttpServletRequest

See Also:
    Constant Field Values

SERVLET_RESPONSE

static final String SERVLET_RESPONSE

    Standard property: servlet response object.

    Type: javax.servlet.http.HttpServletResponse

See Also:
    Constant Field Values

SERVLET_CONTEXT

static final String SERVLET_CONTEXT

    Standard property: servlet context object.

    Type: javax.servlet.ServletContext

See Also:
Constant Field Values

QUERY_STRING

static final String QUERY_STRING

Standard property: Query string for request.
Type: String

See Also:
Constant Field Values

PATH_INFO

static final String PATH_INFO

Standard property: Request Path Info
Type: String

See Also:
Constant Field Values

Method Detail

setScope

void setScope(String name, MessageContext.Scope scope)
Sets the scope of a property.

**Parameters:**
- `name` - Name of the property associated with the `MessageContext` scope
- `scope` - Desired scope of the property

**Throws:**
- `IllegalArgumentException` - if an illegal property name is specified

---

getScope

`MessageContext.Scope getScope(String name)`

Gets the scope of a property.

**Parameters:**
- `name` - Name of the property

**Returns:**
- Scope of the property

**Throws:**
- `IllegalArgumentException` - if a non-existant property name is specified

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.ws.handler  Enum MessageContext.Scope

java.lang.Object
  └ java.lang.Enum<MessageContext.Scope>  
    └ javax.xml.ws.handler.MessageContext.Scope

All Implemented Interfaces:
  Serializable, Comparable<MessageContext.Scope>

Enclosing interface:
  MessageContext

public static enum MessageContext.Scope extends Enum<MessageContext.Scope>

Extends: Enum<E>
Contained within: MessageContext

APPLICATION      HANDLER

Property scope. Properties scoped as APPLICATION are visible to handlers, client applications and service endpoints; properties scoped as HANDLER are only normally visible to handlers.

---

Enum Constant Summary

<table>
<thead>
<tr>
<th>APPLICATION</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HANDLER</td>
<td></td>
</tr>
</tbody>
</table>

---

Method Summary

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>valueOf</td>
<td>(String name)</td>
</tr>
<tr>
<td>static MessageContext.Scope</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>static MessageContext.Scope[]</td>
<td>values() Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Enum**
- clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

**Methods inherited from class java.lang.Object**
- finalize, getClass, notify, notifyAll, wait, wait, wait

**Enum Constant Detail**

**APPLICATION**

public static final MessageContext.Scope APPLICATION

**HANDLER**

public static final MessageContext.Scope HANDLER

**Method Detail**

final public static MessageContext.Scope[] values()
values

public static final MessageContext.Scope[] values()

    Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

    for(MessageContext.Scope c : MessageContext.Scope.values())
        System.out.println(c);

    Returns:
    an array containing the constants of this enum type, in the order they're declared

public static MessageContext.Scope.valueOf(String name)

valueOf

public static MessageContext.Scope.valueOf(String name)

    Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
    name - the name of the enum constant to be returned.

Returns:
    the enum constant with the specified name

Throws:
    IllegalArgumentException - if this enum type has no constant with the specified name
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.event Class MessageCountAdapter

java.lang.Object
   ^ javax.mail.event.MessageCountAdapter

All Implemented Interfaces:
   java.util.EventListener, javax.mail.event.MessageCountListener

public abstract class MessageCountAdapter
    extends Object
    implements MessageCountListener

Implements: MessageCountListener

MessageCount

The adapter which receives MessageCount events. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.

Author:
   John Mani

Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageCountAdapter()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void messagesAdded(MessageCountEvent e)</td>
<td>Invoked when messages are added into a folder.</td>
</tr>
<tr>
<td>void messagesRemoved(MessageCountEvent e)</td>
<td></td>
</tr>
</tbody>
</table>
void Invoked when messages are removed (expunged) from a folder.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public MessageCountAdapter()

MessageCountAdapter

public MessageCountAdapter()

Method Detail

public void messagesAdded(MessageCountEvent e)

messagesAdded

public void messagesAdded(MessageCountEvent e)

Description copied from interface: MessageCountListener
Invoked when messages are added into a folder.

Specified by:
messagesAdded in interface MessageCountListener

public void messagesRemoved(MessageCountEvent e)
public void messagesRemoved(MessageCountEvent e)

Description copied from interface: MessageCountListener
Invoked when messages are removed (expunged) from a folder.

Specified by:
messagesRemoved in interface MessageCountListener
javax.mail.event  **Class MessageCountEvent**

**java.lang.Object**  
  ↳ **java.util.EventObject**  
    ↳ **javax.mail.event.MailEvent**  
      ↳ **javax.mail.event.MessageCountEvent**

**All Implemented Interfaces:**  
  **Serializable**

```java
public class MessageCountEvent
  extends MailEvent

**Extends:** java.util.EventObject > MailEvent
```

MessageCountEvent IMAP MessageCountEvent  
RFC 2060  
getMessageCount  
isConnected  ""

This class notifies changes in the number of messages in a folder.

Note that some folder types may only deliver MessageCountEvents at certain times or after certain operations. IMAP in particular will only notify the client of MessageCountEvents when a client issues a new command. Refer to RFC 2060 [http://www.ietf.org/rfc/rfc2060.txt](http://www.ietf.org/rfc/rfc2060.txt) for details. A client may want "poll" the folder by occasionally calling the `getMessageCount` or `isConnected` methods to solicit any such notifications.

**Author:**  
  John Mani

**See Also:**  
  **Serialized Form**
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int <code>ADDED</code></td>
<td>The messages were added to their folder</td>
</tr>
<tr>
<td>protected <code>msgs</code> Message[]</td>
<td>The messages.</td>
</tr>
<tr>
<td>protected boolean <code>removed</code></td>
<td>If true, this event is the result of an explicit expunge by this client, and the messages in this folder have been renumbered to account for this.</td>
</tr>
<tr>
<td>static int <code>REMOVED</code></td>
<td>The messages were removed from their folder</td>
</tr>
<tr>
<td>protected int <code>type</code></td>
<td>The event type.</td>
</tr>
</tbody>
</table>

### Fields inherited from class java.util.EventObject

- `source`

### Constructor Summary

**MessageCountEvent**

```java
MessageCountEvent(Folder folder, int type, boolean removed, Message[] msgs)
```

Constructor.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>dispatch</code>(Object listener)</td>
<td>Invokes the appropriate MessageCountListener method.</td>
</tr>
<tr>
<td>Message[] <code>getMessages</code>()</td>
<td>Return the array of messages added or removed.</td>
</tr>
<tr>
<td>int <code>getType</code>()</td>
<td>Return the type of this event.</td>
</tr>
<tr>
<td>boolean <code>isRemoved</code>()</td>
<td>Indicates whether this event is the result of an explicit expunge by this client, or due to an expunge from external sources.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.util.EventObject
getSource, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

**ADDED**

public static final int **ADDED**

The messages were added to their folder

**See Also:**

Constant Field Values

**REMOVED**

public static final int **REMOVED**

The messages were removed from their folder

**See Also:**

Constant Field Values
protected int type

    The event type.

protected boolean removed

    If true, this event is the result of an explicit expunge by this client, and the messages in this folder have been renumbered to account for this. If false, this event is the result of an expunge by external sources.

protected transient Message[] msgs

    The messages.

Constructor Detail

public MessageCountEvent(Folder folder, int type, boolean removed, Message[]msgs)

    folder
    type
    removed       true false
    msgs          /
MessageCountEvent

public MessageCountEvent(Folder folder, int type, boolean removed, Message[] msgs)

Constructor.

Parameters:
folder - The containing folder
type - The event type
removed - If true, this event is the result of an explicit expunge by this client, and the messages in this folder have been renumbered to account for this. If false, this event is the result of an expunge by external sources.
msgs - The messages added/removed

Method Detail

public int getType()

    return

getype

public int getType()

    Return the type of this event.

    Returns:
    type

class public boolean isRemoved()

    true
**isRemoved**

```java
public boolean isRemoved()
```

Indicates whether this event is the result of an explicit expunge by this client, or due to an expunge from external sources. If `true`, this event is due to an explicit expunge and hence all remaining messages in this folder have been renumbered. If `false`, this event is due to an external expunge.

Note that this method is valid only if the type of this event is `REMOVED`.

---

**getMessages**

```java
public Message[] getMessages()
```

Return the array of messages added or removed.

---

**dispatch**

```java
public void dispatch(Object listener)
```

Dispatch

**MessageCountListener**

---

**dispatch**
public void dispatch(Object listener)

Invokes the appropriate MessageCountListener method.

Specified by: dispatch in class MailEvent
java.mail.event Interface MessageCountListener

All Superinterfaces:
   EventListener

All Known Implementing Classes:
   MessageCountAdapter

public interface MessageCountListener
   extends EventListener

   Implements: java.util.EventListener
   Implemented by: MessageCountAdapter

   MessageCount Listener

This is the Listener interface for MessageCount events.

Author:
   John Mani

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void messagesAdded(MessageCountEvent e)</td>
</tr>
<tr>
<td>Invoked when messages are added into a folder.</td>
</tr>
<tr>
<td>void messagesRemoved(MessageCountEvent e)</td>
</tr>
<tr>
<td>Invoked when messages are removed (expunged) from a folder.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void messagesAdded(MessageCountEvent e)</td>
</tr>
</tbody>
</table>
messagesAdded

void messagesAdded(MessageCountEvent e)

Invoked when messages are added into a folder.

public void messagesRemoved(MessageCountEvent e)

messagesRemoved

void messagesRemoved(MessageCountEvent e)

Invoked when messages are removed (expunged) from a folder.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Annotation Type MessageDriven

```java
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface MessageDriven

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean

Component-defining annotation for a message driven bean.

### Optional Element Summary

<table>
<thead>
<tr>
<th>ActivationConfigProperty[]</th>
<th>activationConfig</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Activation config properties.</td>
</tr>
<tr>
<td>String</td>
<td>description</td>
</tr>
<tr>
<td>String</td>
<td>mappedName</td>
</tr>
<tr>
<td></td>
<td>A product specific name(e.g. global JNDI name of a queue) that this message-driven bean should be mapped to.</td>
</tr>
<tr>
<td>Class</td>
<td>messageListenerInterface</td>
</tr>
<tr>
<td></td>
<td>Message-listener interface.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
</tr>
<tr>
<td></td>
<td>ejb-name for this bean.</td>
</tr>
</tbody>
</table>

abstract public String name()
Bean  ejb-name
**name**

```java
public abstract String name
```

ejb-name for this bean.

**Default:**
```
"
```

---

**abstract public Class<T> messageListenerInterface()**

---

**messageListenerInterface**

```java
public abstract Class messageListenerInterface
```

Message-listener interface.

**Default:**
```
java.lang.Object.class
```

---

**abstract public ActivationConfigProperty[] activationConfig()**

```java
public abstract ActivationConfigProperty[] activationConfig
```

Activation config properties.

**Default:**
```
{}
```
abstract public String mappedName()
Bean JNDI

mappedName

public abstract String mappedName

A product specific name (e.g. global JNDI name of a queue) that this message-driven bean should be mapped to. Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

Default:"

abstract public String description()

description

public abstract String description

Default:"

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS :
Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD
javax.ejb Interface MessageDrivenBean

All Superinterfaces: 
   EnterpriseBean, Serializable

public interface MessageDrivenBean
extends EnterpriseBean

Implements: EnterpriseBean

MessageDrivenBean Bean MessageDrivenBean Bean

The MessageDrivenBean interface is implemented by every message-driven enterprise Bean class. The container uses the MessageDrivenBean methods to notify the enterprise Bean instances of the instance's life cycle events.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void ejbRemove()</strong></td>
</tr>
<tr>
<td>A container invokes this method before it ends the life of the message-driven object.</td>
</tr>
<tr>
<td><strong>void setMessageDrivenContext(MessageDrivenContext ctx)</strong></td>
</tr>
<tr>
<td>Set the associated message-driven context.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void setMessageDrivenContext(MessageDrivenContext ctx)</td>
</tr>
<tr>
<td>throws EJBException</td>
</tr>
</tbody>
</table>
Bean

`ctx` MessageDrivenContext

Throws **EJBException**: 

**setMessageDrivenContext**

```java
void setMessageDrivenContext(MessageDrivenContext ctx)
throws EJBException
```

Set the associated message-driven context. The container calls this
text after the instance creation.

The enterprise Bean instance should store the reference to the context object in an instance variable.

This method is called with no transaction context.

**Parameters:**

- `ctx` - A MessageDrivenContext interface for the instance.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.

---

**public void ejbRemove() throws EJBException**

```java
Throws EJBException:
```
**ejbRemove**

```java
void ejbRemove() throws EJBException
```

A container invokes this method before it ends the life of the message-driven object. This happens when a container decides to terminate the message-driven object.

This method is called with no transaction context.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FRAMES NO FRAMES</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DETAIL: FIELD CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.management.j2ee.statistics Interface
MessageDrivenBeanStats

All Superinterfaces:
EJBStats, Stats

public interface MessageDrivenBeanStats
extends EJBStats
Implements: EJBStats

Bean

Specifies the statistics provided by a message driven bean.

Method Summary

<table>
<thead>
<tr>
<th>CountStatistic</th>
<th>getMessageCount()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number of messages received.</td>
</tr>
</tbody>
</table>

Methods inherited from interface
djavax.management.j2ee.statistics.EJBStats
getCreateCount, getRemoveCount

Methods inherited from interface
djavax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics

Method Detail
public CountStatistic getMessageCount()

getMessageCount

CountStatistic getMessageCount()  

Number of messages received.
javax.ejb Interface MessageDrivenContext

All Superinterfaces:
   EJBContext

public interface MessageDrivenContext
extends EJBContext

Implements: EJBContext

MessageDrivenContext Bean MessageDrivenContext

The MessageDrivenContext interface provides access to the runtime message-driven context that the container provides for a message-driven enterprise Bean instance. The container passes the MessageDrivenContext interface to an instance after the instance has been created. The message-driven context remains associated with the instance for the lifetime of the instance.

Method Summary

Methods inherited from interface javax.ejb.EJBContext
getCallerIdentity, getCallerPrincipal, getEJBHome,
getEJBLocalHome, getEnvironment, getRollbackOnly, getTimerService,
getUserTransaction, isCallerInRole, isCallerInRole, lookup,
setRollbackOnly

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface MessageEndpoint

version 1.0

This defines a contract for a message endpoint. This is implemented by an application server.

**Version:**
1.0

**Author:**
Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void afterDelivery()</td>
</tr>
<tr>
<td>This is called by a resource adapter after a message is delivered.</td>
</tr>
<tr>
<td>void beforeDelivery(Method method)</td>
</tr>
<tr>
<td>This is called by a resource adapter before a message is delivered.</td>
</tr>
<tr>
<td>void release()</td>
</tr>
<tr>
<td>This method may be called by the resource adapter to indicate that it no longer needs a proxy endpoint instance.</td>
</tr>
</tbody>
</table>

**Method Detail**

public void beforeDelivery(Method method) throws
NoSuchMethodException, ResourceException

method
Throws NoSuchMethodException:
Throws ResourceException:
Throws ApplicationServerInternalException:
Throws IllegalStateException:
Throws UnavailableException:

beforeDelivery

void beforeDelivery(Method method)
throws NoSuchMethodException,
ResourceException

This is called by a resource adapter before a message is delivered.

Parameters:
method - description of a target method. This information about the intended target method allows an application server to decide whether to start a transaction during this method call, depending on the transaction preferences of the target method. The processing (by the application server) of the actual message delivery method call on the endpoint must be independent of the class loader associated with this descriptive method object.

Throws:
NoSuchMethodException - indicates that the specified method does not exist on the target endpoint.
ResourceException - generic exception.
ApplicationServerInternalException - indicates an error condition in the application server.
IllegalStateException - indicates that the endpoint is in an illegal state for the method invocation. For example, this occurs when beforeDelivery and afterDelivery method calls are not paired.
public void afterDelivery() throws ResourceException

<table>
<thead>
<tr>
<th>Throws</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceException:</td>
</tr>
<tr>
<td>ApplicationServerInternalException:</td>
</tr>
<tr>
<td>IllegalStateException: beforeDelivery afterDelivery</td>
</tr>
<tr>
<td>UnavailableException:</td>
</tr>
</tbody>
</table>

This is called by a resource adapter after a message is delivered.

Throws:

- ResourceException - generic exception.
- ApplicationServerInternalException - indicates an error condition in the application server.
- IllegalStateException - indicates that the endpoint is in an illegal state for the method invocation. For example, this occurs when beforeDelivery and afterDelivery method calls are not paired.
- UnavailableException - indicates that the endpoint is not available.

public void release()
void release() 

This method may be called by the resource adapter to indicate that it no longer needs a proxy endpoint instance. This hint may be used by the application server for endpoint pooling decisions.
javax.resource.spi.endpoint Interface MessageEndpointFactory

public interface MessageEndpointFactory

version 1.0

This serves as a factory for creating message endpoints.

Version: 1.0
Author: Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createEndpoint(XAResource xaResource)</code></td>
</tr>
<tr>
<td><code>isDeliveryTransacted(Method method)</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public <code>MessageEndpoint createEndpoint(XAResource xaResource)</code> throws <code>UnavailableException</code></td>
</tr>
</tbody>
</table>
createEndpoint

**MessageEndpoint** createEndpoint(XAResource xaResource)

Throws UnavailableException:

This is used to create a message endpoint. The message endpoint is expected to implement the correct message listener type.

**Parameters:**
- xaResource - an optional XAResource instance used to get transaction notifications when the message delivery is transacted.

**Returns:**
- a message endpoint instance.

**Throws:**
- UnavailableException - indicates a transient failure in creating a message endpoint. Subsequent attempts to create a message endpoint might succeed.

public boolean isDeliveryTransacted(Method method)

Throws NoSuchMethodException:

**Method**

Returns true

isDeliveryTransacted

boolean isDeliveryTransacted(Method method)

Throws NoSuchMethodException:
This is used to find out whether message deliveries to a target method on a message listener interface that is implemented by a message endpoint will be transacted or not. The message endpoint may indicate its transacted delivery preferences (at a per method level) through its deployment descriptor. The message delivery preferences must not change during the lifetime of a message endpoint.

**Parameters:**
- **method**: description of a target method. This information about the intended target method allows an application server to find out whether the target method call will be transacted or not.

**Returns:**
- true, if message endpoint requires transacted message delivery.

**Throws:**
- **NoSuchMethodException** - indicates that the specified method does not exist on the target endpoint.
javax.jms Class MessageEOFException

java.lang.Object
   \u2192 java.lang.Throwable
      \u2192 java.lang.Exception
         \u2192 javax.jms.JMSException
            \u2192 javax.jms.MessageEOFException

All Implemented Interfaces:
   Serializable

public class MessageEOFException extends JMSException

Extends: Throwable > Exception > JMSException

StreamMessage  BytesMessage
version  26 August 1998

This exception must be thrown when an unexpected end of stream has been reached when a StreamMessage or BytesMessage is being read.

Version:
   26 August 1998

Author:
   Rahul Sharma

See Also:
   Serialized Form

Constructor Summary

MessageEOFException(String reason)
   Constructs a MessageEOFException with the specified reason.
MessageEOFException(String reason, String errorCode)

Constructs a MessageEOFException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MessageEOFException(String reason, String errorCode)

MessageEOFException

reason

errorCode

MessageEOFException

public MessageEOFException(String reason, String errorCode)

Constructs a MessageEOFException with the specified reason and error code.
public MessageEOFException(String reason)

MessageEOFException

public MessageEOFException(String reason)

Constructs a MessageEOFException with the specified reason. The error code defaults to null.

Parameters:
reason - a description of the exception
javax.xml.soap  **Class MessageFactory**

```java
java.lang.Object
  ^ javax.xml.soap.MessageFactory
public abstract class MessageFactory
  extends Object
```

A **factory for creating SOAPMessage objects.**

A SAAJ client can create a `MessageFactory` object using the method

```java
MessageFactory mf = MessageFactory.newInstance();
MessageFactory mf12 = MessageFactory.newInstance(SOAPConstants.SOAP_...
```

- `SOAPPart`
- `SOAPEnvelope`
- `SOAPBody`
- `SOAPHeader`

```java
MessageFactory.createMessage()
MessageFactory.createMessage(MimeHeaders, java.io.InputStream) --
InputStream   MimeHeaders
```

```java
MessageFactory
  newInstance
  MessageFactory
```
newInstance, as shown in the following lines of code.

```java
MessageFactory mf = MessageFactory.newInstance();
MessageFactory mf12 = MessageFactory.newInstance(SOAPConstants.SOAP_1_2_PROTOCOL);
```

All `MessageFactory` objects, regardless of how they are created, will produce `SOAPMessage` objects that have the following elements by default:

- A `SOAPPart` object
- A `SOAPEnvelope` object
- A `SOAPBody` object
- A `SOAPHeader` object

In some cases, specialized `MessageFactory` objects may be obtained that produce messages prepopulated with additional entries in the `SOAPHeader` object and the `SOAPBody` object. The content of a new `SOAPMessage` object depends on which of the two `MessageFactory` methods is used to create it.

- `createMessage()`
  This is the method clients would normally use to create a request message.
- `createMessage(MimeHeaders, java.io.InputStream)` -- message has content from the `InputStream` object and headers from the `MimeHeaders` object
  This method can be used internally by a service implementation to create a message that is a response to a request.

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>MessageFactory()</th>
</tr>
</thead>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>createMessage()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Creates a new <code>SOAPMessage</code> object with the</td>
</tr>
<tr>
<td>abstract <code>SOAPMessage</code></td>
</tr>
</tbody>
</table>
SOAPPart, SOAPEnvelope, SOAPBody, and SOAPHeader objects.

Create Message

```java
createMessage(MimeHeaders headers, InputStream in)
```

Internalizes the contents of the given InputStream object into a new SOAPMessage object and returns the SOAPMessage object.

Message Factory

```java
newInstance()
```

Creates a new MessageFactory object that is an instance of the default implementation (SOAP 1.1). This method uses the following ordered lookup procedure to determine the MessageFactory implementation class to load: Use the javax.xml.soap.MessageFactory system property.

```java
newInstance(String protocol)
```

Creates a new MessageFactory object that is an instance of the specified implementation.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

```java
public MessageFactory()
```

MessageFactory

```java
public MessageFactory()
```

Method Detail

```java
public static MessageFactory newInstance() throws SOAPException
```
MessageFactory (SOAP 1.1)

- `javax.xml.soap.MessageFactory`
- JRE "lib/jaxm.properties" `java.util.Properties`

- Services API JAR Services API jar
  META-INF/services/javax.xml.soap.MessageFactory

- `SAAJMetaFactory MessageFactory`

  ```java
  return MessageFactory
  ```

  Throws `SOAPException: MessageFactory`

  See also `javax.xml.soap.SAAJMetaFactory`

`newInstance`

```java
public static MessageFactory newInstance() throws SOAPException
```

Creates a new `MessageFactory` object that is an instance of the default implementation (SOAP 1.1). This method uses the following ordered lookup procedure to determine the `MessageFactory` implementation class to load:

- Use the `javax.xml.soap.MessageFactory` system property.
- Use the properties file "lib/jaxm.properties" in the JRE directory. This configuration file is in standard `java.util.Properties` format and contains the fully qualified name of the implementation class with the key being the system property defined above.
- Use the Services API (as detailed in the JAR specification), if available, to determine the classname. The Services API will look for a classname in the file META-INF/services/javax.xml.soap.MessageFactory in jars available to the runtime.
- Use the `SAAJMetaFactory` instance to locate the `MessageFactory` implementation class.
Returns:
a new instance of a MessageFactory

Throws:
SOAPException - if there was an error in creating the default implementation of the MessageFactory.

See Also:
SAAJMetaFactory

---

```java
public static MessageFactory newInstance(String protocol)
throws SOAPException

MessageFactory SOAP 1.1  SOAP 1.2
createMessage MIME  SAAJMetaFactory
MessageFactory
return MessageFactory
protocol DYNAMIC_SOAP_PROTOCOLDEFAULT_
SOAP_1_1_PROTOCOL SOAP_1_2_PROTOCOL

Throws
SOAPException: MessageFactory
since SAAJ 1.3
See also javax.xml.soap.SAAJMetaFactory
```

**newInstance**

```java
public static MessageFactory newInstance(String protocol)
throws SOAPException

Creates a new MessageFactory object that is an instance of the specified implementation. May be a dynamic message factory, a SOAP 1.1 message factory, or a SOAP 1.2 message factory. A dynamic message factory creates messages based on the MIME headers specified as arguments to the createMessage method. This method uses the SAAJMetaFactory to locate the implementation class and create the MessageFactory instance.

Parameters:
protocol - a string constant representing the class of the
specified message factory implementation. May be either
DYNAMIC_SOAP_PROTOCOL, DEFAULT_SOAP_PROTOCOL (which is the
same as) SOAP_1_1_PROTOCOL, OR SOAP_1_2_PROTOCOL.

Returns:
a new instance of a MessageFactory

Throws:
SOAPException - if there was an error in creating the specified
implementation of MessageFactory.

Since:
SAAJ 1.3

See Also:
SAAJMetaFactory

abstract public SOAPMessage createMessage() throws
SOAPException

SOAPPart SOAPEnvelope SOAPBody SOAPHeader SOAPMessage

SOAPPart SOAP "" SOAPMessage
AttachmentPart XML AttachmentPart SOAPMessage

Throws
SOAPException: SOAP

Throws
UnsupportedOperationException: MessageFactory

DYNAMIC_SOAP_PROTOCOL

createMessage

public abstract SOAPMessage createMessage() throws SOAPException

Creates a new SOAPMessage object with the default SOAPPart,
SOAPEnvelope, SOAPBody, and SOAPHeader objects. Profile-specific
message factories can choose to prepopulate the SOAPMessage object
with profile-specific headers.
Content can be added to this message's SOAPPart object, and the message can be sent "as is" when a message containing only a SOAP part is sufficient. Otherwise, the SOAPMessage object needs to create one or more AttachmentPart objects and add them to itself. Any content that is not in XML format must be in an AttachmentPart object.

**Returns:**

- a new SOAPMessage object

**Throws:**

- SOAPException - if a SOAP error occurs
- UnsupportedOperationException - if the protocol of this MessageFactory instance IS DYNAMIC_SOAP_PROTOCOL

```java
abstract public SOAPMessage createMessage(MimeHeaders headers, java.io.InputStream in) throws java.io.IOException, SOAPException

InputStream SOAPMessage SOAPMessage
in
headers

return SOAPMessage InputStream

Throws java.io.IOException:

Throws SOAPException:

IllegalArgumentException: MessageFactory MIME

Throws headers SOAP_1_1_PROTOCOL SOAP_1_2_PROTOCOL

MessageFactory IllegalArgumentException

createMessage

public abstract SOAPMessage createMessage(MimeHeaders headers, InputStream in) throws IOException, SOAPException

Internalizes the contents of the given InputStream object into a new SOAPMessage object and returns the SOAPMessage object.
Parameters:

in - the InputStream object that contains the data for a message
headers - the transport-specific headers passed to the message
in a transport-independent fashion for creation of the message

Returns:

a new SOAPMessage object containing the data from the given
InputStream object

Throws:

IOException - if there is a problem in reading data from the input
stream
SOAPException - may be thrown if the message is invalid
IllegalArgumentException - if the MessageFactory requires one
or more MIME headers to be present in the headers parameter
and they are missing. MessageFactory implementations for
SOAP_1_1_PROTOCOL or SOAP_1_2_PROTOCOL must not throw
IllegalArgumentException for this reason.
javax.jms Class MessageFormatException

java.lang.Object
   ↳ java.lang.Throwable
      ↳ java.lang.Exception
         ↳ javax.jms.JMSException
            ↳ javax.jms.MessageFormatException

All Implemented Interfaces:
   Serializable

public class MessageFormatException
    extends JMSException

Extends: Throwable > Exception > JMSException

This exception must be thrown when a JMS client attempts to use a data type not supported by a message or attempts to read data in a message as the wrong type. It must also be thrown when equivalent type errors are made with message property values. For example, this exception must be thrown if StreamMessage.writeObject is given an unsupported class or if StreamMessage.readShort is used to read a boolean value. Note that the special case of a failure caused by an attempt to read improperly formatted string data as numeric values must throw the java.lang.NumberFormatException.

Version:
   26 August 1998

Author:
## Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MessageFormatException(String reason)</code></td>
<td>Constructs a <code>MessageFormatException</code> with the specified reason.</td>
</tr>
<tr>
<td><code>MessageFormatException(String reason, String errorCode)</code></td>
<td>Constructs a <code>MessageFormatException</code> with the specified reason and error code.</td>
</tr>
</tbody>
</table>

## Method Summary

Methods inherited from class `javax.jms.JMSException`:
- `getErrorCode`, `getLinkedException`, `setLinkedException`

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getCause`, `getLocaleMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

```java
public MessageFormatException(String reason, String errorCode)
```

`MessageFormatException`

`reason`

`errorCode`
MessageFormatException

public MessageFormatException(String reason, String errorCode)

Constructs a MessageFormatException with the specified reason and error code.

Parameters:
reason - a description of the exception
errorCode - a string specifying the vendor-specific error code

public MessageFormatException(String reason)
MessageFormatException null
reason

MessageFormatException

public MessageFormatException(String reason)

Constructs a MessageFormatException with the specified reason. The error code defaults to null.

Parameters:
reason - a description of the exception

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.mail.search  **Class MessageIDTerm**

`java.lang.Object`  
  ↓ `javax.mail.search.SearchTerm`  
    ↓ `javax.mail.search.StringTerm`  
        ↓ `javax.mail.search.MessageIDTerm`

**All Implemented Interfaces:**  
  [Serializable](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/Serializable.html)

```java
public final class MessageIDTerm
  extends StringTerm
```

**Extends:** [SearchTerm](https://docs.oracle.com/en/java/javase/11/docs/api/java.mail/javax/mail/search/SearchTerm.html) > [StringTerm](https://docs.oracle.com/en/java/javase/11/docs/api/java.mail/javax/mail/search/StringTerm.html)

RFC822 “MessageId” - Internet “-ID”“-ID”  
MessageId

MessageId  String

This term models the RFC822 "MessageId" - a message-id for Internet messages that is supposed to be unique per message. Clients can use this term to search a folder for a message given its MessageId.

The MessageId is represented as a String.

**Author:**  
Bill Shannon, John Mani

**See Also:**  
[Serialized Form](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/Serializable.html)

---

**Field Summary**
Fields inherited from class javax.mail.search.StringTerm
ignoreCase, pattern

Constructor Summary

public MessageIDTerm(String msgid)

Method Summary

public MessageIDTerm(String msgid)

Constructor.
Parameters:
  msgid - the msgid to search for

Method Detail

public boolean match(Message msg)

match

The match method.

Specified by:
  match in class SearchTerm

Parameters:
  msg - the match is applied to this Message's Message-ID header

Returns:
  true if the match succeeds, otherwise false

public boolean equals(Object obj)

equals

public boolean equals(Object obj)

Equality comparison.

Overrides:
  equals in class StringTerm
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
A `MessageListener` object is used to receive asynchronously delivered messages.

Each session must insure that it passes messages serially to the listener. This means that a listener assigned to one or more consumers of the same session can assume that the `onMessage` method is not called with the next message until the session has completed the last call.

**Version:**
1.0 - 13 March 1998

**Author:**
Mark Hapner, Rich Burridge

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>onMessage</code></td>
<td><code>void onMessage(Message message)</code></td>
<td>Passes a message to the listener.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void onMessage(Message message)
```
onMessage

void onMessage(Message message)

Passes a message to the listener.

Parameters:
message - the message passed to the listener
Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY: NESTED  |  FIELD  |  CONSTR  |  METHOD
FRAMES  NO FRAMES
DETAIL: FIELD  |  CONSTR  |  METHOD
public interface MessageListener

Bean- EIS -

    version 1.0

This serves as a request-response message listener type that message endpoints (message-driven beans) may implement. This allows an EIS to communicate with an endpoint using a request-response style.

Version:
    1.0
Author:
    Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>onMessage</td>
<td>Record onMessage(Record inputData)</td>
<td>This method allows an EIS to call a message endpoint using a request-response style communication.</td>
</tr>
</tbody>
</table>

Method Detail

public Record onMessage(Record inputData) throws ResourceException

EIS -

    inputData Record
    return Record null
Throws ResourceException:
onMessage

Record onMessage(Record inputData)
throws ResourceException

This method allows an EIS to call a message endpoint using a request-response style communication.

**Parameters:**

inputData - a Record instance.

**Returns:**

a Record instance or null.

**Throws:**

ResourceException - indicates an exceptional condition.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jms Class MessageNotReadableException

java.lang.Object  
  └ java.lang.Throwable  
    └ java.lang.Exception  
      └ javax.jms.JMSException  
        └ javax.jms.MessageNotReadableException

All Implemented Interfaces:
  Serializable

public class MessageNotReadableException
  extends JMSException

Extends: Throwable > Exception > JMSException

JMS  
  version 26 August 1998

This exception must be thrown when a JMS client attempts to read a write-only message.

Version:  
  26 August 1998

Author:  
  Rahul Sharma

See Also:  
  Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageNotReadableException(String reason)</td>
<td>Constructs a MessageNotReadableException with the specified reason</td>
</tr>
</tbody>
</table>
Constructs a MessageNotReadableException with the specified reason and error code.

**Method Summary**

Methods inherited from class `javax.jms.JMSException`:
- `getErrorCode`, `getLinkedException`, `setLinkedException`

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getCause`, `getLocaleMessage`, `getMessage`,
- `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,
- `wait`, `wait`, `wait`

**Constructor Detail**

```java
public MessageNotReadableException(String reason, String errorCode)

    MessageNotReadableException
    reason
    errorCode

MessageNotReadableException

public MessageNotReadableException(String reason, String errorCode)

    Constructs a MessageNotReadableException with the specified reason
and error code.

**Parameters:**
- reason - a description of the exception
- errorCode - a string specifying the vendor-specific error code

```java
public MessageNotReadableException(String reason)
    MessageNotReadableException null
    reason
```

`MessageNotReadableException`

```java
public MessageNotReadableException(String reason)
    Constructs a MessageNotReadableException with the specified reason. The error code defaults to null.

**Parameters:**
- reason - a description of the exception
```

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
<th>FRAMES</th>
<th>NO FRAMES</th>
<th>DETAIL: FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.jms Class MessageNotWriteableException

java.lang.Object
   └ java.lang.Throwable
       └ java.lang.Exception
           └ javax.jms.JMSException
               └ javax.jms.MessageNotWriteableException

All Implemented Interfaces:
   Serializable

public class MessageNotWriteableException
   extends JMSException

Extends: Throwable > Exception > JMSException

JMS version 26 August 1998

This exception must be thrown when a JMS client attempts to write to a read-only message.

Version: 26 August 1998
Author: Rahul Sharma
See Also: Serialized Form

Constructor Summary

MessageNotWriteableException(String reason)
   Constructs a MessageNotWriteableException with the specified
MessageNotWriteableException(String reason, String errorCode)

Constructs a MessageNotWriteableException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MessageNotWriteableException(String reason, String errorCode)

MessageNotWriteableException

reason

errorCode

MessageNotWriteableException

public MessageNotWriteableException(String reason, String errorCode)

Constructs a MessageNotWriteableException with the specified
reason and error code.

**Parameters:**
- `reason` - a description of the exception
- `errorCode` - a string specifying the vendor-specific error code

```java
public MessageNotWriteableException(String reason)
```

```java
public MessageNotWriteableException(String reason)
```

Constructs a `MessageNotWriteableException` with the specified reason. The error code defaults to null.

**Parameters:**
- `reason` - a description of the exception

---

**Overview** | **Package** | **Tree** | **Deprecated** | **Index** | **Help**
--- | --- | --- | --- | --- | ---
PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL | FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.search Class MessageNumberTerm

java.lang.Object
  ▼ javax.mail.search.SearchTerm
      ▼ javax.mail.search.ComparisonTerm
          ▼ javax.mail.search.IntegerComparisonTerm
              ▼ javax.mail.search.MessageNumberTerm

All Implemented Interfaces:
  Serializable

public final class MessageNumberTerm
  extends IntegerComparisonTerm

Extends: searchTerm > ComparisonTerm > IntegerComparisonTerm

Message

This class implements comparisons for Message numbers.

Author:
  Bill Shannon, John Mani

See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search&lt;IntegerComparisonTerm&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.ComparisonTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison, EQ, GE, GT, LE, LT, NE</td>
</tr>
</tbody>
</table>
## Constructor Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MessageNumberTerm(int number)</code></td>
<td>Constructor.</td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean equals(Object obj)</code></td>
<td>Equality comparison.</td>
<td></td>
</tr>
<tr>
<td><code>boolean match(Message msg)</code></td>
<td>The match method.</td>
<td></td>
</tr>
</tbody>
</table>

## Methods inherited from class `javax.mail.search.IntegerComparisonTerm`
- `getComparison`, `getNumber`, `hashCode`, `match`

## Methods inherited from class `java.lang.Object`
- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

## Constructor Detail

```java
public MessageNumberTerm(int number)
```

`number` Message

## MessageNumberTerm

```java
public MessageNumberTerm(int number)
```

Constructor.

**Parameters:**
number - the Message number

### Method Detail

**public boolean match(Message msg)**

<table>
<thead>
<tr>
<th>msg</th>
<th>Message</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

**match**

public boolean match(Message msg)

The match method.

**Specified by:**

- [match](#) in class [SearchTerm](#)

**Parameters:**

- msg - the Message number is matched with this Message

**Returns:**

- true if the match succeeds, otherwise false

---

**public boolean equals(Object obj)**

**equals**

public boolean equals(Object obj)

Equality comparison.

**Overrides:**

- [equals](#) in class [IntegerComparisonTerm](#)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface MessageProducer

All Known Subinterfaces:
  QueueSender, TopicPublisher

public interface MessageProducer

Implemented by: QueueSender, TopicPublisher

  MessageProducer  Destination -  MessageProducer

  JMSReplyTo

GMT

JMS JMS API

version 1.1 - February 2, 2002

See also javax.jms.TopicPublisher, javax.jms.QueueSender, createProducer

A client uses a MessageProducer object to send messages to a destination. A MessageProducer object is created by passing a Destination object to a message-producer creation method supplied by a session.

MessageProducer is the parent interface for all message producers.

A client also has the option of creating a message producer without supplying a destination. In this case, a destination must be provided with every send operation. A typical use for this kind of message producer is to send replies to requests using the request's JMSReplyTo destination.
A client can specify a default delivery mode, priority, and time to live for messages sent by a message producer. It can also specify the delivery mode, priority, and time to live for an individual message.

A client can specify a time-to-live value in milliseconds for each message it sends. This value defines a message expiration time that is the sum of the message's time-to-live and the GMT when it is sent (for transacted sends, this is the time the client sends the message, not the time the transaction is committed).

A JMS provider should do its best to expire messages accurately; however, the JMS API does not define the accuracy provided.

**Version:**
1.1 - February 2, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
- TopicPublisher
- QueueSender
- Session.createProducer(javax.jms.Destination)

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void close()</code></td>
<td>Closes the message producer.</td>
</tr>
<tr>
<td><code>int getDeliveryMode()</code></td>
<td>Gets the producer's default delivery mode.</td>
</tr>
<tr>
<td><code>Destination getDestination()</code></td>
<td>Gets the destination associated with this MessageProducer.</td>
</tr>
<tr>
<td><code>boolean getDisableMessageID()</code></td>
<td>Gets an indication of whether message IDs are disabled.</td>
</tr>
<tr>
<td><code>boolean getDisableMessageTimestamp()</code></td>
<td>Gets an indication of whether message timestamps are disabled.</td>
</tr>
<tr>
<td><code>getPriority()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>int</code></td>
<td>Gets the producer's default priority.</td>
</tr>
<tr>
<td><code>long</code></td>
<td>Gets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.</td>
</tr>
<tr>
<td><code>send(Destination destination, Message message)</code></td>
<td>Sends a message to a destination for an unidentified message producer.</td>
</tr>
<tr>
<td><code>send(Destination destination, Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Sends a message to a destination for an unidentified message producer, specifying delivery mode, priority and time to live.</td>
</tr>
<tr>
<td><code>send(Message message)</code></td>
<td>Sends a message using the MessageProducer's default delivery mode, priority, and time to live.</td>
</tr>
<tr>
<td><code>send(Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Sends a message to the destination, specifying delivery mode, priority, and time to live.</td>
</tr>
<tr>
<td><code>setDeliveryMode(int deliveryMode)</code></td>
<td>Sets the producer's default delivery mode.</td>
</tr>
<tr>
<td><code>setDisableMessageID(boolean value)</code></td>
<td>Sets whether message IDs are disabled.</td>
</tr>
<tr>
<td><code>setDisableMessageTimestamp(boolean value)</code></td>
<td>Sets whether message timestamps are disabled.</td>
</tr>
<tr>
<td><code>setPriority(int defaultPriority)</code></td>
<td>Sets the producer's default priority.</td>
</tr>
<tr>
<td><code>setTimeToLive(long timeToLive)</code></td>
<td>Sets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.</td>
</tr>
</tbody>
</table>
public void setDisableMessageID(boolean value) throws JMSException

Sets whether message IDs are disabled.

Since message IDs take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that the message ID is not used by an application. By calling the setDisableMessageID method on this message producer, a JMS client enables this potential optimization for all messages sent by this message producer. If the JMS provider accepts this hint, these messages must have the message ID set to null; if the provider ignores the hint, the message ID must be set to its normal unique value.

Message IDs are enabled by default.

Parameters:
value - indicates if message IDs are disabled

Throws:
JMSException - if the JMS provider fails to set message ID to disabled due to some internal error.
public boolean getDisableMessageID() throws JMSException

    ID

    return ID

    Throws JMSException: JMS ID

getDisableMessageID

boolean getDisableMessageID() throws JMSException

    Gets an indication of whether message IDs are disabled.

    Returns:
    an indication of whether message IDs are disabled

    Throws:
    JMSException - if the JMS provider fails to determine if message IDs are disabled due to some internal error.

public void setDisableMessageTimestamp(boolean value) throws JMSException

    JMS setDisableMessageTimestamp JMS

    value

    Throws JMSException: JMS

setDisableMessageTimestamp

void setDisableMessageTimestamp(boolean value)
Sets whether message timestamps are disabled.

Since timestamps take some effort to create and increase a message's size, some JMS providers may be able to optimize message overhead if they are given a hint that the timestamp is not used by an application. By calling the `setDisableMessageTimestamp` method on this message producer, a JMS client enables this potential optimization for all messages sent by this message producer. If the JMS provider accepts this hint, these messages must have the timestamp set to zero; if the provider ignores the hint, the timestamp must be set to its normal value.

Message timestamps are enabled by default.

**Parameters:**

- `value` - indicates if message timestamps are disabled

**Throws:**

- `JMSException` - if the JMS provider fails to set timestamps to disabled due to some internal error.

```
public boolean getDisableMessageTimestamp() throws JMSException {
    return false;
}
```

**Throws:**

- `JMSException` - JMS

**getDisableMessageTimestamp**

```
boolean getDisableMessageTimestamp() throws JMSException {
    return false;
}
```

Gets an indication of whether message timestamps are disabled.

**Returns:**

- an indication of whether message timestamps are disabled
**Throws:**

`JMSException` - if the JMS provider fails to determine if timestamps are disabled due to some internal error.

---

**public void setDeliveryMode(int deliveryMode)** throws `JMSException`

**PERSISTENT**

| deliveryMode | DeliveryMode.NON_PERSISTENT | DeliveryMode.PERSISTENT |

**Throws**  
`JMSException`: JMS

**See also**

getDeliveryMode, NON_PERSISTENT, PERSISTENT, DEFAULT_DELIVERY_MODE

---

**setDeliveryMode**

```java
void setDeliveryMode(int deliveryMode)
```

void **setDeliveryMode**(int deliveryMode)  
throws `JMSException`

Sets the producer's default delivery mode.

Delivery mode is set to PERSISTENT by default.

**Parameters:**

deliveryMode - the message delivery mode for this message producer; legal values are DeliveryMode.NON_PERSISTENT and DeliveryMode.PERSISTENT

**Throws:**

`JMSException` - if the JMS provider fails to set the delivery mode due to some internal error.

**See Also:**

getDeliveryMode(), DeliveryMode.NON_PERSISTENT, DeliveryMode.PERSISTENT, Message.DEFAULT_DELIVERY_MODE
public int getDeliveryMode() throws JMSException

    return

    Throws JMSException: JMS

    See also setDeliveryMode

getDeliveryMode

int getDeliveryMode() throws JMSException

    Gets the producer's default delivery mode.

    Returns: the message delivery mode for this message producer

    Throws: JMSException - if the JMS provider fails to get the delivery mode due to some internal error.

    See Also: setDeliveryMode(int)

public void setPriority(int defaultPriority) throws JMSException

JMS API 10 0 9 0-4 5-9 4

defaultPriority 0 9

    Throws JMSException: JMS

    See also getPriority, DEFAULT_PRIORITY

setPriority

void setPriority(int defaultPriority)

    throws JMSException
Sets the producer's default priority.

The JMS API defines ten levels of priority value, with 0 as the lowest priority and 9 as the highest. Clients should consider priorities 0-4 as gradations of normal priority and priorities 5-9 as gradations of expedited priority. Priority is set to 4 by default.

**Parameters:**
- `defaultPriority` - the message priority for this message producer; must be a value between 0 and 9

**Throws:**
- `JMSException` - if the JMS provider fails to set the priority due to some internal error.

**See Also:**
- `getPriority()`, `Message.DEFAULT_PRIORITY`
public void setTimeToLive(long timeToLive) throws JMSException

0

timeToLive 0

Throws JMSException: JMS
See also getTimeToLive, DEFAULT_TIME_TO_LIVE

setTimeToLive

void setTimeToLive(long timeToLive)
throws JMSException

Sets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

Time to live is set to zero by default.

Parameters:

timeToLive - the message time to live in milliseconds; zero is unlimited

Throws:

JMSException - if the JMS provider fails to set the time to live due to some internal error.

See Also:

getTimeToLive(), Message.DEFAULT_TIME_TO_LIVE

public long getTimeToLive() throws JMSException

return 0

Throws JMSException: JMS
See also setTimeToLive
**getTimeToLive**

```java
long getTimeToLive() throws JMSException
```

Gets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

**Returns:**
the message time to live in milliseconds; zero is unlimited

**Throws:**
JMSException - if the JMS provider fails to get the time to live due to some internal error.

**See Also:**
setTimeToLive(long)

---

**public Destination getDestination() throws JMSException**

```java
MessageProducer return Destination/
```

**Throws**
JMSException: JMSMessageProducer

since 1.1

---

**getDestination**

```java
Destination getDestination() throws JMSException
```

Gets the destination associated with this MessageProducer.

**Returns:**
this producer's Destination/

**Throws:**
JMSException - if the JMS provider fails to get the destination for this MessageProducer due to some internal error.
public void close() throws JMSException

Java MessageProducer
Throws JMSException: JMS

close

void close()
throws JMSException

Closes the message producer.

Since a provider may allocate some resources on behalf of a MessageProducer outside the Java virtual machine, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Throws:
JMSException - if the JMS provider fails to close the producer due to some internal error.

public void send(Message message) throws JMSException

MessageProducer
message
Throws JMSException: JMS
Throws MessageFormatException: MessageProducer
Throws InvalidDestinationException: MessageProducer
Throws UnsupportedOperationException: MessageProducer
since 1.1
See also createProducer, javax.jms.MessageProducer
send

void send(Message message)
    throws JMSException

Sends a message using the MessageProducer's default delivery mode, priority, and time to live.

Parameters:
message - the message to send

Throws:
    JMSException - if the JMS provider fails to send the message due to some internal error.
    MessageFormatException - if an invalid message is specified.
    InvalidDestinationException - if a client uses this method with a MessageProducer with an invalid destination.
    UnsupportedOperationException - if a client uses this method with a MessageProducer that did not specify a destination at creation time.

Since:
    1.1

See Also:
    Session.createProducer(javax.jms.Destination),
    MessageProducer

public void send(Message message, int deliveryMode, int priority, long timeToLive) throws JMSException

message
deliveryMode
priority
timeToLive

Throws
    JMSException: JMS
    MessageFormatException:
    InvalidDestinationException:
**send**

void `send(Message message, int deliveryMode, int priority, long timeToLive)`

throws `JMSException`

Sends a message to the destination, specifying delivery mode, priority, and time to live.

**Parameters:**
- `message` - the message to send
- `deliveryMode` - the delivery mode to use
- `priority` - the priority for this message
- `timeToLive` - the message's lifetime (in milliseconds)

**Throws:**
- `JMSException` - if the JMS provider fails to send the message due to some internal error.
- `MessageFormatException` - if an invalid message is specified.
- `InvalidDestinationException` - if a client uses this method with a `MessageProducer` with an invalid destination.
- `UnsupportedOperationException` - if a client uses this method with a `MessageProducer` that did not specify a destination at creation time.

**Since:**
1.1

**See Also:**
- `Session.createProducer(javax.jms.Destination)`

```
public void send(Destination destination, Message message) throws JMSException
```
send

void send(Destination destination, Message message) throws JMSException

Sends a message to a destination for an unidentified message producer. Uses the MessageProducer's default delivery mode, priority, and time to live.

Typically, a message producer is assigned a destination at creation time; however, the JMS API also supports unidentified message producers, which require that the destination be supplied every time a message is sent.

Parameters:
- destination - the destination to send this message to
- message - the message to send

Throws:
- JMSEception - if the JMS provider fails to send the message due to some internal error.
- MessageFormatException - if an invalid message is specified.
- InvalidDestinationException - if a client uses this method with
an invalid destination.

UnsupportedOperationException - if a client uses this method with a MessageProducer that specified a destination at creation time.

Since:
1.1

See Also:
Session.createProducer(javax.jms.Destination),
MessageProducer

public void send(Destination destination, Message message, int deliveryMode, int priority, long timeToLive) throws JMSException

JMS API

destination
message
deliveryMode
priority
timeToLive

Throws JMSException: JMS
Throws MessageFormatException:
Throws InvalidDestinationException:

since 1.1

See also createProducer

send

void send(Destination destination,
Message message,
int deliveryMode,
int priority,
long timeToLive)
throws JMSException
Sends a message to a destination for an unidentified message producer, specifying delivery mode, priority and time to live.

Typically, a message producer is assigned a destination at creation time; however, the JMS API also supports unidentified message producers, which require that the destination be supplied every time a message is sent.

**Parameters:**
- **destination** - the destination to send this message to
- **message** - the message to send
- **deliveryMode** - the delivery mode to use
- **priority** - the priority for this message
- **timeToLive** - the message's lifetime (in milliseconds)

**Throws:**
- **JMSException** - if the JMS provider fails to send the message due to some internal error.
- **MessageFormatException** - if an invalid message is specified.
- **InvalidDestinationException** - if a client uses this method with an invalid destination.

**Since:**
1.1

**See Also:**
- `Session.createProducer(javax.jms.Destination)`

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.mail Class MessageRemovedException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.mail.MessagingException
              └ javax.mail.MessageRemovedException

All Implemented Interfaces:
  Serializable

public class MessageRemovedException
extends MessagingException

The exception thrown when an invalid method is invoked on an expunged Message. The only valid methods on an expunged Message are isExpunged() and getMessageNumber().

Author:
  John Mani

See Also:
  Message.isExpunged(), Message.getMessageNumber(), Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageRemovedException()</td>
<td>Constructs a MessageRemovedException with no detail message.</td>
</tr>
<tr>
<td>MessageRemovedException(String s)</td>
<td>Constructs a MessageRemovedException with the specified detail message.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
**Constructor Detail**

**MessageRemovedException**

```java
public MessageRemovedException()
```

Constructs a MessageRemovedException with no detail message.

**MessageRemovedException**

```java
public MessageRemovedException(String s)
```

Constructs a MessageRemovedException with the specified detail message.

**Parameters:**

- s - the detail message
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Public class **MessagingException**

Extends: Exception

**Extends by:** AuthenticationFailedException, FolderClosedException, FolderNotFoundException, IllegalWriteException, MessageRemovedException, MethodNotSupportedException, NoSuchProviderException, ParseException, ReadOnlyFolderException, SearchException, SendFailedException, StoreClosedException

Messaging

The base class for all exceptions thrown by the Messaging classes

**Author:**
John Mani, Bill Shannon

**See Also:**
Serialized Form
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MessagingException()</code></td>
<td>Constructs a MessagingException with no detail message.</td>
</tr>
<tr>
<td><code>MessagingException(String s)</code></td>
<td>Constructs a MessagingException with the specified detail message.</td>
</tr>
<tr>
<td><code>MessagingException(String s, Exception e)</code></td>
<td>Constructs a MessagingException with the specified Exception and detail message.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Throwable.getCause()</code></td>
<td>Overrides the <code>getCause</code> method of <code>Throwable</code> to return the next exception in the chain of nested exceptions.</td>
</tr>
<tr>
<td><code>Exception.getNextException()</code></td>
<td>Get the next exception chained to this one.</td>
</tr>
<tr>
<td><code>boolean setNextException(Exception ex)</code></td>
<td>Add an exception to the end of the chain.</td>
</tr>
<tr>
<td><code>String toString()</code></td>
<td>Override <code>toString</code> method to provide information on nested exceptions.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
Constructor Detail

public MessagingException()
MessagingException

MessagingException

public MessagingException()
    Constructs a MessagingException with no detail message.

public MessagingException(String s)
MessagingException
    s

MessagingException

public MessagingException(String s)
    Constructs a MessagingException with the specified detail message.

    Parameters:
    s - the detail message

public MessagingException(String s, Exception e)
Exception MessagingException
    s
    e
See also
    getNextException, getNextException, getCause

MessagingException
public MessagingException(String s, Exception e)

Constructs a MessagingException with the specified Exception and detail message. The specified exception is chained to this exception.

Parameters:
s - the detail message
e - the embedded exception

See Also:
getNextException(), setNextException(java.lang.Exception), getCause()

Method Detail

public Exception getNextException()

public Throwable getCause()
public Throwable getCause()

Overrides the getCause method of Throwable to return the next exception in the chain of nested exceptions.

Overrides: 

gCause in class Throwable

Returns: 

next Exception, null if none.

public boolean setNextException(Exception ex)

MessagingException

ex

return Exception true false

setNextException

public boolean setNextException(Exception ex)

Add an exception to the end of the chain. If the end is not a MessagingException, this exception cannot be added to the end.

Parameters:

ex - the new end of the Exception chain

Returns: 

true if this Exception was added, false otherwise.

public String toString()

toString

public String toString()
Override toString method to provide information on nested exceptions.

Overrides:

`toString` in class `Throwable`
javax.faces.el Class MethodBinding

java.lang.Object
  └─javax.faces.el.MethodBinding

**Deprecated.** This has been replaced by MethodExpression.

<table>
<thead>
<tr>
<th>MethodBinding</th>
<th>ValueBinding</th>
<th>Web</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.faces.application.Application</td>
<td>createMethodBinding()</td>
<td>MethodBinding</td>
</tr>
</tbody>
</table>

| deprecated | javax.el.MethodExpression |

public abstract class MethodBinding
extends Object

MethodBinding is an object that can be used to call an arbitrary public method, on an instance that is acquired by evaluatng the leading portion of a method binding expression via a ValueBinding. An immutable MethodBinding for a particular method binding expression can be acquired by calling the createMethodBinding() method of the Application instance for this web application.

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>MethodBinding()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>String</th>
<th>getExpressionString()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated.</td>
<td>Return the (possibly null) expression String, with leading and trailing delimiters, from which this MethodBinding</td>
</tr>
</tbody>
</table>
### Constructor Detail

**public MethodBinding()**

### MethodBinding

**public MethodBinding()**

**Deprecated.**

### Method Detail

```java
abstract public Object invoke(FacesContext context, Object[] params) throws EvaluationException, MethodNotFoundException
```

*FacesContext*
**invoke**

public abstract Object invoke(FacesContext context, Object[] params) throws EvaluationException, MethodNotFoundException, NullPointerException

Deprecated.

Return the return value (if any) resulting from a call to the method identified by this method binding expression, passing it the specified parameters, relative to the specified FacesContext.

**Parameters:**
- context - FacesContext for the current request
- params - Array of parameters to be passed to the called method, or null for no parameters

**Throws:**
- EvaluationException - if an exception is thrown by the called method (the thrown exception must be included as the cause property of this exception)
- MethodNotFoundException - if no suitable method can be found
- NullPointerException - if context is null

---

abstract public Class<T> getType(FacesContext context) throws MethodNotFoundException

Java

context - FacesContext

Throws - MethodNotFoundException:
Throws
NullPointerException: context null

**getExpressionString**

```java
public String getExpressionString()
```

**Deprecated.**

Return the (possibly null) expression String, with leading and trailing delimiters, from which this MethodBinding was built. The default implementation returns null.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS    NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD |

FRAMES    NO FRAMES
DETAIL: FIELD | CONSTR | METHOD |
javax.el **Class MethodExpression**

`java.lang.Object`  
  `javax.el.Expression`  
  `javax.el.MethodExpression`

All Implemented Interfaces:  
`Serializable`

```java
public abstract class MethodExpression extends Expression

Extends: Expression

Expression

ExpressionFactory#createMethodExpression MethodExpression
  FunctionMapper ELContext

#MethodInfo #invoke ELContext ELResolver .
[] ELResolver#getValue base base null
PropertyNotFoundException property String
MethodExpression

Expression javadoc
since JSP 2.1
See `javax.el.ELResolver, javax.el.Expression`
also `javax.el.ExpressionFactory`
```

An Expression that refers to a method on an object.

The `ExpressionFactory#createMethodExpression(javax.el.ELContext, java.lang.String, java.lang.Class, java.lang.Class[])` method can be used to parse an expression string and return a concrete instance of `MethodExpression` that encapsulates the parsed expression. The
FunctionMapper is used at parse time, not evaluation time, so one is not needed to evaluate an expression using this class. However, the ELContext is needed at evaluation time.

The getMethodInfo(javax.el.ELContext) and invoke(javax.el.ELContext, java.lang.Object[]) methods will evaluate the expression each time they are called. The ELResolver in the ELContext is used to resolve the top-level variables and to determine the behavior of the . and [] operators. For any of the two methods, the ELResolver.getValue(javax.el.ELContext, java.lang.Object, java.lang.Object) method is used to resolve all properties up to but excluding the last one. This provides the base object on which the method appears. If the base object is null, a PropertyNotFoundException must be thrown. At the last resolution, the final property is then coerced to a String, which provides the name of the method to be found. A method matching the name and expected parameters provided at parse time is found and it is either queried or invoked (depending on the method called on this MethodExpression).

See the notes about comparison, serialization and immutability in the Expression javadocs.

Since:
JSP 2.1

See Also:
ELResolver, Expression, ExpressionFactory, Serialized Form

---

**Constructor Summary**

| MethodExpression() |

**Method Summary**

<table>
<thead>
<tr>
<th>abstract MethodInfo</th>
<th>getMethodInfo(ELContext context)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Evaluates the expression relative to the provided context, and returns information about the actual referenced method.</td>
</tr>
</tbody>
</table>
abstract public MethodInfo getMethodInfo(ELContext context)

    context
    return MethodInfo

    Throws NullPointerException: context
    Throws PropertyNotFoundException:
    Throws MethodNotFoundException:
    Throws ELException: cause
getMethodInfo

public abstract MethodInfo getMethodInfo(ELContext context)

Evaluates the expression relative to the provided context, and returns information about the actual referenced method.

Parameters:
context - The context of this evaluation

Returns: 
an instance of MethodInfo containing information about the method the expression evaluated to.

Throws: 
NullPointerException - if context is null
PropertyNotFoundException - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.
MethodNotFoundException - if no suitable method can be found.
ELException - if an exception was thrown while performing property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

abstract public Object invoke(ELContext context, Object[] params)
String String
String expectedReturnType ELException EL
“1.18 ” String
true

context
params null
return void null
Throws NullPointerException: context null
Throws PropertyNotFoundException:
Throws MethodNotFoundException:
ELException: String MethodExpression
Throws expectedReturnType void String expectedReturnType
“1.18 ”
invoke

public abstract Object invoke(ELContext context, Object[] params)

If a String literal is specified as the expression, returns the String literal coerced to the expected return type of the method signature. An ELException is thrown if expectedReturnType is void or if the coercion of the String literal to the expectedReturnType yields an error (see Section "1.18 Type Conversion" of the EL specification). If not a String literal, evaluates the expression relative to the provided context, invokes the method that was found using the supplied parameters, and returns the result of the method invocation. Any parameters passed to this method is ignored if isLiteralText() is true.

Parameters:
- context - The context of this evaluation.
- params - The parameters to pass to the method, or null if no parameters.

Returns:
the result of the method invocation (null if the method has a void return type).

throws:
- NullPointerException - if context is null
- PropertyNotFoundException - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.
- MethodNotFoundException - if no suitable method can be found.
- ELException - if a String literal is specified and expectedReturnType of the MethodExpression is void or if the coercion of the String literal to the expectedReturnType yields an error (see Section "1.18 Type Conversion").
- ELException - if an exception was thrown while performing property or variable resolution. The thrown exception must be included as the cause property of this exception, if available. If
the exception thrown is an InvocationTargetException, extract its cause and pass it to the ELException constructor.

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS NEXT CLASS</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td>FRAMES NO FRAMES</td>
</tr>
</tbody>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.event Class

MethodExpressionActionListener

java.lang.Object
  ↓ javax.faces.event.MethodExpressionActionListener

All Implemented Interfaces:
   EventListener, StateHolder, ActionListener, FacesListener

public class MethodExpressionActionListener
  extends Object
  implements ActionListener, StateHolder

Implements: ActionListener, StateHolder

MethodExpressionActionListener MethodExpression
ActionListener ActionEvent MethodExpression

MethodExpressionActionListener is an ActionListener that wraps a
MethodExpression. When it receives a ActionEvent, it executes a method
on an object identified by the MethodExpression.

Constructor Summary

<table>
<thead>
<tr>
<th>MethodExpressionActionListener()</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodExpressionActionListener(MethodExpression methodExpression)</td>
</tr>
</tbody>
</table>

Construct a ValueChangeListener that contains a
MethodExpression.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isTransient()</td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void processAction(ActionEvent actionEvent)</td>
<td>Invoked when the action described by the specified ActionEvent occurs.</td>
</tr>
<tr>
<td>void restoreState(FacesContext context, Object state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object saveState(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void setTransient(boolean newTransientValue)</td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

close, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

public MethodExpressionActionListener()

**MethodExpressionActionListener**

public MethodExpressionActionListener()

public MethodExpressionActionListener(MethodExpression methodExpression)

MethodExpression ValueChangeListener
MethodExpressionActionListener

public MethodExpressionActionListener(MethodExpression methodExpression)

Construct a ValueChangeListener that contains a MethodExpression.

Method Detail

public void processAction(ActionEvent actionEvent)
throws AbortProcessingException

Throws  
NullPointerException: NullPointerException
AbortProcessingException: NullPointerException  
JavaServer Face

processAction

public void processAction(ActionEvent actionEvent)
throws AbortProcessingException

Description copied from interface: ActionListener

Invoked when the action described by the specified ActionEvent occurs.

Specified by: processAction in interface ActionListener

Parameters:
actionEvent - The ActionEvent that has occurred

Throws:
NullPointeException
AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed
public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**

saveState in interface StateHolder

public void restoreState(FacesContext context, Object state)

restoreState

public void restoreState(FacesContext context, Object state)
Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**

`restoreState` in interface **StateHolder**

---

**public boolean isTransient()**

**isTransient**

**public boolean isTransient()**

Description copied from interface: **StateHolder**

If true, the Object implementing this interface must not participate in state saving or restoring.

**Specified by:**

`isTransient` in interface **StateHolder**

---

**public void setTransient(boolean newTransientValue)**

**setTransient**

**public void setTransient(boolean newTransientValue)**

Description copied from interface: **StateHolder**
Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Specified by:**

`setTransient` in interface `StateHolder`

**Parameters:**

`newTransientValue` - boolean pass `true` if this Object will participate in state saving or restoring, otherwise pass `false`.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.validator Class MethodExpressionValidator

java.lang.Object  
   javax.faces.validator.MethodExpressionValidator

All Implemented Interfaces:
   java.util.EventListener, javax.faces.validator.StateHolder, javax.faces.validator.Validator

public class MethodExpressionValidator
   extends java.lang.Object
   implements javax.faces.validator.Validator, javax.faces.validator.StateHolder

Implements: javax.faces.validator.Validator, javax.faces.validator.StateHolder

MethodExpressionValidator is a javax.faces.validator.Validator that wraps a javax.faces.validator.MethodExpression, and it performs validation by executing a method on an object identified by the javax.faces.validator.MethodExpression.

Field Summary

 Fields inherited from interface javax.faces.validator.Validator
 NOT_IN_RANGE_MESSAGE_ID

Constructor Summary

 MethodExpressionValidator()

 MethodExpressionValidator(javax.faces.validator.MethodExpression methodExpression)
Construct a Validator that contains a MethodExpression.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>isTransient()</td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void</td>
<td>restoreState(FacesContext context, Object state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td>saveState(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td>setTransient(boolean transientValue)</td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void</td>
<td>validate(FacesContext context, UIComponent component, Object value)</td>
<td>Perform the correctness checks implemented by this Validator against the specified UIComponent.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public MethodExpressionValidator()
```

### MethodExpressionValidator

```java
public MethodExpressionValidator()
```
public MethodExpressionValidator
(MethodExpression methodExpression)

MethodExpressionValidator

public MethodExpressionValidator
(MethodExpression methodExpression)

Construct a Validator that contains a MethodExpression.

Method Detail

public void validate
(FacesContext context, UIComponent component, Object value)
throws ValidatorException

Throws
NullPointerException: NullPointe
context
component null

Throws
ValidatorException: NullPointerException

validate

public void validate
(FacesContext context, UIComponent component, Object value)
throws ValidatorException

Description copied from interface: Validator

Perform the correctness checks implemented by this Validator against the specified UIComponent. If any violations are found, a ValidatorException will be thrown containing the FacesMessage describing the failure.

Specified by:
validate in interface Validator
Parameters:
- `context` - FacesContext for the request we are processing
- `component` - UIComponent we are checking for correctness
- `value` - the value to validate

Throws:
- `NullPointerException` - if `context` or `component` is null
- `ValidatorException` - if validation fails

public `Object` `saveState(FacesContext` `context)`

Description copied from interface: `StateHolder`

Gets the state of the instance as a `Serializable` Object.

If the class that implements this interface has references to instances that implement `StateHolder` (such as a `UIComponent` with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

**Specified by:**
- `saveState` in interface `StateHolder`
public void restoreState(FacesContext context, Object state)

**restoreState**

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface:** StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

**Specified by:**

```
restoreState in interface StateHolder
```

---

public boolean isTransient()

**isTransient**

```java
public boolean isTransient()
```

**Description copied from interface:** StateHolder

If true, the Object implementing this interface must not participate in state saving or restoring.

**Specified by:**

```
isTransient in interface StateHolder
```
public void setTransient(boolean transientValue)

setTransient

public void setTransient(boolean transientValue)

Description copied from interface: StateHolder

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

Specified by:

setTransient in interface StateHolder

Parameters:

transientValue - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.
javax.faces.event Class
MethodExpressionValueChangeListener

java.lang.Object
   | javax.faces.event.MethodExpressionValueChangeListener

All Implemented Interfaces:
   com.sun.faces.event.ActionListener,
   javax.faces.event.ValueChangeListener

public class MethodExpressionValueChangeListener
   extends Object
   implements ValueChangeListener, StateHolder

Implements: ValueChangeListener, StateHolder

MethodExpressionValueChangeListener is a ValueChangeListener that wraps a MethodExpression. When it receives a ValueChangeEvent, it executes a method on an object identified by the MethodExpression.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodExpressionValueChangeListener()</td>
</tr>
<tr>
<td>MethodExpressionValueChangeListener(MethodExpression methodExpression)</td>
</tr>
</tbody>
</table>

   Construct a ValueChangeListener that contains a MethodExpression.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>isTransient()</td>
</tr>
</tbody>
</table>

| boolean | If true, the Object implementing this interface must not participate in state saving or restoring. |
|-----------------------------------------------|
| void processValueChange(ValueChangeEvent valueChangeEvent) | Invoked when the value change described by the specified ValueChangeEvent occurs. |
| void restoreState(FacesContext context, Object state) | Perform any processing required to restore the state from the entries in the state Object. |
| Object saveState(FacesContext context) | Gets the state of the instance as a Serializable Object. |
| void setTransient(boolean newTransientValue) | Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring. |

Methods inherited from class java.lang.Object:
call, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Constructor Detail

**public MethodExpressionValueChangeListener()**

```
MethodExpressionValueChangeListener
```

**public MethodExpressionValueChangeListener()**

```
MethodExpressionValueChangeListener
```

**public MethodExpressionValueChangeListener(MethodExpression methodExpression)**

```
MethodExpression ValueChangeListener
```
MethodExpressionValueChangeListener

public MethodExpressionValueChangeListener(MethodExpression methodExpression)

Construct a ValueChangeListener that contains a MethodExpression.

Method Detail

public void processValueChange(ValueChangeEvent valueChangeEvent) throws AbortProcessingException

Throws

NullPointerException: NullPointerException

Throws

AbortProcessingException: NullPointerException JavaServer Face

processValueChange

public void processValueChange(ValueChangeEvent valueChangeEvent) throws AbortProcessingException

Description copied from interface: ValueChangeListener

Invoked when the value change described by the specified ValueChangeEvent occurs.

Specified by: ValueChangeListener

Parameters:

valueChangeEvent - The ValueChangeEvent that has occurred

Throws:

NullPointerException

AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed
public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

Specified by:

saveState in interface StateHolder

public void restoreState(FacesContext context, Object state)

restoreState

public void restoreState(FacesContext context, Object state)
Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: 
restoreState in interface StateHolder

---

public boolean isTransient()

isTransient

public boolean isTransient()  

Description copied from interface: StateHolder

If true, the Object implementing this interface must not participate in state saving or restoring.

Specified by:  
isTransient in interface StateHolder

---

public void setTransient(boolean newTransientValue)

setTransient

public void setTransient(boolean newTransientValue)

Description copied from interface: StateHolder
Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Specified by:**
setTransient in interface StateHolder

**Parameters:**
newTransientValue - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.el   **Class MethodInfo**

java.lang.Object  
   - javax.el.MethodInfo

public class **MethodInfo**
extends **Object**

```java

MethodExpression
since JSP 2.1
```

Holds information about a method that a **MethodExpression** evaluated to.

**Since:**
JSP 2.1

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MethodInfo</strong>(String name, Class&lt;?&gt; returnType, Class&lt;?&gt;[] paramTypes)</td>
<td></td>
</tr>
</tbody>
</table>
| Creates a new instance of **MethodInfo** with the given information.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getName</strong>()</td>
<td>Returns the name of the method</td>
</tr>
<tr>
<td><strong>getParamTypes</strong>()</td>
<td>Returns the parameter types of the method</td>
</tr>
<tr>
<td><strong>getReturnType</strong>()</td>
<td>Returns the return type of the method</td>
</tr>
</tbody>
</table>

---

Methods inherited from class java.lang.**Object**
Constructor Detail

MethodInfo

public MethodInfo(String name,
                   Class<? extends ?> returnType,
                   Class<? extends []> paramTypes)

Creates a new instance of MethodInfo with the given information.

Parameters:
- name - The name of the method
- returnType - The return type of the method
- paramTypes - The types of each of the method's parameters

Method Detail

public String getName()

    return

dgetName

public String getName()

Returns the name of the method

Returns:
the name of the method
public Class<T> getReturnType()
    
    return

getReturnType

public Class<?> getReturnType()

    Returns the return type of the method

    Returns:
        the return type of the method

public Class<T>[] getParamTypes()

    return

getParamTypes

public Class<?>[] getParamTypes()

    Returns the parameter types of the method

    Returns:
        the parameter types of the method
PS:
javax.el  Class MethodNotFoundException

    java.lang.Object
      ├ java.lang.Throwable
      │   └ java.lang.Exception
      │     └ java.lang.RuntimeException
      │        └ javax.el.ELException
      │              └ javax.el.MethodNotFoundException

All Implemented Interfaces:
    Serializable

public class MethodNotFoundException
    extends ELException

Extends: Throwable > Exception > RuntimeException > ELException

    MethodExpression

       since JSP 2.1

       See also  javax.el.MethodExpression

Thrown when a method could not be found while evaluating a MethodExpression.

Since:
    JSP 2.1

See Also:
    MethodExpression, Serialized Form

---

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodNotFoundException()</td>
</tr>
<tr>
<td>Creates a MethodNotFoundException with no detail message.</td>
</tr>
<tr>
<td>MethodNotFoundException(String message)</td>
</tr>
<tr>
<td>Creates a MethodNotFoundException with the provided detail</td>
</tr>
</tbody>
</table>
MethodNotFoundException(String pMessage, Throwable pRootCause)

Creates a MethodNotFoundException with the given detail message and root cause.

MethodNotFoundException(Throwable exception)

Creates a MethodNotFoundException with the given root cause.

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MethodNotFoundException()

MethodNotFoundException

public MethodNotFoundException()

Creates a MethodNotFoundException with no detail message.

public MethodNotFoundException(String message)

MethodNotFoundException

message
MethodNotFoundException

public MethodNotFoundException(String message)

    Creates a MethodNotFoundException with the provided detail message.

    Parameters:
    message - the detail message

public MethodNotFoundException(Throwable exception)

    MethodNotFoundException
    exception

MethodNotFoundException

public MethodNotFoundException(Throwable exception)

    Creates a MethodNotFoundException with the given root cause.

    Parameters:
    exception - the originating cause of this exception

public MethodNotFoundException(String pMessage, Throwable pRootCause)

    MethodNotFoundException
    pMessage
    pRootCause

MethodNotFoundException

public MethodNotFoundException(String pMessage,
(Throwable pRootCause)

Creates a MethodNotFoundException with the given detail message and root cause.

**Parameters:**

- `pMessage` - the detail message
- `pRootCause` - the originating cause of this exception
javax.faces.el Class MethodNotFoundException

java.lang.Object
    └ java.lang.Throwable
        └ java.lang.Exception
            └ java.lang.RuntimeException
                └ javax.faces.FacesException
                    └ javax.faces.el.EvaluationException
                        └ javax.faces.el.MethodNotFoundException

All Implemented Interfaces:
    Serializable

------------------------------------------
Deprecated. This has been replaced by MethodNotFoundException.

Extends: Throwable > Exception > RuntimeException > FacesException
          > EvaluationException

base

    deprecated javax.el.MethodNotFoundException

public class MethodNotFoundException

extends EvaluationException

An exception caused by a method name that cannot be resolved against a base object.

See Also:
    Serialized Form

------------------------------------------

Constructor Summary

MethodNotFoundException()

    Deprecated. Construct a new exception with no detail message
or root cause.

<table>
<thead>
<tr>
<th>MethodNotFoundException(String message)</th>
<th>Deprecated. Construct a new exception with the specified detail message and no root cause.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodNotFoundException(String message, Throwable cause)</td>
<td>Deprecated. Construct a new exception with the specified detail message and root cause.</td>
</tr>
<tr>
<td>MethodNotFoundException(Throwables cause)</td>
<td>Deprecated. Construct a new exception with the specified root cause.</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class javax.faces.FacesException
getCause

Methods inherited from class java.lang Throwable
fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MethodNotFoundException()
MethodNotFoundException

public MethodNotFoundException()

    Deprecated.

    Construct a new exception with no detail message or root cause.

public MethodNotFoundException(String message)

    message

MethodNotFoundException

public MethodNotFoundException(String message)

    Deprecated.

    Construct a new exception with the specified detail message and no root cause.

    Parameters:
    message - The detail message for this exception

public MethodNotFoundException(Throwables cause)

    (cause == null ? null : cause.toString())
    cause

MethodNotFoundException
public MethodNotFoundException(Throwable cause)

Deprecated.

Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

Parameters:
cause - The root cause for this exception

public MethodNotFoundException(String message, Throwable cause)

message
cause

MethodNotFoundException

public MethodNotFoundException(String message, Throwable cause)

Deprecated.

Construct a new exception with the specified detail message and root cause.

Parameters:
message - The detail message for this exception
cause - The root cause for this exception
javax.mail  Class MethodNotSupportedException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.mail.MessagingException
            ↓ javax.mail.MethodNotSupportedException

All Implemented Interfaces:
   Serializable

public class MethodNotSupportedException
   extends MessagingException

Extends: Throwable > Exception > MessagingException

The exception thrown when a method is not supported by the implementation

Author:  
   John Mani
See Also:  
   Serialized Form

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodNotSupportedException()</td>
</tr>
<tr>
<td>Constructs a MethodNotSupportedException with no detail message.</td>
</tr>
<tr>
<td>MethodNotSupportedException(String s)</td>
</tr>
<tr>
<td>Constructs a MethodNotSupportedException with the specified detail message.</td>
</tr>
</tbody>
</table>
## Method Summary

### Methods inherited from class javax.mail.

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessagingException</td>
</tr>
<tr>
<td>getCause, getNextException, setNextException, toString</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable</td>
</tr>
<tr>
<td>fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
</tr>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

## Constructor Detail

```java
class MethodNotSupportedException
```

### public MethodNotSupportedException()

MethodNotSupportedException

### Constructor Method

#### public MethodNotSupportedException()

Constructs a MethodNotSupportedException with no detail message.

#### public MethodNotSupportedException(String s)

```
MethodNotSupportedException s
```

MethodNotSupportedException
public MethodNotSupportedException(String s)

Constructs a MethodNotSupportedException with the specified detail message.

Parameters:
   s - the detail message
javax.mail.internet Class MimeBodyPart

java.lang.Object
   ▼ javax.mail.BodyPart
      ▼ javax.mail.internet.MimeBodyPart

All Implemented Interfaces:
   MimePart, Part

Direct Known Subclasses:
   PreencodedMimeBodyPart

public class MimeBodyPart

extends BodyPart
implements MimePart

Extends: BodyPart
Implements: MimePart
Extended by: PreencodedMimeBodyPart

MIME          BodyPart   MimePart   MimeBodyPart
MimeMultipart

MimeBodyPart   InternetHeaders

RFC 822 MIME

RFC 822     US-ASCII    ASCII MIME RFC 2047
MimeUtility  setHeader   addHeader   addHeaderLine MIME
SMTP 1000

See also
   javax.mail.Part, javax.mail.internet.MimePart,
   javax.mail.internet.MimeUtility

This class represents a MIME body part. It implements the BodyPart
abstract class and the MimePart interface. MimeBodyParts are contained in MimeMultipart objects.

MimeBodyPart uses the InternetHeaders class to parse and store the headers of that body part.

A note on RFC 822 and MIME headers

RFC 822 header fields must contain only US-ASCII characters. MIME allows non ASCII characters to be present in certain portions of certain headers, by encoding those characters. RFC 2047 specifies the rules for doing this. The MimeUtility class provided in this package can be used to achieve this. Callers of the setHeader, addHeader, and addHeaderLine methods are responsible for enforcing the MIME requirements for the specified headers. In addition, these header fields must be folded (wrapped) before being sent if they exceed the line length limitation for the transport (1000 bytes for SMTP). Received headers may have been folded. The application is responsible for folding and unfolding headers as appropriate.

Author:
John Mani, Bill Shannon, Kanwar Oberoi

See Also:
Part, MimePart, MimeUtility

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected byte[] content</td>
<td>Byte array that holds the bytes of the content of this Part.</td>
</tr>
<tr>
<td>protected InputStream contentStream</td>
<td>If the data for this body part was supplied by an InputStream that implements the SharedInputStream interface, contentStream is another such stream representing the content of this body part.</td>
</tr>
<tr>
<td>protected DataHandler dh</td>
<td>The DataHandler object representing this Part's</td>
</tr>
</tbody>
</table>
protected InternetHeaders headers

The InternetHeaders object that stores all the headers of this body part.

Fields inherited from class javax.mail.BodyPart

parent

Fields inherited from interface javax.mail.Part

ATTACHMENT, INLINE

Constructor Summary

MimeBodyPart()

An empty MimeBodyPart object is created.

MimeBodyPart(InputStream is)

Constructs a MimeBodyPart by reading and parsing the data from the specified input stream.

MimeBodyPart(InternetHeaders headers, byte[] content)

Constructs a MimeBodyPart using the given header and content bytes.

Method Summary

void addHeader(String name, String value)

Add this value to the existing values for this header_name.

void addHeaderLine(String line)

Add a header line to this body part

void attachFile(File file)

Use the specified file to provide the data for this part.

void attachFile(String file)

Use the specified file to provide the data for this part.

Enumeration getAllHeaderLines()

Get all header lines as an Enumeration of Strings.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Enumeration getAllHeaders()</code></td>
<td>Return all the headers from this Message as an Enumeration of Header objects.</td>
</tr>
<tr>
<td><code>Object getContent()</code></td>
<td>Return the content as a Java object.</td>
</tr>
<tr>
<td><code>String getContentID()</code></td>
<td>Returns the value of the &quot;Content-ID&quot; header field.</td>
</tr>
<tr>
<td><code>String[] getContentLanguage()</code></td>
<td>Get the languages specified in the Content-Language header of this MimePart.</td>
</tr>
<tr>
<td><code>String getContentMD5()</code></td>
<td>Return the value of the &quot;Content-MD5&quot; header field.</td>
</tr>
<tr>
<td><code>protected InputStream getContentStream()</code></td>
<td>Produce the raw bytes of the content.</td>
</tr>
<tr>
<td><code>String getContentType()</code></td>
<td>Returns the value of the RFC 822 &quot;Content-Type&quot; header field.</td>
</tr>
<tr>
<td><code>DataHandler getDataHandler()</code></td>
<td>Return a DataHandler for this body part's content.</td>
</tr>
<tr>
<td><code>String getDescription()</code></td>
<td>Returns the &quot;Content-Description&quot; header field of this body part.</td>
</tr>
<tr>
<td><code>String getDisposition()</code></td>
<td>Returns the value of the &quot;Content-Disposition&quot; header field.</td>
</tr>
<tr>
<td><code>String getEncoding()</code></td>
<td>Returns the content transfer encoding from the &quot;Content-Transfer-Encoding&quot; header field.</td>
</tr>
<tr>
<td><code>String getFileName()</code></td>
<td>Get the filename associated with this body part.</td>
</tr>
<tr>
<td><code>String[] getHeader(String name)</code></td>
<td>Get all the headers for this header_name.</td>
</tr>
<tr>
<td><code>String[] getHeader(String name, String delimiter)</code></td>
<td>Get all the headers for this header name, returned as a single String, with headers separated by the delimiter.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>InputStream getInputStream()</code></td>
<td>Return a decoded input stream for this body part's &quot;content&quot;.</td>
</tr>
<tr>
<td><code>int getLineCount()</code></td>
<td>Return the number of lines for the content of this Part.</td>
</tr>
<tr>
<td><code>Enumeration getMatchingHeaderLines(String[] names)</code></td>
<td>Get matching header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><code>Enumeration getMatchingHeaders(String[] names)</code></td>
<td>Return matching headers from this Message as an Enumeration of Header objects.</td>
</tr>
<tr>
<td><code>Enumeration getNonMatchingHeaderLines(String[] names)</code></td>
<td>Get non-matching header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><code>Enumeration getNonMatchingHeaders(String[] names)</code></td>
<td>Return non-matching headers from this Message as an Enumeration of Header objects.</td>
</tr>
<tr>
<td><code>InputStream getRawInputStream()</code></td>
<td>Return an InputStream to the raw data with any Content-Transfer-Encoding intact.</td>
</tr>
<tr>
<td><code>int getSize()</code></td>
<td>Return the size of the content of this body part in bytes.</td>
</tr>
<tr>
<td><code>boolean isMimeType(String mimeType)</code></td>
<td>Is this Part of the specified MIME type?</td>
</tr>
<tr>
<td><code>void removeHeader(String name)</code></td>
<td>Remove all headers with this name.</td>
</tr>
<tr>
<td><code>void saveFile(File file)</code></td>
<td>Save the contents of this part in the specified file.</td>
</tr>
<tr>
<td><code>void saveFile(String file)</code></td>
<td>Save the contents of this part in the specified file.</td>
</tr>
<tr>
<td><code>void setContent(Multipart mp)</code></td>
<td>This method sets the body part's content to a Multipart object.</td>
</tr>
<tr>
<td><code>void setContent(Object o, String type)</code></td>
<td>A convenience method for setting this body part's</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>setContentID(String cid)</code></td>
<td>Set the &quot;Content-ID&quot; header field of this body part.</td>
</tr>
<tr>
<td><code>setContentLanguage(String[] languages)</code></td>
<td>Set the Content-Language header of this MimePart.</td>
</tr>
<tr>
<td><code>setContentMD5(String md5)</code></td>
<td>Set the &quot;Content-MD5&quot; header field of this body part.</td>
</tr>
<tr>
<td><code>setDataHandler(DataHandler dh)</code></td>
<td>This method provides the mechanism to set this body part's content.</td>
</tr>
<tr>
<td><code>setDescription(String description)</code></td>
<td>Set the &quot;Content-Description&quot; header field for this body part.</td>
</tr>
<tr>
<td><code>setDescription(String description, String charset)</code></td>
<td>Set the &quot;Content-Description&quot; header field for this body part.</td>
</tr>
<tr>
<td><code>setDisposition(String disposition)</code></td>
<td>Set the &quot;Content-Disposition&quot; header field of this body part.</td>
</tr>
<tr>
<td><code>setFileName(String filename)</code></td>
<td>Set the filename associated with this body part, if possible.</td>
</tr>
<tr>
<td><code>setHeader(String name, String value)</code></td>
<td>Set the value for this header name.</td>
</tr>
<tr>
<td><code>setText(String text)</code></td>
<td>Convenience method that sets the given String as this part's content, with a MIME type of &quot;text/plain&quot;.</td>
</tr>
<tr>
<td><code>setText(String text, String charset)</code></td>
<td>Convenience method that sets the given String as this part's content, with a MIME type of &quot;text/plain&quot; and the specified charset.</td>
</tr>
<tr>
<td><code>setText(String text, String charset, String subtype)</code></td>
<td>Convenience method that sets the given String as this part's content, with a primary MIME type of &quot;text&quot; and the specified MIME subtype.</td>
</tr>
</tbody>
</table>
protected void updateHeaders()

Examine the content of this body part and update the appropriate MIME headers.

void writeTo(OutputStream os)

Output the body part as an RFC 822 format stream.

Methods inherited from class javax.mail.BodyPart
getParent

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

dh

protected DataHandler dh

The DataHandler object representing this Part's content.

content

protected byte[] content

Byte array that holds the bytes of the content of this Part.

contentStream
protected InputStream contentStream

If the data for this body part was supplied by an InputStream that implements the SharedInputStream interface, contentStream is another such stream representing the content of this body part. In this case, content will be null.

Since:
JavaMail 1.2

headers

protected InternetHeaders headers

The InternetHeaders object that stores all the headers of this body part.

Constructor Detail

public MimeBodyPart()

An empty MimeBodyPart object is created. This body part maybe filled in by a client constructing a multipart message.

public MimeBodyPart(java.io.InputStream is) throws MessagingException

MimeBodyPart MIME
MimeBodyPart

public MimeBodyPart(InputStream is) throws MessagingException

Constructs a MimeBodyPart by reading and parsing the data from the specified input stream. The parser consumes data till the end of the given input stream. The input stream must start at the beginning of a valid MIME body part and must terminate at the end of that body part.

Note that the "boundary" string that delimits body parts must not be included in the input stream. The intention is that the MimeMultipart parser will extract each body part's bytes from a multipart stream and feed them into this constructor, without the delimiter strings.

Parameters:

is - the body part Input Stream

Throws:

MessagingException

public MimeBodyPart(InternetHeaders headers, byte[] content) throws MessagingException

---

MimeBodyPart
public MimeBodyPart(InternetHeaders headers, byte[] content)
throws MessagingException

Constructs a MimeBodyPart using the given header and content bytes.

Used by providers.

**Parameters:**
- headers - The header of this part
- content - bytes representing the body of this part.

**Throws:**
- MessagingException

### Method Detail

**public int getSize() throws MessagingException**
-1

```
content null      contentStream null      available
-1
return -1
```

**getSize**

public int getSize()
throws MessagingException

Return the size of the content of this body part in bytes. Return -1 if the size cannot be determined.

Note that this number may not be an exact measure of the content size and may or may not account for any transfer encoding of the
content.

This implementation returns the size of the content array (if not null),
or, if contentStream is not null, and the available method returns a
positive number, it returns that number as the size. Otherwise, it
returns -1.

Specified by:
getSize in interface Part

Returns:
size in bytes, or -1 if not known

Throws:
MessagingException

public int getLineCount() throws MessagingException
Part -1

-1
    return -1

getLineCount

public int getLineCount()
    throws MessagingException

Return the number of lines for the content of this Part. Return -1 if
this number cannot be determined.

Note that this number may not be an exact measure of the content
length and may or may not account for any transfer encoding of the
content.

This implementation returns -1.
public String getContentType() throws MessagingException

RFC 822 "Content-Type" null
"text/plain"

    getHeader(name)
    return Content-Type

getContentType

public String getContentType() throws MessagingException

Returns the value of the RFC 822 "Content-Type" header field. This represents the content type of this body part. This value must not be null. If this field is unavailable, "text/plain" should be returned.

This implementation uses getHeader(name) to obtain the requisite header field.

Specified by: getContentType in interface Part

Returns: Content-Type of this body part

Throws: MessagingException

See Also: DataHandler
public boolean isMimeType(String mimeType) throws MessagingException

Part MIME primaryType subType

"text/plain" "text/plain; charset=foobar"

true

mimeType subType '*' subtype

isMimeType

public boolean isMimeType(String mimeType) throws MessagingException

Is this Part of the specified MIME type? This method compares only the primaryType and subType. The parameters of the content types are ignored.

For example, this method will return true when comparing a Part of content type "text/plain" with "text/plain; charset=foobar".

If the subType of mimeType is the special character '*', then the subtype is ignored during the comparison.

Specified by:

isMimeType in interface Part

Throws:

MessagingException

public String getDisposition() throws MessagingException
"Content-Disposition" disposition disposition

Content-Disposition null
getDisposition

public String getDisposition()
    throws MessagingException

    Returns the value of the "Content-Disposition" header field. This represents the disposition of this part. The disposition describes how the part should be presented to the user.

    If the Content-Disposition field is unavailable, null is returned.

    This implementation uses getHeader(name) to obtain the requisite header field.

Specify by:
    getDisposition in interface Part

Returns:
    disposition of this part, or null if unknown

Throws:
    MessagingException

See Also:
    headers

public void setDisposition(String disposition) throws MessagingException
"Content-Disposition" disposition null
"Content-Disposition"

    Throws IllegalArgumentException:
    Throws IllegalStateException: READ_ONLY
public void setDisposition(String disposition) throws MessagingException

Set the "Content-Disposition" header field of this body part. If the disposition is null, any existing "Content-Disposition" header field is removed.

Specified by:
   setDisposition in interface Part

Parameters:
   disposition - disposition of this part

Throws:
   IllegalWriteException - if the underlying implementation does not support modification
   IllegalStateException - if this body part is obtained from a READ_ONLY folder.
   MessagingException

See Also:
   Part.ATTACHMENT, Part.INLINE,
   Part.setFileName(java.lang.String)

public String getEncoding() throws MessagingException
"Content-Transfer-Encoding"

   getHeader(name)
   See also headers

getcoding

public String getEncoding() throws MessagingException

Returns the content transfer encoding from the "Content-Transfer-Encoding" header field. Returns null if the header is unavailable or its value is absent.
This implementation uses `getHeader(name)` to obtain the requisite header field.

**Specified by:**
- `getEncoding` in interface `MimePart`

**Returns:**
- `content-transfer-encoding`

**Throws:**
- `MessagingException`

**See Also:**
- `headers`

```java
public String getContentID() throws MessagingException {
    return null;
}
```

### getContentID

**public String getContentID() throws MessagingException**

Returns the value of the "Content-ID" header field. Returns `null` if the field is unavailable or its value is absent.

This implementation uses `getHeader(name)` to obtain the requisite header field.

**Specified by:**
- `getContentID` in interface `MimePart`

**Returns:**
- `content-ID`

**Throws:**
- `MessagingException`
public void setContentID(String cid) throws MessagingException

"Content-ID" cid null "Content-ID"

Throws IllegalWriteException:

Throws IllegalStateException: READ_ONLY

Throws MessagingException:

since JavaMail 1.3

setContentID

public void setContentID(String cid) throws MessagingException

Set the "Content-ID" header field of this body part. If the cid parameter is null, any existing "Content-ID" is removed.

Throws:

IllegalWriteException - if the underlying implementation does not support modification

IllegalStateException - if this body part is obtained from a READ_ONLY folder.

MessagingException

Since:

JavaMail 1.3

public String getContentMD5() throws MessagingException

"Content-MD5"

getHeader(name)

getContentMD5

public String getContentMD5() throws MessagingException

null
Return the value of the "Content-MD5" header field. Returns null if this field is unavailable or its value is absent.

This implementation uses getHeader(name) to obtain the requisite header field.

Specified by:
getContentMD5 in interface MimePart

Returns:
content-MD5

Throws:
MessagingException

public void setContentMD5(String md5) throws MessagingException
"Content-MD5"

Throws: IllegalWriteException:
Throws IllegalStateException: READ_ONLY

setContentMD5

public void setContentMD5(String md5)

throws MessagingException

Set the "Content-MD5" header field of this body part.

Specified by:
setContentMD5 in interface MimePart

Parameters:
md5 - the MD5 value

Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this body part is obtained from a READ_ONLY folder.
MessagingException
public String[] getContentLanguage() throws MessagingException

MimePart  Content-Language  Content-Language  RFC 1766  null

getHeader(name)

getContentLanguage

public String[] getContentLanguage() throws MessagingException

Get the languages specified in the Content-Language header of this MimePart. The Content-Language header is defined by RFC 1766. Returns null if this header is not available or its value is absent.

This implementation uses getHeader(name) to obtain the requisite header field.

Specified by:

getContentLanguage in interface MimePart

Throws:

MessagingException

public void setContentLanguage(String[] languages) throws MessagingException

MimePart  Content-Language  Content-Language  RFC 1766

languages

setContentLanguage

public void setContentLanguage(String[] languages) throws MessagingException
Set the Content-Language header of this MimePart. The Content-Language header is defined by RFC 1766.

**Specified by:**
`setContentLanguage` in interface `MimePart`

**Parameters:**
- `languages` - array of language tags

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification
- `MessagingException`

```java
public String getDescription() throws MessagingException
"Content-Description" null

Content-Description RFC 2047 Unicode

    getHeader(name)
    return

getDescription

public String getDescription() throws MessagingException

    Returns the "Content-Description" header field of this body part. This typically associates some descriptive information with this part. Returns null if this field is unavailable or its value is absent.

    If the Content-Description field is encoded as per RFC 2047, it is decoded and converted into Unicode. If the decoding or conversion fails, the raw data is returned as is.

    This implementation uses `getHeader(name)` to obtain the requisite header field.
```
public void setDescription(String description) throws MessagingException

"Content-Description" description

"Content-Description"

description US-ASCII description US-ASCII

MessagingException MessagingException
UnsupportedEncodingException

Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:
UnsupportedEncodingException

setDescription

public void setDescription(String description) throws MessagingException

Set the "Content-Description" header field for this body part. If the description parameter is null, then any existing "Content-Description" fields are removed.

If the description contains non US-ASCII characters, it will be encoded using the platform's default charset. If the description
contains only US-ASCII characters, no encoding is done and it is used as is.

Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.

**Specified by:**

`setDescription` in interface `Part`  

**Parameters:**

- `description` - content description

**Throws:**

- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this body part is obtained from a READ_ONLY folder.  
- `MessagingException` - otherwise; an UnsupportedEncodingException may be included in the exception chain if the charset conversion fails.

```java
class MyClass {
    public void setDescription(String description, String charset) throws MessagingException {
        "Content-Description" description "Content-Description"
        description US-ASCII description US-ASCII
        MessagingException MessagingException UnsupportedEncodingException

        description
        charset
        throws IllegalWriteException:
        Throws IllegalStateException: READ_ONLY
        MessagingException:
    }
}
```
setDescription

public void setDescription(String description, String charset)
    throws MessagingException

Set the "Content-Description" header field for this body part. If the description parameter is null, then any existing "Content-Description" fields are removed.

If the description contains non US-ASCII characters, it will be encoded using the specified charset. If the description contains only US-ASCII characters, no encoding is done and it is used as is.

Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.

Parameters:
    description - Description
    charset - Charset for encoding

 Throws:
    IllegalWriteException - if the underlying implementation does not support modification
    IllegalStateException - if this body part is obtained from a READ_ONLY folder.
    MessagingException - otherwise; an UnsupportedEncodingException may be included in the exception chain if the charset conversion fails.

public String getFileName() throws MessagingException
getFileName

```java
public String getFileName() throws MessagingException {
    Get the filename associated with this body part.

    Returns the value of the "filename" parameter from the "Content-Disposition" header field of this body part. If its not available, returns the value of the "name" parameter from the "Content-Type" header field of this body part. Returns null if both are absent.

    If the mail.mime.encodefilename System property is set to true, the MimeUtility.decodeText method will be used to decode the filename. While such encoding is not supported by the MIME spec, many mailers use this technique to support non-ASCII characters in filenames. The default value of this property is false.

    Specified by: getFileName in interface Part

    Returns: filename

    Throws: MessagingException
```

public void setFileName(String filename) throws MessagingException
"Content-Disposition" "filename" "Content-Type" "name"

mail.mime.encodefilename true
MimeUtility.encodeText MIME ASCII
false

Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY

setFileName

public void setFileName(String filename)
    throws MessagingException

Set the filename associated with this body part, if possible.

Sets the "filename" parameter of the "Content-Disposition" header field of this body part. For compatibility with older mailers, the "name" parameter of the "Content-Type" header is also set.

If the mail.mime.encodefilename System property is set to true, the MimeUtility.encodeText method will be used to encode the filename. While such encoding is not supported by the MIME spec, many mailers use this technique to support non-ASCII characters in filenames. The default value of this property is false.

Specified by:
    setFileName in interface Part

Parameters:
    filename - Filename to associate with this part

Throws:
    IllegalWriteException - if the underlying implementation does not support modification
    IllegalStateException - if this body part is obtained from a
public java.io.InputStream getInputStream() throws java.io.IOException, MessagingException "content"

DataHandler getDataHandler().getInputStream();

return InputStream

Throws MessagingException:

Throws java.io.IOException: DataHandler

javax.activation.DataHandler

See also getContentStream, getInputStream

getInputStream

public InputStream getInputStream()

throws IOException, MessagingException

Return a decoded input stream for this body part's "content".

This implementation obtains the input stream from the DataHandler. That is, it invokes getDataHandler().getInputStream();

Specified by:

getInputStream in interface Part

Returns:

an InputStream

Throws:

MessagingException

IOException - this is typically thrown by the DataHandler. Refer to the documentation for javax.activation.DataHandler for more details.

See Also:
**protected java.io.InputStream getContentStream() throws MessagingException**

DataHandler Part

See also content, getContentStream

---

**getContentStream**

protected InputStream getContentStream() throws MessagingException

Produce the raw bytes of the content. This method is used when creating a DataHandler object for the content. Subclasses that can provide a separate input stream for just the Part content might want to override this method.

**Throws:** MessagingException

**See Also:**
content, MimeMessage.getContentStream()
**getRawInputStream**

```java
public InputStream getRawInputStream() throws MessagingException
```

Return an InputStream to the raw data with any Content-Transfer-Encoding intact. This method is useful if the "Content-Transfer-Encoding" header is incorrect or corrupt, which would prevent the getInputStream method or getContent method from returning the correct data. In such a case the application may use this method and attempt to decode the raw data itself.

This implementation simply calls the getContentStream method.

**Throws:**
- MessagingException

**Since:**
JavaMail 1.2

**See Also:**
- getInputStream(), getContentStream()

---

**getDataHandler()**

```java
public DataHandler getDataHandler() throws MessagingException
```

Return a DataHandler for this body part's content.

**getDataHandler**

```java
public DataHandler getDataHandler() throws MessagingException
```

**See also**
- getDataHandler
The implementation provided here works just like the implementation in MimeMessage.

**Specified by:**
*getDataHandler* in interface *Part*

**Returns:**
DataHandler for the content

**Throws:**
*MessagingException*

**See Also:**
*MimeMessage.getDataHandler()*

```java
public Object getContent() throws java.io.IOException, MessagingException,
   java.activation.DataHandler

Java text/plain  String "multipart"
Multipart  DataHandler

DataHandler getDataHandler().getContent()
Multipart  Message
return

Throws *MessagingException;*

Throws java.io.IOException:  DataHandler
   javax.activation.DataHandler
```

**getContent**

```java
public Object getContent() throws IOException, MessagingException
```

Return the content as a Java object. The type of the object returned is of course dependent on the content itself. For example, the native format of a text/plain content is usually a String object. The native format for a "multipart" content is always a Multipart subclass. For content types that are unknown to the DataHandler system, an input
stream is returned as the content.

This implementation obtains the content from the DataHandler. That is, it invokes `getDataHandler().getContent();` If the content is a Multipart or Message object and was created by parsing a stream, the object is cached and returned in subsequent calls so that modifications to the content will not be lost.

**Specified by:**

`getContent` in interface `Part`

**Returns:**

Object

**Throws:**

`MessagingException`

`IOException` - this is typically thrown by the DataHandler. Refer to the documentation for `javax.activation.DataHandler` for more details.

**See Also:**

`DataHandler.getContent()`

---

**public void setDataHandler(DataHandler dh) throws MessagingException**

**DataHandler**

`dh` DataHandler

**Throws**

`IllegalWriteException`:

**Throws**

`IllegalStateException`: `READ_ONLY`

**setDataHandler**

**public void setDataHandler(DataHandler dh) throws MessagingException**

This method provides the mechanism to set this body part's content. The given DataHandler object should wrap the actual content.

**Specified by:**

`setDataHandler` in interface `Part`
Parameters:
   dh - The DataHandler for the content

Throws:
   IllegalArgumentException - if the underlying implementation does not support modification
   IllegalStateException - if this body part is obtained from a READ_ONLY folder.
   MessagingException

public void setContent(Object o, String type) throws MessagingException

DataHandler DataContentHandler JavaMail
setContent(foobar, "application/x-foobar")

DataContentHandler Java Activation Framework

"application/x-foobar" DataContentHandler Java

A convenience method for setting this body part's content.

The content is wrapped in a DataHandler object. Note that a DataContentHandler class for the specified type should be available to the JavaMail implementation for this to work right. That is, to do setContent(foobar, "application/x-foobar"), a DataContentHandler for "application/x-foobar" should be installed.
Refer to the Java Activation Framework for more information.

Specified by:

`setContent` in interface `Part`

Parameters:

- `o` - the content object
- `type` - Mime type of the object

Throws:

- `IllegalArgumentException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this body part is obtained from a `READ_ONLY` folder.
- `MessagingException`  

```java
public void setText(String text) throws MessagingException
MIME "text/plain" String US-ASCII
"charset"

text

charset setText

text

Throws

See also

setText(String text, String charset)
```

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain". If the string contains non US-ASCII characters, it will be encoded using the platform's default charset. The charset is also used to set the "charset" parameter.
Note that there may be a performance penalty if `text` is large, since this method may have to scan all the characters to determine what charset to use.

If the charset is already known, use the `setText` method that takes the charset parameter.

**Specified by:**
- `setText` in interface `MimePart`

**Specified by:**
- `setText` in interface `Part`

**Parameters:**
- `text` - the text content to set

**Throws:**
- `MessagingException` - if an error occurs

**See Also:**
- `setText(String text, String charset)`

```
public void setText(String text, String charset) throws MessagingException
MIME "text/plain" String Unicode
"charset"

    text
    charset

Throws                MessagingException:

setText

public void setText(String text, String charset)
    throws MessagingException

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain" and the specified charset. The given Unicode string will be charset-encoded using the specified charset. The charset is also used to set the "charset" parameter.
public void setText(String text, String charset, String subtype) throws MessagingException

MIME "text" MIME String Unicode "charset"

Convenience method that sets the given String as this part's content, with a primary MIME type of "text" and the specified MIME subtype. The given Unicode string will be charset-encoded using the specified charset. The charset is also used to set the "charset" parameter.

Specified by:
	setText in interface MimePart

Parameters:

text - the text content to set
charset - the charset to use for the text
subtype - the MIME subtype to use (e.g., "html")
public void setContent(Multipart mp) throws MessagingException

Multipart

mp Message multipart
Throws IllegalArgumentException:
Throws IllegalStateException: READ_ONLY

setContent

public void setContent(Multipart mp)
throws MessagingException

This method sets the body part's content to a Multipart object.

Specified by:
setContent in interface Part

Parameters:
mp - The multipart object that is the Message's content

Throws:
IllegalArgumentException - if the underlying implementation does not support modification of existing values.
IllegalStateException - if this body part is obtained from a READ_ONLY folder.
MessagingException

public void attachFile(java.io.File file) throws java.io.IOException, MessagingException

file File

Throws java.io.IOException:
public void attachFile(String file)
throws java.io.IOException, MessagingException

Use the specified file to provide the data for this part. The file name is used as the file name for this part and the data in the file is used as the data for this part. The encoding will be chosen appropriately for the file data.

Parameters:

- file - the File object to attach

Throws:

- java.io.IOException - errors related to accessing the file
- MessagingException - message related errors

Since:
JavaMail 1.4
Use the specified file to provide the data for this part. The simple file name is used as the file name for this part and the data in the file is used as the data for this part. The encoding will be chosen appropriately for the file data.

**Parameters:**
- `file` - the name of the file to attach

**Throws:**
- `IOException` - errors related to accessing the file
- `MessagingException` - message related errors

**Since:**
JavaMail 1.4

```java
public void saveFile(java.io.File file) throws java.io.IOException, MessagingException
```

MIME

<table>
<thead>
<tr>
<th>file</th>
<th>File</th>
</tr>
</thead>
</table>

**Throws**

- `IOException`: errors related to accessing the file
- `MessagingException`: message related errors

**since**
JavaMail 1.4

Save the contents of this part in the specified file. The content is decoded and saved, without any of the MIME headers.

**Parameters:**
- `file` - the File object to write to

**Throws:**
- `IOException` - errors related to accessing the file
- `MessagingException` - message related errors

**Since:**
public void saveFile(String file) throws java.io.IOException, MessagingException

MIME

file

Throws java.io.IOException: 
Throws MessagingException: 
since JavaMail 1.4

saveFile

public void saveFile(String file) throws IOException, MessagingException

Save the contents of this part in the specified file. The content is decoded and saved, without any of the MIME headers.

Parameters:
file - the name of the file to write to

Throws:
IOException - errors related to accessing the file
MessagingException - message related errors

Since:
JavaMail 1.4

public void writeTo(java.io.OutputStream os) throws java.io.IOException, MessagingException
RFC 822

Throws MessagingException: 
Throws java.io.IOException: javax.activation
See also writeTo
**writeTo**

```java
public void writeTo(OutputStream os)
throws IOException, MessagingException
```

Output the body part as an RFC 822 format stream.

**Specified by:**
writeTo in interface Part

**Throws:**
MessagingException
IOException - if an error occurs writing to the stream or if an error is generated by the javax.activation layer.

**See Also:**
DataHandler.writeTo(java.io.OutputStream)

---

**public String[] getHeader(String name) throws MessagingException**

**header_name** US-ASCII RFC 2047

- **name**
- **return**

See also
javax.mail.internet.MimeUtility

**getHeader**

```java
public String[] getHeader(String name)
throws MessagingException
```

Get all the headers for this header_name. Note that certain headers may be encoded as per RFC 2047 if they contain non US-ASCII characters and these should be decoded.

**Specified by:**
getHeader in interface Part

**Parameters:**
name - name of header
public String getHeader(String name, String delimiter) throws MessagingException

String delimiter delimiter
  name
delimiter
return

Throws: MessagingException

getHeader

public String getHeader(String name, String delimiter) throws MessagingException

String delimiter delimiter
  name
delimiter
return

Throws: MessagingException

Get all the headers for this header name, returned as a single String, with headers separated by the delimiter. If the delimiter is null, only the first header is returned.

Specified by: getHeader in interface MimePart

Parameters:
  name - the name of this header
delimiter - delimiter between fields in returned string

Returns:
  the value fields for all headers with this name

Throws: MessagingException

public void setHeader(String name, String value) throws
setHeader

public void setHeader(String name, String value)
throws MessagingException

Set the value for this header_name. Replaces all existing header values with this new value. Note that RFC 822 headers must contain only US-ASCII characters, so a header that contains non US-ASCII characters must be encoded as per the rules of RFC 2047.

Specified by:  
setHeader in interface Part

Parameters:

name - header name
value - header value

Throws:

MessagingException
IllegalWriteException - if the underlying implementation does not support modification of existing values

See Also:
MimeUtility

public void addHeader(String name, String value) throws MessagingException

header_name RFC 822  US-ASCII  US-ASCII  RFC 2047

name
value
addHeader

public void addHeader(String name, String value) throws MessagingException

Add this value to the existing values for this header_name. Note that
RFC 822 headers must contain only US-ASCII characters, so a
header that contains non US-ASCII characters must be encoded as
per the rules of RFC 2047.

Specified by:
   addHeader in interface Part

Parameters:
   name - header name
   value - header value

Throws:
   MessagingException
   IllegalWriteException - if the underlying implementation does
   not support modification of existing values

See Also:
   MimeUtility

public void removeHeader(String name) throws MessagingException

removeHeader

public void removeHeader(String name) throws MessagingException

Remove all headers with this name.
public java.util.Enumeration<E> getAllHeaders() throws MessagingException

Header Message
getAllHeaders

public Enumeration getAllHeaders() throws MessagingException

Return all the headers from this Message as an Enumeration of Header objects.

Specified by:

getAllHeaders in interface Part

Returns:
enumeration of Header objects

Throws:

MessagingException

public java.util.Enumeration<E> getMatchingHeaders(String[] names) throws MessagingException

Header Message
getMatchingHeaders

Specified by:

getAllHeaders in interface Part

Returns:
enumeration of Header objects

Throws:

MessagingException
getMatchingHeaders

public Enumeration getMatchingHeaders(String[] names) throws MessagingException

Return matching headers from this Message as an Enumeration of Header objects.

Specified by:
getMatchingHeaders in interface Part

Returns:
enumeration of Header objects

Throws:
MessagingException

getNonMatchingHeaders

public java.util.Enumeration<E>
getNonMatchingHeaders(String[] names) throws MessagingException
Header Message

getNonMatchingHeaders

public Enumeration getNonMatchingHeaders(String[] names) throws MessagingException

Return non-matching headers from this Message as an Enumeration of Header objects.

Specified by:
getNonMatchingHeaders in interface Part

Returns:
enumeration of Header objects

Throws:
MessagingException

public void addHeaderLine(String line) throws
**MessagingException**

**addHeaderLine**

```java
public void addHeaderLine(String line)
throws MessagingException
```

Add a header line to this body part

**Specified by:**

addHeaderLine in interface MimePart

**Throws:**

IllegalWriteException - if the underlying implementation does not support modification

**MessagingException**

---

```java
public java.util.Enumeration<String> getAllHeaderLines()
throws MessagingException
```

Get all header lines as an Enumeration of Strings. A Header line is a raw RFC 822 header line, containing both the "name" and "value" field.

**Specified by:**

getAllHeaderLines in interface MimePart

**Throws:**

MessagingException
public java.util.Enumeration<E> getMatchingHeaderLines(String[] names) throws MessagingException
String Header RFC 822 "name" "value"

getMatchingHeaderLines

public Enumeration getMatchingHeaderLines(String[] names) throws MessagingException

Get matching header lines as an Enumeration of Strings. A Header line is a raw RFC 822 header line, containing both the "name" and "value" field.

Specified by:
getMatchingHeaderLines in interface MimePart

Throws:
MessagingException

public java.util.Enumeration<E> getNonMatchingHeaderLines(String[] names) throws MessagingException
String Header RFC 822 "name" "value"

getNonMatchingHeaderLines

public Enumeration getNonMatchingHeaderLines(String[] names) throws MessagingException

Get non-matching header lines as an Enumeration of Strings. A Header line is a raw RFC 822 header line, containing both the "name" and "value" field.
protected void updateHeaders() throws MessagingException

MIME

- Store
  Message.saveChanges

updateHeaders

protected void updateHeaders()
    throws MessagingException

Examine the content of this body part and update the appropriate MIME headers. Typical headers that get set here are Content-Type and Content-Transfer-Encoding. Headers might need to be updated in two cases:
- A message being crafted by a mail application will certainly need to activate this method at some point to fill up its internal headers.
- A message read in from a Store will have obtained all its headers from the store, and so doesn't need this. However, if this message is editable and if any edits have been made to either the content or message structure, we might need to resync our headers.
In both cases this method is typically called by the Message.saveChanges method.

Throws:
    MessagingException
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.soap Class MimeHeader

java.lang.Object
   \ javax.xml.soap.MimeHeader

public class MimeHeader
extends Object

MIME MimeHeaders MimeHeader

See also javax.xml.soap.MimeHeaders

An object that stores a MIME header name and its value. One or more MimeHeader objects may be contained in a MimeHeaders object.

See Also:
    MimeHeaders

Constructor Summary

<table>
<thead>
<tr>
<th>MimeHeader(String name, String value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a MimeHeader object initialized with the given name and value.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>String getName()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the name of this MimeHeader object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String getValue()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the value of this MimeHeader object.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Constructor Detail

public MimeHeader(String name, String value)

MimeHeader

name String

codevalue String

MimeHeader

public MimeHeader(String name, String value)

Constructs a MimeHeader object initialized with the given name and value.

Parameters:

name - a String giving the name of the header
value - a String giving the value of the header

Method Detail

public String getName()

MimeHeader

return String

getName

public String getName()

Returns the name of this MimeHeader object.

Returns:
the name of the header as a String

```java
public String getValue()
    MimeHeader
    return String

getValue

public String getValue()

    Returns the value of this MimeHeader object.

    Returns:
    the value of the header as a String
```

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.soap  

**Class MimeHeaders**

```java
java.lang.Object
  |javax.xml.soap.MimeHeaders
```

public class MimeHeaders extends Object

MimeHeader MIME MIME

MIME AttachmentPart SAAJ API MIME

See also getAttachments, javax.xml.soap.AttachmentPart

A container for MimeHeader objects, which represent the MIME headers present in a MIME part of a message.

This class is used primarily when an application wants to retrieve specific attachments based on certain MIME headers and values. This class will most likely be used by implementations of AttachmentPart and other MIME dependent parts of the SAAJ API.

See Also:

SOAPMessage.getAttachments(), AttachmentPart

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MimeHeaders()</strong></td>
<td>Constructs a default MimeHeaders object initialized with an empty Vector object.</td>
</tr>
</tbody>
</table>

### Method Summary
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addHeader(String name, String value)</code></td>
<td>Adds a MimeHeader object with the specified name and value to this MimeHeaders object's list of headers.</td>
</tr>
<tr>
<td><code>Iterator getAllHeaders()</code></td>
<td>Returns all the MimeHeaders in this MimeHeaders object.</td>
</tr>
<tr>
<td><code>String[] getHeader(String name)</code></td>
<td>Returns all of the values for the specified header as an array of String objects.</td>
</tr>
<tr>
<td><code>Iterator getMatchingHeaders(String[] names)</code></td>
<td>Returns all the MimeHeader objects whose name matches a name in the given array of names.</td>
</tr>
<tr>
<td><code>Iterator getNonMatchingHeaders(String[] names)</code></td>
<td>Returns all of the MimeHeader objects whose name does not match a name in the given array of names.</td>
</tr>
<tr>
<td><code>void removeAllHeaders()</code></td>
<td>Removes all the header entries from this MimeHeaders object.</td>
</tr>
<tr>
<td><code>void removeHeader(String name)</code></td>
<td>Remove all MimeHeader objects whose name matches the given name.</td>
</tr>
<tr>
<td><code>void setHeader(String name, String value)</code></td>
<td>Replaces the current value of the first header entry whose name matches the given name with the given value, adding a new header if no existing header name matches.</td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public MimeHeaders()
```

Vector MimeHeaders
MimeHeaders

public MimeHeaders()

Constructs a default MimeHeaders object initialized with an empty Vector object.

Method Detail

public String[] getHeader(String name)

String

name

return String

See also

setHeader

g.getHeader

public String[] getHeader(String name)

Returns all of the values for the specified header as an array of String objects.

Parameters:

name - the name of the header for which values will be returned

Returns:

a String array with all of the values for the specified header

See Also:

setHeader(java.lang.String, java.lang.String)

g.getHeader

public void setHeader(String name, String value)

RFC822 US-ASCII

name String

value String
**setHeader**

public void **setHeader**(String name, String value)

Replaces the current value of the first header entry whose name matches the given name with the given value, adding a new header if no existing header name matches. This method also removes all matching headers after the first one.

Note that RFC822 headers can contain only US-ASCII characters.

**Parameters:**
- name - a String with the name of the header for which to search
- value - a String with the value that will replace the current value of the specified header

**Throws:**
- IllegalArgumentException - if there was a problem in the mime header name or the value being set

**See Also:**
- [getHeader](#)

---

**public void addHeader(String name, String value)**

MimeHeader MimeHeaders

RFC822 US-ASCII

- name String
- value String

**Throws**
- IllegalArgumentException: MIME
addHeader

public void addHeader(String name, String value)

Adds a MimeHeader object with the specified name and value to this MimeHeaders object's list of headers.

Note that RFC822 headers can contain only US-ASCII characters.

Parameters:
- name - a String with the name of the header to be added
- value - a String with the value of the header to be added

Throws:
- IllegalArgumentException - if there was a problem in the mime header name or value being added

removeHeader

public void removeHeader(String name)

Remove all MimeHeader objects whose name matches the given name.

Parameters:
- name - a String with the name of the header for which to search

removeAllHeaders

public void removeAllHeaders()

Remove all MimeHeader objects from the list.
removeAllHeaders

public void removeAllHeaders()

Removes all the header entries from this MimeHeaders object.

public java.util.Iterator<E> getAllHeaders()

MimeHeaders MimeHeader
return Iterator MimeHeaders MimeHeader

getAllHeaders

public Iterator getAllHeaders()

Returns all the MimeHeaders in this MimeHeaders object.

Returns:

an Iterator object over this MimeHeaders object's list of MimeHeader objects

getMatchingHeaders

public java.util.Iterator<E> getMatchingHeaders(String[] names)

MimeHeader

String
return Iterator MimeHeader

getMatchingHeaders

public Iterator getMatchingHeaders(String[] names)

Returns all the MimeHeader objects whose name matches a name in the given array of names.

Parameters:
getNonMatchingHeaders

public java.util.Iterator<E>
getNonMatchingHeaders(String[] names)

Returns:
an Iterator object over the MimeHeader objects whose name matches one of the names in the given list

Parameters:
 names - an array of String objects with the names for which to search

Returns:
an Iterator object over the MimeHeader objects whose name matches one of the names in the given list

Returns all of the MimeHeader objects whose name does not match a name in the given array of names.

Parameters:
 names - an array of String objects with the names for which to search

Returns:
an Iterator object over the MimeHeader objects whose name does not match one of the names in the given list
PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.mail.internet  Class MimeMessage

java.lang.Object
  ├ javax.mail.Message
  │  └ javax.mail.internet.MimeMessage

All Implemented Interfaces:
  MimePart, Part

public class MimeMessage
  extends Message
  implements MimePart

Extends: Message
Implements: MimePart
Inner classes: MimeMessage.RecipientType

MIME  Message  MimePart

MIME  MimeMessage

MIME  MimeMessage  MIME

MimeMessage  InternetHeaders  RFC 822

mail.mime.address.strict  "false"

InternetAddress

---

RFC 822  MIME

RFC 822   US-ASCII  ASCII MIME RFC 2047
MimeUtility  setHeader  addHeader  addHeaderLine  MIME
SMTP  1000

See also
  javax.mail.internet.MimeUtility, javax.mail.Part,
  javax.mail.Message, javax.mail.internet.MimePart,
  javax.mail.internet.InternetAddress
This class represents a MIME style email message. It implements the Message abstract class and the MimePart interface.

Clients wanting to create new MIME style messages will instantiate an empty MimeMessage object and then fill it with appropriate attributes and content.

Service providers that implement MIME compliant backend stores may want to subclass MimeMessage and override certain methods to provide specific implementations. The simplest case is probably a provider that generates a MIME style input stream and leaves the parsing of the stream to this class.

MimeMessage uses the InternetHeaders class to parse and store the top level RFC 822 headers of a message.

The mail.mime.address.strict session property controls the parsing of address headers. By default, strict parsing of address headers is done. If this property is set to "false", strict parsing is not done and many illegal addresses that sometimes occur in real messages are allowed. See the InternetAddress class for details.

---

A note on RFC 822 and MIME headers

RFC 822 header fields must contain only US-ASCII characters. MIME allows non ASCII characters to be present in certain portions of certain headers, by encoding those characters. RFC 2047 specifies the rules for doing this. The MimeUtility class provided in this package can be used to achieve this. Callers of the setHeader, addHeader, and addHeaderLine methods are responsible for enforcing the MIME requirements for the specified headers. In addition, these header fields must be folded (wrapped) before being sent if they exceed the line length limitation for the transport (1000 bytes for SMTP). Received headers may have been folded. The application is responsible for folding and unfolding headers as appropriate.

Author:
### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>MimeMessage.RecipientType</th>
</tr>
</thead>
<tbody>
<tr>
<td>This inner class extends the javax.mail.Message.RecipientType class to add additional RecipientTypes.</td>
<td></td>
</tr>
</tbody>
</table>

### Field Summary

<table>
<thead>
<tr>
<th>protected byte[]</th>
<th>content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Byte array that holds the bytes of this Message's content.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected InputStream</th>
<th>contentStream</th>
</tr>
</thead>
<tbody>
<tr>
<td>If the data for this message was supplied by an InputStream that implements the SharedInputStream interface, contentStream is another such stream representing the content of this message.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected DataHandler</th>
<th>dh</th>
</tr>
</thead>
<tbody>
<tr>
<td>The DataHandler object representing this Message's content.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected Flags</th>
<th>flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Flags for this message.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected InternetHeaders</th>
<th>headers</th>
</tr>
</thead>
<tbody>
<tr>
<td>The InternetHeaders object that stores the header of this message.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected boolean</th>
<th>modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>A flag indicating whether the message has been modified.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected boolean</th>
<th>saved</th>
</tr>
</thead>
<tbody>
<tr>
<td>Does the saveChanges method need to be called on this message?</td>
<td></td>
</tr>
</tbody>
</table>
### Fields inherited from class `javax.mail.Message`  
expunged, folder, msgnum, session

### Fields inherited from interface `javax.mail.Part`  
ATTACHMENT, INLINE

### Constructor Summary

- **protected**  
  `MimeMessage(Folder folder, InputStream is, int msgnum)`  
  Constructs a MimeMessage by reading and parsing the data from the specified MIME InputStream.

- **protected**  
  `MimeMessage(Folder folder, int msgnum)`  
  Constructs an empty MimeMessage object with the given Folder and message number.

- **protected**  
  `MimeMessage(Folder folder, InternetHeaders headers, byte[] content, int msgnum)`  
  Constructs a MimeMessage from the given InternetHeaders object and content.

- **protected**  
  `MimeMessage(MimeMessage source)`  
  Constructs a new MimeMessage with content initialized from the source MimeMessage.

- **protected**  
  `MimeMessage(Session session)`  
  Default constructor.

- **protected**  
  `MimeMessage(Session session, InputStream is)`  
  Constructs a MimeMessage by reading and parsing the data from the specified MIME InputStream.

### Method Summary

- **void**  
  `addFrom(Address[] addresses)`  
  Add the specified addresses to the existing "From" field.

- **void**  
  `addHeader(String name, String value)`  
  Add this value to the existing values for this header_name.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addHeaderLine(String line)</code></td>
<td>Add a raw RFC 822 header-line.</td>
</tr>
<tr>
<td><code>addRecipients(Message.RecipientType type, Address[] addresses)</code></td>
<td>Add the given addresses to the specified recipient type.</td>
</tr>
<tr>
<td><code>addRecipients(Message.RecipientType type, String addresses)</code></td>
<td>Add the given addresses to the specified recipient type.</td>
</tr>
<tr>
<td><code>createInternetHeaders(InputStream is)</code></td>
<td>Create and return an InternetHeaders object that loads the headers from the given InputStream.</td>
</tr>
<tr>
<td><code>createMimeMessage(Session session)</code></td>
<td>Create and return a MimeMessage object.</td>
</tr>
<tr>
<td><code>getAllHeaderLines()</code></td>
<td>Get all header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><code>getAllHeaders()</code></td>
<td>Return all the headers from this Message as an enumeration of Header objects.</td>
</tr>
<tr>
<td><code>getAllRecipients()</code></td>
<td>Get all the recipient addresses for the message.</td>
</tr>
<tr>
<td><code>getContent()</code></td>
<td>Return the content as a Java object.</td>
</tr>
<tr>
<td><code>getContentID()</code></td>
<td>Returns the value of the &quot;Content-ID&quot; header field.</td>
</tr>
<tr>
<td><code>getContentLanguage()</code></td>
<td>Get the languages specified in the &quot;Content-Language&quot; header field of this message.</td>
</tr>
<tr>
<td><code>getContentMD5()</code></td>
<td>Return the value of the &quot;Content-MD5&quot; header field.</td>
</tr>
<tr>
<td><code>getContentStream()</code></td>
<td>Produce the raw bytes of the content.</td>
</tr>
<tr>
<td><code>getContentType()</code></td>
<td>Returns the value of the RFC 822 &quot;Content-Type&quot; header field.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getDataHandler()</code></td>
<td>Return a DataHandler for this Message's content.</td>
</tr>
<tr>
<td><code>getDescription()</code></td>
<td>Returns the &quot;Content-Description&quot; header field of this Message.</td>
</tr>
<tr>
<td><code>getDisposition()</code></td>
<td>Returns the value of the &quot;Content-Disposition&quot; header field.</td>
</tr>
<tr>
<td><code>getEncoding()</code></td>
<td>Returns the content transfer encoding from the &quot;Content-Transfer-Encoding&quot; header field.</td>
</tr>
<tr>
<td><code>getFileName()</code></td>
<td>Get the filename associated with this Message.</td>
</tr>
<tr>
<td><code>getFlags()</code></td>
<td>Return a Flags object containing the flags for this message.</td>
</tr>
<tr>
<td><code>getFrom()</code></td>
<td>Returns the value of the RFC 822 &quot;From&quot; header fields.</td>
</tr>
<tr>
<td><code>getHeader(String name)</code></td>
<td>Get all the headers for this header_name.</td>
</tr>
<tr>
<td><code>getHeader(String name, String delimiter)</code></td>
<td>Get all the headers for this header name, returned as a single String, with headers separated by the delimiter.</td>
</tr>
<tr>
<td><code>getInputStream()</code></td>
<td>Return a decoded input stream for this Message's &quot;content&quot;.</td>
</tr>
<tr>
<td><code>getLineCount()</code></td>
<td>Return the number of lines for the content of this message.</td>
</tr>
<tr>
<td><code>getMatchingHeaderLines(String[] names)</code></td>
<td>Get matching header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td><code>getMatchingHeaders(String[] names)</code></td>
<td>Return matching headers from this Message as an</td>
</tr>
</tbody>
</table>
### Enumeration of Header objects.

- **String getMessageID()**
  - Returns the value of the "Message-ID" header field.

- **Enumeration getNonMatchingHeaderLines(String[] names)**
  - Get non-matching header lines as an Enumeration of Strings.

- **Enumeration getNonMatchingHeaders(String[] names)**
  - Return non-matching headers from this Message as an Enumeration of Header objects.

- **InputStream getRawInputStream()**
  - Return an InputStream to the raw data with any Content-Transfer-Encoding intact.

- **Date getReceivedDate()**
  - Returns the Date on this message was received.

- **Address[] getRecipients(Message.RecipientType type)**
  - Returns the recepients specified by the type.

- **Address[] getReplyTo()**
  - Return the value of the RFC 822 "Reply-To" header field.

- **Address getSender()**
  - Returns the value of the RFC 822 "Sender" header field.

- **Date getSentDate()**
  - Returns the value of the RFC 822 "Date" field.

- **int getSize()**
  - Return the size of the content of this message in bytes.

- **String getSubject()**
  - Returns the value of the "Subject" header field.

- **boolean isMimeType(String mimeType)**
  - Is this Part of the specified MIME type?

- **boolean isSet(Flags.Flag flag)**
  - Check whether the flag specified in the flag argument is set in this message.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected void</td>
<td><code>parse(InputStream is)</code> Parse the InputStream setting the headers and content fields appropriately.</td>
</tr>
<tr>
<td>void</td>
<td><code>removeHeader(String name)</code> Remove all headers with this name.</td>
</tr>
<tr>
<td><code>Message</code></td>
<td><code>reply(boolean replyToAll)</code> Get a new Message suitable for a reply to this message.</td>
</tr>
<tr>
<td>void</td>
<td><code>saveChanges()</code> Updates the appropriate header fields of this message to be consistent with the message's contents.</td>
</tr>
<tr>
<td>void</td>
<td><code>setContent(Multipart mp)</code> This method sets the Message's content to a Multipart object.</td>
</tr>
<tr>
<td>void</td>
<td><code>setContent(Object o, String type)</code> A convenience method for setting this Message's content.</td>
</tr>
<tr>
<td>void</td>
<td><code>setContentID(String cid)</code> Set the &quot;Content-ID&quot; header field of this Message.</td>
</tr>
<tr>
<td>void</td>
<td><code>setContentLanguage(String[] languages)</code> Set the &quot;Content-Language&quot; header field of this MimePart.</td>
</tr>
<tr>
<td>void</td>
<td><code>setContentMD5(String md5)</code> Set the &quot;Content-MD5&quot; header field of this Message.</td>
</tr>
<tr>
<td>void</td>
<td><code>setDataHandler(DataHandler dh)</code> This method provides the mechanism to set this part's content.</td>
</tr>
<tr>
<td>void</td>
<td><code>setDescription(String description)</code> Set the &quot;Content-Description&quot; header field for this Message.</td>
</tr>
<tr>
<td>void</td>
<td><code>setDescription(String description, String charset)</code> Set the &quot;Content-Description&quot; header field for this Message.</td>
</tr>
<tr>
<td>void</td>
<td><code>setDisposition(String disposition)</code> Set the &quot;Content-Disposition&quot; header field of this Message.</td>
</tr>
<tr>
<td>Method</td>
<td>Parameters</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
</tr>
<tr>
<td><code>void setFileName(String filename)</code></td>
<td>Sets the filename associated with this part, if possible.</td>
</tr>
<tr>
<td><code>void setFlags(Flags flag, boolean set)</code></td>
<td>Sets the flags for this message.</td>
</tr>
<tr>
<td><code>void setFrom()</code></td>
<td>Sets the RFC 822 &quot;From&quot; header field using the value of the <code>InternetAddress.getLocalAddress</code> method.</td>
</tr>
<tr>
<td><code>void setFrom(Address address)</code></td>
<td>Sets the RFC 822 &quot;From&quot; header field.</td>
</tr>
<tr>
<td><code>void setHeader(String name, String value)</code></td>
<td>Sets the value for this header name.</td>
</tr>
<tr>
<td><code>void setRecipients(Message.RecipientType type, Address[] addresses)</code></td>
<td>Sets the specified recipient type to the given addresses.</td>
</tr>
<tr>
<td><code>void setRecipients(Message.RecipientType type, String addresses)</code></td>
<td>Sets the specified recipient type to the given addresses.</td>
</tr>
<tr>
<td><code>void setReplyTo(Address[] addresses)</code></td>
<td>Sets the RFC 822 &quot;Reply-To&quot; header field.</td>
</tr>
<tr>
<td><code>void setSender(Address address)</code></td>
<td>Sets the RFC 822 &quot;Sender&quot; header field.</td>
</tr>
<tr>
<td><code>void setSentDate(Date d)</code></td>
<td>Sets the RFC 822 &quot;Date&quot; header field.</td>
</tr>
<tr>
<td><code>void setSubject(String subject)</code></td>
<td>Sets the &quot;Subject&quot; header field.</td>
</tr>
<tr>
<td><code>void setSubject(String subject, String charset)</code></td>
<td>Sets the &quot;Subject&quot; header field.</td>
</tr>
<tr>
<td><code>void setText(String text)</code></td>
<td>Convenience method that sets the given String as this part's content, with a MIME type of &quot;text/plain&quot;.</td>
</tr>
</tbody>
</table>
| `void setText(String text, String charset)` | Convenience method that sets the given String as
### Field Detail

<table>
<thead>
<tr>
<th>dh</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected DataHandler dh</td>
</tr>
</tbody>
</table>

The DataHandler object representing this Message's content.
protected byte[] content

Byte array that holds the bytes of this Message's content.

contentStream

protected InputStream contentStream

If the data for this message was supplied by an InputStream that implements the SharedInputStream interface, contentStream is another such stream representing the content of this message. In this case, content will be null.

Since:
JavaMail 1.2

headers

protected InternetHeaders headers

The InternetHeaders object that stores the header of this message.

flags

protected Flags flags

The Flags for this message.
modified

protected boolean modified

A flag indicating whether the message has been modified. If the message has not been modified, any data in the content array is assumed to be valid and is used directly in the writeTo method. This flag is set to true when an empty message is created or when the saveChanges method is called.

Since:
JavaMail 1.2

saved

protected boolean saved

Does the saveChanges method need to be called on this message? This flag is set to false by the public constructor and set to true by the saveChanges method. The writeTo method checks this flag and calls the saveChanges method as necessary. This avoids the common mistake of forgetting to call the saveChanges method on a newly constructed message.

Since:
JavaMail 1.2

Constructor Detail

public MimeMessage(<Session session>)

headers InternetHeaders flag

modified true
MimeMessage

`public MimeMessage(Session session)`

Default constructor. An empty message object is created. The `headers` field is set to an empty `InternetHeaders` object. The `flags` field is set to an empty `Flags` object. The `modified` flag is set to `true`.

`public MimeMessage(Session session, java.io.InputStream is) throws MessagingException`

MIME InputStream MimeMessageInputStream

**MIME**

`session
is
Throws MessagingException:`

MimeMessage

`public MimeMessage(Session session, InputStream is) throws MessagingException`

Constructs a `MimeMessage` by reading and parsing the data from the specified MIME InputStream. The InputStream will be left positioned at the end of the data for the message. Note that the input stream parse is done within this constructor itself.

The input stream contains an entire MIME formatted message with headers and data.

**Parameters:**
session - Session object for this message
is - the message input stream

Throws:
MessagingException

public MimeMessage(MimeMessage source) throws MessagingException

source MimeMessage  MimeMessage

source

Throws
MessagingException:

since
JavaMail 1.2

MimeMessage

public MimeMessage(MimeMessage source)
throws MessagingException

Constructs a new MimeMessage with content initialized from the source MimeMessage. The new message is independent of the original.

Note: The current implementation is rather inefficient, copying the data more times than strictly necessary.

Parameters:
source - the message to copy content from

Throws:
MessagingException

Since:
JavaMail 1.2

protected MimeMessage(Folder folder, int msgnum)
Folder MimeMessage

MimeMessage

protected MimeMessage(Folder folder, int msgnum)

Constructs an empty MimeMessage object with the given Folder and message number.

This method is for providers subclassing MimeMessage.

protected MimeMessage(Folder folder, java.io.InputStream is, int msgnum) throws MessagingException
MIME InputStream MimeMessageInputStream

MimeMessage

folder

is

msgnum

Throws MessagingException:

MimeMessage

protected MimeMessage(Folder folder, InputStream is, int msgnum)

throws MessagingException

Constructs a MimeMessage by reading and parsing the data from
the specified MIME InputStream. The InputStream will be left positioned at the end of the data for the message. Note that the input stream parse is done within this constructor itself.

This method is for providers subclassing MimeMessage.

**Parameters:**
- folder - The containing folder.
- is - the message input stream
- msgnum - Message number of this message within its folder

**Throws:**
- MessagingException

```java
protected MimeMessage(Folder folder, InternetHeaders headers, byte[] content, int msgnum) throws MessagingException
```

**InternetHeaders MimeMessage**

```java
folder
headers
content
msgnum
```

**Throws**
- MessagingException

**MimeMessage**

```java
protected MimeMessage(Folder folder,
                       InternetHeaders headers,
                       byte[] content,
                       int msgnum)
throws MessagingException
```

Constructs a MimeMessage from the given InternetHeaders object and content. This method is for providers subclassing MimeMessage.

**Parameters:**
- folder - The containing folder.
- headers - The headers
content - The message content
msgnum - Message number of this message within its folder

Throws:
MessagingException

---

**Method Detail**

```java
protected void parse(java.io.InputStream is) throws MessagingException
```

**InputStream**

`is`  

Throws
MessagingException:

---

**parse**

```java
protected void parse(InputStream is) throws MessagingException
```

Parse the InputStream setting the headers and content fields appropriately. Also resets the modified flag.

This method is intended for use by subclasses that need to control when the InputStream is parsed.

**Parameters:**

is - The message input stream

**Throws:**
MessagingException

---

```java
public Address[] getFrom() throws MessagingException
```

RFC 822 "From" "Sender" "Sender"

null
getFrom

public Address[] getFrom() throws MessagingException

Returns the value of the RFC 822 "From" header fields. If this header field is absent, the "Sender" header field is used. If the "Sender" header field is also absent, null is returned.

This implementation uses the getHeader method to obtain the requisite header field.

Specified by:
    getFrom in class Message

Returns:
    Address object

Throws:
    MessagingException

See Also:
    headers

public void setFrom(Address address) throws MessagingException

RFC 822 "From" address

address

Throws
    IllegalArgumentException:
    Throws
    IllegalStateException: READ_ONLY
    Throws
    MessagingException:
setFrom

public void setFrom(Address address) throws MessagingException

Set the RFC 822 "From" header field. Any existing values are replaced with the given address. If address is null, this header is removed.

Specified by:
    setFrom in class Message

Parameters:
    address - the sender of this message

Throws:
    IllegalWriteException - if the underlying implementation does not support modification of existing values
    IllegalStateException - if this message is obtained from a READ_ONLY folder.
    MessagingException

public void setFrom() throws MessagingException

    InternetAddress.getLocalAddress RFC 822 "From"

    Throws IllegalWriteException:
    Throws IllegalStateException: READ_ONLY
    Throws MessagingException:

setFrom

public void setFrom() throws MessagingException

Set the RFC 822 "From" header field using the value of the InternetAddress.getLocalAddress method.

Specified by:
    setFrom in class Message

Throws:
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public void addFrom(Address[] addresses) throws MessagingException
"From" "From"
addresses
Throws IllegalWriteException:
Throws IllegalStateException:  READ_ONLY
Throws MessagingException:

addFrom

public void addFrom(Address[] addresses) throws MessagingException
Add the specified addresses to the existing "From" field. If the "From" field does not already exist, it is created.

Specified by:
addFrom in class Message
Parameters:
addresses - the senders of this message
Throws:
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public Address getSender() throws MessagingException
RFC 822 "Sender" "Sender"
getHeader

return Address

Throws MessagingException:

since JavaMail 1.3

See also headers

getSender

public Address getSender() throws MessagingException

    Returns the value of the RFC 822 "Sender" header field. If the "Sender" header field is absent, null is returned.

    This implementation uses the getHeader method to obtain the requisite header field.

    Returns:
    Address object

    Throws:
    MessagingException

    Since:
    JavaMail 1.3

    See Also:
    headers

public void setSender(Address address) throws MessagingException

RFC 822 "Sender" address

address

Throws IllegalArgumentException:

Throws IllegalStateException: READ_ONLY

Throws MessagingException:

since JavaMail 1.3
setSender

public void setSender(Address address) throws MessagingException

Set the RFC 822 "Sender" header field. Any existing values are replaced with the given address. If address is null, this header is removed.

Parameters:
address - the sender of this message

Throws:
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

Since:
JavaMail 1.3

public Address[] getRecipients(Message.RecipientType type) throws MessagingException

RFC 822

Message.RecipientType.TO "To"
Message.RecipientType.CC "Cc"
Message.RecipientType.BCC "Bcc"
MimeMessage.RecipientType.NEWSGROUPS "Newsgroups"

null

g.getHeader

return Address
getRecipients

public Address[] getRecipients(Message.RecipientType type) throws MessagingException

Returns the recipients specified by the type. The mapping between the type and the corresponding RFC 822 header is as follows:

- Message.RecipientType.TO
- Message.RecipientType.CC
- Message.RecipientType.BCC
- MimeMessage.RecipientType.NEWSGROUPS

Returns null if the header specified by the type is not found or if its value is empty.

This implementation uses the getHeader method to obtain the requisite header field.

Specified by: getRecipients in class Message

Parameters:
- type - Type of recipient

Returns:
- array of Address objects

Throws:
- MessagingException - if header could not be retrieved
- AddressException - if the header is misformatted

See Also:
- headers, Message.RecipientType.TO, Message.RecipientType.CC,
  Message.RecipientType.BCC,
  MimeMessage.RecipientType.NEWSGROUPS
public Address[] getAllRecipients() throws MessagingException

TOCCBCC  NEWSGROUPS

return Address

Throws MessagingException:

See also TO, CC, BCC, NEWSGROUPS

getAllRecipients

public Address[] getAllRecipients()

throws MessagingException

Get all the recipient addresses for the message. Extracts the TO, CC, BCC, and NEWSGROUPS recipients.

Overrides:

getAllRecipients in class Message

Returns:
array of Address objects

Throws:
MessagingException

See Also:
Message.RecipientType.TO, Message.RecipientType.CC,
Message.RecipientType.BCC,
MimeMessage.RecipientType.NEWSGROUPS

public void setRecipients(Message.RecipientType type, Address[] addresses) throws MessagingException

null

type

addresses

Throws IllegalArgumentException:

Throws IllegalStateException: READ_ONLY

Throws MessagingException:

See also getRecipients
### setRecipients

```java
public void setRecipients(Message.RecipientType type,
                        Address[] addresses)
    throws MessagingException
```

Set the specified recipient type to the given addresses. If the address parameter is `null`, the corresponding recipient field is removed.

**Specified by:**
- `setRecipients` in class `Message`

**Parameters:**
- `type` - Recipient type
- `addresses` - Addresses

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this message is obtained from a `READ_ONLY` folder.
- `MessagingException`

**See Also:**
- `getRecipients(javax.mail.Message.RecipientType)`

---

```java
public void setRecipients(Message.RecipientType type,
                        String addresses)
    throws MessagingException
```

**Throws**
- `AddressException`
- `IllegalWriteException`
- `IllegalStateException: READ_ONLY`

**See also**
- `getRecipients`
setRecipients

public void setRecipients(Message.RecipientType type,
                          String addresses)
                          throws MessagingException

Set the specified recipient type to the given addresses. If the address parameter is null, the corresponding recipient field is removed.

Parameters:

  type - Recipient type
  addresses - Addresses

Throws:

  AddressException - if the attempt to parse the addresses String fails
  IllegalWriteException - if the underlying implementation does not support modification of existing values
  IllegalStateException - if this message is obtained from a READ_ONLY folder.
  MessagingException

Since:

  JavaMail 1.2

See Also:

  getRecipients(javax.mail.Message.RecipientType)

public void addRecipients(Message.RecipientType type,
Address[] addresses) throws MessagingException

  type
  addresses

  Throws          IllegalWriteException:
  Throws  IllegalStateException:  READ_ONLY
  Throws  MessagingException:
addRecipients

public void addRecipients(Message.RecipientType type,
                          Address[] addresses)
                          throws MessagingException

Add the given addresses to the specified recipient type.

Specified by:
    addRecipients in class Message

Parameters:
    type - Recipient type
    addresses - Addresses

Throws:
    IllegalWriteException - if the underlying implementation does
    not support modification of existing values
    IllegalStateException - if this message is obtained from a
    READ_ONLY folder.
    MessagingException

public void addRecipients(Message.RecipientType type,
                          String addresses) throws MessagingException

      type
addresses

Throws             AddressException:
Throws             IllegalWriteException:
Throws             IllegalStateException: READ_ONLY
Throws             MessagingException:
since              JavaMail 1.2
Add the given addresses to the specified recipient type.

**Parameters:**
- `type` - Recipient type
- `addresses` - Addresses

**Throws:**
- `AddressException` - if the attempt to parse the addresses String fails
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this message is obtained from a READ_ONLY folder.
- `MessagingException`

**Since:**
JavaMail 1.2

```java
public Address[] getReplyTo() throws MessagingException
RFC 822 "Reply-To"
```

- **Throws**
  - `MessagingException`
- **See also**
  - headers

**getReplyTo**

```java
public Address[] getReplyTo()
```

Return the value of the RFC 822 "Reply-To" header field. If this header is unavailable or its value is absent, then the `getFrom` method is called and its value is returned. This implementation uses the `getHeader` method to obtain the requisite header field.

**Overrides:**
- `getReplyTo` in class `Message`

**Returns:**
- addresses to which replies should be directed

**Throws:**
public void setReplyTo(Address[] addresses) throws MessagingException
RFC 822 "Reply-To" null

<table>
<thead>
<tr>
<th>Throws</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td>IllegalWriteException:</td>
<td></td>
</tr>
<tr>
<td>IllegalStateException: READ_ONLY:</td>
<td></td>
</tr>
<tr>
<td>MessagingException:</td>
<td></td>
</tr>
</tbody>
</table>

setReplyTo

public void setReplyTo(Address[] addresses) throws MessagingException

Set the RFC 822 "Reply-To" header field. If the address parameter is null, this header is removed.

Overrides:
setReplyTo in class Message

Parameters:
addresses - addresses to which replies should be directed

Throws:
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

MethodNotSupportedException - if the underlying implementation does not support setting this attribute

public String getSubject() throws MessagingException
"Subject" subject null
subject  RFC 2047  Unicode

getHeader
return Subject
Throws MessagingException: en
See also headers

getSubject

public String getSubject() throws MessagingException

Returns the value of the "Subject" header field. Returns null if the
subject field is unavailable or its value is absent.

If the subject is encoded as per RFC 2047, it is decoded and
converted into Unicode. If the decoding or conversion fails, the raw
data is returned as is.

This implementation uses the getHeader method to obtain the
requisite header field.

Specified by: getSubject in class Message
Returns: Subject
Throws: MessagingException
See Also: headers

public void setSubject(String subject) throws MessagingException
"Subject" subject US-ASCII subject
US-ASCII subject null "Subject"
subject

MessagingException MessagingException
UnsupportedEncodingException

subject Subject

Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException.: UnsupportedEncodingException

setSubject

public void setSubject(String subject)
    throws MessagingException

Set the "Subject" header field. If the subject contains non US-ASCII characters, it will be encoded using the platform's default charset. If the subject contains only US-ASCII characters, no encoding is done and it is used as-is. If the subject is null, the existing "Subject" field is removed.

The application must ensure that the subject does not contain any line breaks.

Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.

Specified by:
    setSubject in class Message

Parameters:
subject - The subject

Throws:
    IllegalWriteException - if the underlying implementation does not support modification of existing values
    IllegalStateException - if this message is obtained from a
READ_ONLY folder.

MessagingException. - An UnsupportedEncodingException may be included in the exception chain if the charset conversion fails.

```
public void setSubject(String subject, String charset)
throws MessagingException
"Subject" subject US-ASCII subject
US-ASCII subject null "Subject"

subject

MessagingException MessagingException
UnsupportedEncodingException

subject Subject
charset

Throws IllegalArgumentException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException.: UnsupportedEncodingException
```

```
setSubject

public void setSubject(String subject, String charset)
throws MessagingException

Set the "Subject" header field. If the subject contains non US-ASCII characters, it will be encoded using the specified charset. If the subject contains only US-ASCII characters, no encoding is done and it is used as-is. If the subject is null, the existing "Subject" header field is removed.

The application must ensure that the subject does not contain any line breaks.
```
Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.

**Parameters:**
- subject - The subject
- charset - The charset

**Throws:**
- IllegalArgumentException - if the underlying implementation does not support modification of existing values
- IllegalStateException - if this message is obtained from a READ_ONLY folder.
- MessagingException. - An UnsupportedEncodingException may be included in the exception chain if the charset conversion fails.

```java
public java.util.Date getSentDate() throws MessagingException

RFC 822 "Date" null
```

```java
getHeader
return
```

**Throws**
- MessagingException:

### getSentDate

```java
public Date getSentDate()
throws MessagingException
```

Returns the value of the RFC 822 "Date" field. This is the date on which this message was sent. Returns null if this field is unavailable or its value is absent.

This implementation uses the `getHeader` method to obtain the
requisite header field.

Specified by:
getSentDate in class Message

Returns:
The sent Date

Throws:
MessagingException

public void setSentDate(java.util.Date d) throws MessagingException
RFC 822 "Date"

Throws
IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:

setSentDate

public void setSentDate(Date d)
throws MessagingException

Set the RFC 822 "Date" header field. This is the date on which the creator of the message indicates that the message is complete and ready for delivery. If the date parameter is null, the existing "Date" field is removed.

Specified by:
setSentDate in class Message

Parameters:
d - the sent date of this message

Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException
public java.util.Date getReceivedDate() throws MessagingException

RFC 822

    null
    return

    Throws

getReceivedDate

public Date getReceivedDate()
    throws MessagingException

    Returns the Date on this message was received. Returns null if this
date cannot be obtained.

    Note that RFC 822 does not define a field for the received date.
Hence only implementations that can provide this date need return a
valid value.

    This implementation returns null.

    Specified by:
    getReceivedDate in class Message

    Returns:
    the date this message was received

    Throws:
    MessagingException

public int getSize() throws MessagingException
-1
GetSize

public int getSize() throws MessagingException

Return the size of the content of this message in bytes. Return -1 if the size cannot be determined.

Note that this number may not be an exact measure of the content size and may or may not account for any transfer encoding of the content.

This implementation returns the size of the content array (if not null), or, if contentStream is not null, and the available method returns a positive number, it returns that number as the size. Otherwise, it returns -1.

Specified by:
getSize in interface Part

Returns:
size of content in bytes

Throws:
MessagingException
public int getLineCount() throws MessagingException

Return the number of lines for the content of this message. Return -1 if this number cannot be determined.

Note that this number may not be an exact measure of the content length and may or may not account for any transfer encoding of the content.

This implementation returns -1.

Specified by: getLineCount in interface Part

Returns: number of lines in the content.

Throws: MessagingException

public String getContentType() throws MessagingException

RFC 822 "Content-Type" null "text/plain"

getHeader

return

Throws: MessagingException

See also javax.activation.DataHandler
**getContentType**

```java
public String getContentType() throws MessagingException {
    return getHeader("Content-Type").getValue();
}
```

Returns the value of the RFC 822 "Content-Type" header field. This represents the content-type of the content of this message. This value must not be null. If this field is unavailable, "text/plain" should be returned.

This implementation uses the `getHeader` method to obtain the requisite header field.

**Specified by:**
`getContentType` in interface `Part`

**Returns:**
The `ContentType` of this part

**Throws:**
`MessagingException`

**See Also:**
`DataHandler`

---

**public boolean isMimeType(String mimeType) throws MessagingException**

```java
public boolean isMimeType(String mimeType) throws MessagingException {
    if (mimeType.equals("text/plain") || mimeType.equals("text/plain; charset=foobar") || mimeType.equals("*")) {
        return true;
    }
    return false;
}
```

**isMimeType**

```java
public boolean isMimeType(String mimeType) throws MessagingException {
    if (mimeType.equals("text/plain") || mimeType.equals("text/plain; charset=foobar") || mimeType.equals("*")) {
        return true;
    }
    return false;
}
```
Is this Part of the specified MIME type? This method compares **only** the primaryType and subType. The parameters of the content types are ignored.

For example, this method will return `true` when comparing a Part of content type "text/plain" with "text/plain; charset=foobar".

If the subType of mimeType is the special character "*", then the subtype is ignored during the comparison.

**Specified by:**

`isMimeType` in interface `Part`

**Throws:**

`MessagingException`

```
public String getDisposition() throws MessagingException

"Content-Disposition" disposition disposition
```

**Content-Disposition**

<table>
<thead>
<tr>
<th>getHeader</th>
<th>disposition null</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>null</td>
</tr>
</tbody>
</table>

**Throws**

`MessagingException`

```
getDisposition
```

Returns the value of the "Content-Disposition" header field. This represents the disposition of this part. The disposition describes how the part should be presented to the user.

If the Content-Disposition field is unavailable, `null` is returned.
This implementation uses the `getHeader` method to obtain the requisite header field.

**Specified by:**
getDisposition in interface Part

**Returns:**
disposition of this part, or null if unknown

**Throws:**
MessagingException

**See Also:**
Part.ATTACHMENT, Part.INLINE, Part.getFileName()

---

```java
public void setDisposition(String disposition) throws MessagingException

Message "Content-Disposition"
disposition
null "Content-Disposition"

Throws
IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:
```

**setDisposition**

```java
public void setDisposition(String disposition) throws MessagingException

Set the "Content-Disposition" header field of this Message. If disposition is null, any existing "Content-Disposition" header field is removed.

**Specified by:**
setDisposition in interface Part

**Parameters:**
disposition - disposition of this part

**Throws:**
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

See Also:
Part.ATTACHMENT, Part.INLINE,
Part.setFileName(java.lang.String)

public String getEncoding() throws MessagingException
"Content-Transfer-Encoding"

    getHeader
    return

Throws
MessagingException:

getEncoding

public String getEncoding() throws MessagingException

    Returns the content transfer encoding from the "Content-Transfer-Encoding" header field. Returns null if the header is unavailable or its value is absent.

    This implementation uses the getHeader method to obtain the requisite header field.

    Specified by:
getEncoding in interface MimePart

Returns:
content-transfer-encoding

Throws:
MessagingException

public String getContentID() throws MessagingException
"Content-ID"

null
**getHeader**

return ID

Throws MessagingException:

---

**getContentID**

public String getContentID() throws MessagingException

Returns the value of the "Content-ID" header field. Returns null if the field is unavailable or its value is absent.

This implementation uses the getHeader method to obtain the requisite header field.

**Specified by:**

getContentID in interface MimePart

**Returns:**

content-ID

**Throws:**

MessagingException

---

**public void setContentID(String cid) throws MessagingException**

**Message** "Content-ID" cid null "Content-ID"

**Throws**

IllegalWriteException:

Throws IllegalStateException: READ_ONLY

Throws MessagingException:

---

**setContentID**

public void setContentID(String cid) throws MessagingException

Throws MessagingException
Set the "Content-ID" header field of this Message. If the cid parameter is null, any existing "Content-ID" is removed.

Throws:
- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this message is obtained from a READ_ONLY folder.
- `MessagingException`

```java
public String getContentMD5() throws MessagingException {
    return MD5;
}
```

`getContentMD5`

Returns the value of the "Content-MD5" header field. Returns `null` if this field is unavailable or its value is absent.

This implementation uses the `getHeader` method to obtain the requisite header field.

Specified by:
- `getContentMD5` in interface `MimePart`

Returns: content-MD5

Throws: `MessagingException`
public void setContentMD5(String md5) throws MessagingException

Message "Content-MD5"

Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:

setContentMD5

public void setContentMD5(String md5)
throws MessagingException

Set the "Content-MD5" header field of this Message.

Specified by:
setContentMD5 in interface MimePart

Parameters:
md5 - the MD5 value

Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public String getDescription() throws MessagingException

Message "Content-Description" null

Content-Description RFC 2047 Unicode

getHeader
return
Throws MessagingException:
getDescription

public String getDescription() throws MessagingException

    Returns the "Content-Description" header field of this Message. This typically associates some descriptive information with this part. Returns null if this field is unavailable or its value is absent.

    If the Content-Description field is encoded as per RFC 2047, it is decoded and converted into Unicode. If the decoding or conversion fails, the raw data is returned as-is

    This implementation uses the getHeader method to obtain the requisite header field.

Specified by: getDescription in interface Part

Returns: content-description

Throws: MessagingException

public void setDescription(String description) throws MessagingException

    Message "Content-Description" description
    null "Content-Description"

    description US-ASCII description US-ASCII

    MessagingException MessagingException UnsupportedEncodingException

    description

    Throws IllegalWriteException:
setDescription

public void setDescription(String description) throws MessagingException

Set the "Content-Description" header field for this Message. If the description parameter is null, then any existing "Content-Description" fields are removed.

If the description contains non US-ASCII characters, it will be encoded using the platform's default charset. If the description contains only US-ASCII characters, no encoding is done and it is used as-is.

Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.

Specified by:
    setDescription in interface Part
Parameters:
    description - content-description
Throws:
    IllegalWriteException - if the underlying implementation does not support modification
    IllegalStateException - if this message is obtained from a READ_ONLY folder.
    MessagingException. - An UnsupportedEncodingException may be included in the exception chain if the charset conversion fails.
    MessagingException
public void setDescription(String description, String charset) throws MessagingException

Message "Content-Description" description null "Content-Description"

description US-ASCII description US-ASCII

MessagingException MessagingException UnsupportedEncodingException

Throws IllegalWriteException:

Throws IllegalStateException: READ_ONLY

Throws MessagingException:

Throws UnsupportedEncodingException

setDescription

public void setDescription(String description, String charset)

throws MessagingException

Set the "Content-Description" header field for this Message. If the description parameter is null, then any existing "Content-Description" fields are removed.

If the description contains non US-ASCII characters, it will be encoded using the specified charset. If the description contains only US-ASCII characters, no encoding is done and it is used as-is.

Note that if the charset encoding process fails, a MessagingException is thrown, and an UnsupportedEncodingException is included in the chain of nested exceptions within the MessagingException.
public String[] getContentLanguage() throws MessagingException

Message "Content-Language" Content-Language
RFC 1766 null

getHeader
return Content-Language
Throws MessagingException:

getContentTypeLanguage

public String[] getContentLanguage() throws MessagingException

Get the languages specified in the "Content-Language" header field of this message. The Content-Language header is defined by RFC 1766. Returns null if this field is unavailable or its value is absent.

This implementation uses the getHeader method to obtain the requisite header field.

Specified by:
getContentLanguage in interface MimePart
Returns:
value of content-language header.

Throws:
MessagingException

public void setContentLanguage(String[] languages)
throws MessagingException
MimePart "Content-Language" Content-Language
RFC 1766

languages

Throws
IllegalWriteException:
Throws
IllegalStateException: READ_ONLY
Throws
MessagingException:

setContentLanguage

public void setContentLanguage(String[] languages)
throws MessagingException

Set the "Content-Language" header of this MimePart. The Content-Language header is defined by RFC 1766.

Specified by:
setContentLanguage in interface MimePart

Parameters:
languages - array of language tags

Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public String getMessageID() throws MessagingException
"Message-ID" null
getMessageID

public String getMessageID() throws MessagingException

Returns the value of the "Message-ID" header field. Returns null if this field is unavailable or its value is absent.

The default implementation provided here uses the getHeader method to return the value of the "Message-ID" field.

Returns:
Message-ID

Throws:
MessagingException - if the retrieval of this field causes any exception.

Since:
JavaMail 1.1

See Also:
javax.mail.search.MessageIDTerm

public String getFileName() throws MessagingException

"Content-Disposition" "filename" BodyPart "Content-Type" "name"

mail.mime.encodefilename true
getFileName

public String getFileName() throws MessagingException

Get the filename associated with this Message.

Returns the value of the "filename" parameter from the "Content-Disposition" header field of this message. If it's not available, returns the value of the "name" parameter from the "Content-Type" header field of this BodyPart. Returns null if both are absent.

If the mail.mime.encodefilename System property is set to true, the MimeUtility.decodeText method will be used to decode the filename. While such encoding is not supported by the MIME spec, many mailers use this technique to support non-ASCII characters in filenames. The default value of this property is false.

Specified by:
getFileName in interface Part

Returns:
filename

Throws:
MessagingException

public void setFileName(String filename) throws MessagingException

Message "Content-Disposition" "filename"
mail.mime.encodefilename true
MimeUtility.encodeText MIME ASCII
false

Throws
IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:

setFileName

public void setFileName(String filename)
throws MessagingException

Set the filename associated with this part, if possible.

Sets the "filename" parameter of the "Content-Disposition" header field of this message.

If the mail.mime.encodefilename System property is set to true, the
MimeUtility.encodeText method will be used to encode the filename.
While such encoding is not supported by the MIME spec, many
mailers use this technique to support non-ASCII characters in
filenames. The default value of this property is false.

Specified by:
setFileName in interface Part

Parameters:
filename - Filename to associate with this part

Throws:
IllegalWriteException - if the underlying implementation does
not support modification
IllegalStateException - if this message is obtained from a
READ_ONLY folder.
MessagingException

public java.io.InputStream getInputStream() throws
java.io.IOException, MessagingException
Message "content"

DataHandler getDataHandler().getInputStream()
return InputStream
Throws MessagingException:
Throws java.io.IOException: DataHandler javax.activation.DataHandler
See also getContentStream, getInputStream

getInputStream

public InputStream getInputStream()
throws IOException, MessagingException

Return a decoded input stream for this Message's "content".

This implementation obtains the input stream from the DataHandler, that is, it invokes getDataHandler().getInputStream().

Specified by:
getInputStream in interface Part
Returns:
an InputStream
Throw:
MessagingException
IOException - this is typically thrown by the DataHandler. Refer to the documentation for javax.activation.DataHandler for more details.
See Also:
getContentStream(), DataHandler.getInputStream()

protected java.io.InputStream getContentStream() throws
MessagingException
DataHandler

getContentStream

protected InputStream getContentStream() throws MessagingException

Produce the raw bytes of the content. This method is used during parsing, to create a DataHandler object for the content. Subclasses that can provide a separate input stream for just the message content might want to override this method.

This implementation returns a SharedInputStream, if contentStream is not null. Otherwise, it returns a ByteArrayInputStream constructed out of the content byte array.

Throws:
   MessagingException
See Also:
   content
getRawInputStream

public InputStream getRawInputStream() throws MessagingException

Return an InputStream to the raw data with any Content-Transfer-Encoding intact. This method is useful if the "Content-Transfer-Encoding" header is incorrect or corrupt, which would prevent the getInputStream method or getContent method from returning the correct data. In such a case the application may use this method and attempt to decode the raw data itself.

This implementation simply calls the getContentStream method.

Throws:
   MessagingException

Since:
   JavaMail 1.2

See Also:
   getInputStream(), getContentStream()

public DataHandler getDataHandler() throws MessagingException

Message DataHandler

g getContentStream

dataHandler() {
   if (dh == null) {
      dh = new DataHandler(new MimePartDataSource(this));
   }
   return dh;
}
Throws **MessagingException**: 

**getDataHandler**

```java
class MimePartDataSource implements DataSource {
    public getInputStream() {
        return MimeUtility.decode(
            getContentStream(), getEncoding());
    }
    ....
}
```

```java
public DataHandler getDataHandler() throws MessagingException
```

Return a DataHandler for this Message's content.

The implementation provided here works as follows. Note the use of the `getContentStream` method to generate the byte stream for the content. Also note that any transfer-decoding is done automatically within this method.

```java
gDataHandler() {
    if (dh == null) {
        dh = new DataHandler(new MimePartDataSource(this));
    }
    return dh;
}
```

```java
class MimePartDataSource implements DataSource {
    public getInputStream() {
        return MimeUtility.decode(
            getContentStream(), getEncoding());
    }
    ....
}
```
Specified by:
getDatasHandler in interface Part

Returns:
DataHandler for the content

Throws:
MessagingException

public Object getContent() throws java.io.IOException, MessagingException,
java.io.IOException:
java.activation.DataHandler
javax.mail.Part, getContent

getContents

public Object getContent() throws IOException,
MessagingException

Return the content as a Java object. The type of this object is
dependent on the content itself. For example, the native format of a
"text/plain" content is usually a String object. The native format for a
"multipart" message is always a Multipart subclass. For content
types that are unknown to the DataHandler system, an input stream
is returned as the content.
This implementation obtains the content from the DataHandler, that is, it invokes `getDataHandler().getContent()`. If the content is a Multipart or Message object and was created by parsing a stream, the object is cached and returned in subsequent calls so that modifications to the content will not be lost.

**Specified by:**
`getContent` in interface `Part`

**Returns:**
Object

**Throws:**
`MessagingException`
`IOException` - this is typically thrown by the DataHandler. Refer to the documentation for `javax.activation.DataHandler` for more details.

**See Also:**
`Part`, `DataHandler.getContent()`

---

```java
public void setDataHandler(DataHandler dh) throws MessagingException
```

**DataHandler**

- `dh` DataHandler
- **Throws**
  - `IllegalWriteException`
  - `IllegalStateException`: `READ_ONLY`
  - `MessagingException`

**setDataHandler**

```java
public void setDataHandler(DataHandler dh)
```

This method provides the mechanism to set this part's content. The given DataHandler object should wrap the actual content.

**Specified by:**
`setDataHandler` in interface `Part`
Parameters:

dh - The DataHandler for the content.

Throws:

- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this message is obtained from a READONLY folder.
- `MessagingException`

```java
public void setContent(Object o, String type) throws MessagingException
```

**Message**

**DataHandler**  **DataContentHandler**

**JavaMail**  ```java
setContent(foobar, "application/x-foobar")
```

**"application/x-foobar"**  **DataContentHandler**  **Java Activation Framework**

```java
type
Mime
```

Throws:

- `IllegalWriteException`
- `IllegalStateException`: READONLY
- `MessagingException`

**setContent**

```java
public void setContent(Object o, String type)
throws MessagingException
```

**A convenience method for setting this Message's content.**

The content is wrapped in a DataHandler object. Note that a DataContentHandler class for the specified type should be available to the JavaMail implementation for this to work right. i.e., to do `setContent(foobar, "application/x-foobar")`, a
DataContentHandler for "application/x-foobar" should be installed. Refer to the Java Activation Framework for more information.

**Specified by:**
```
setContent in interface Part
```

**Parameters:**
- `o` - the content object
- `type` - Mime type of the object

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this message is obtained from a READ_ONLY folder.
- `MessagingException`

---

```java
public void setText(String text) throws MessagingException

MIME "text/plain" String string US-ASCII "charset"

text

charset

setText

Throws: MessagingException:

See also

setText(String text, String charset)
```

---

**setText**

```java
public void setText(String text)

throws MessagingException
```

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain". If the string contains non US-ASCII characters, it will be encoded using the platform's default charset.
The charset is also used to set the "charset" parameter.

Note that there may be a performance penalty if text is large, since this method may have to scan all the characters to determine what charset to use.

If the charset is already known, use the setText method that takes the charset parameter.

**Specified by:**
- `setText` in interface `MimePart`
- `setText` in interface `Part`

**Parameters:**
- `text` - the text content to set

**Throws:**
- `MessagingException` - if an error occurs

**See Also:**
- `setText(String text, String charset)`

```
public void setText(String text, String charset) throws MessagingException

MIME "text/plain" String Unicode "charset"

  text
  charset

Throws       MessagingException:
```

### setText

```
public void setText(String text, String charset) throws MessagingException

MIME "text/plain" String Unicode "charset"

  text
  charset

Throws       MessagingException:
```

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain" and the specified charset. The given Unicode string will be charset-encoded using the specified charset.
The charset is also used to set the "charset" parameter.

**Specified by:**
```
setText in interface MimePart
```

**Parameters:**
- `text` - the text content to set
- `charset` - the charset to use for the text

**Throws:**
```
MessagingException - if an error occurs
```

```java
public void setText(String text, String charset, String subtype)
throws MessagingException
```

MIME "text" MIME String Unicode "charset"
```
  text
  charset
  subtype MIME "html"
```

**Throws**
```
MessagingException:
```

```
since JavaMail 1.4
```
charset - the charset to use for the text
subtype - the MIME subtype to use (e.g., "html")

Throws:
MessagingException - if an error occurs

Since:
JavaMail 1.4

public void setContent(Multipart mp) throws MessagingException

Message Multipart

ToFile multipart

Throws
IllegalWriteException:

Throws
IllegalStateException: READ_ONLY

Throws
MessagingException:

setContent

public void setContent(Multipart mp) throws MessagingException

This method sets the Message's content to a Multipart object.

Specified by:
setContent in interface Part

Parameters:
mp - The multipart object that is the Message's content

Throws:
IllegalWriteException - if the underlying implementation does not support modification of existing values
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public Message reply(boolean replyToAll) throws MessagingException

Message
replyToAll Message getReplyTo

"Subject" "Re:" "Re:" "Message-Id"
"In-Reply-To" ANSWERED "References"
"References" "In-Reply-To" "Message-Id" RFC 2822

replyToAll

return Message

Throws MessagingException:

reply

public Message reply(boolean replyToAll)

throws MessagingException

Get a new Message suitable for a reply to this message. The new Message will have its attributes and headers set up appropriately. Note that this new message object will be empty, i.e., it will not have a "content". These will have to be suitably filled in by the client.

If replyToAll is set, the new Message will be addressed to all recipients of this message. Otherwise, the reply will be addressed to only the sender of this message (using the value of the getReplyTo method).

The "Subject" field is filled in with the original subject prefixed with "Re:" (unless it already starts with "Re:"). The "In-Reply-To" header is set in the new message if this message has a "Message-Id" header. The ANSWERED flag is set in this message. The current implementation also sets the "References" header in the new message to include the contents of the "References" header (or, if missing, the "In-Reply-To" header) in this message, plus the contents of the "Message-Id" header of this message, as described in RFC 2822.
Specified by:
    reply in class Message
Parameters:
    replyToAll - reply should be sent to all recipients of this message
Returns:
    the reply Message
Throws:
    MessagingException

public void writeTo(java.io.OutputStream os) throws java.io.IOException, MessagingException
RFC 822
FilterOutputStream FilterOutputStream MIME Internet CRLF

    writeTo(OutputStream, String[]) null
    Throws java.io.IOException: javax.activation
    Throws MessagingException: writeTo
See also

writeTo

public void writeTo(OutputStream os)
    throws IOException, MessagingException

Output the message as an RFC 822 format stream.

Note that, depending on how the messag was constructed, it may use a variety of line termination conventions. Generally the output should be sent through an appropriate FilterOutputStream that converts the line terminators to the desired form, either CRLF for MIME compatibility and for use in Internet protocols, or the local
platform's line terminator for storage in a local text file.

This implementation calls the `writeTo(OutputStream, String[])` method with a null ignore list.

**Specified by:**
- `writeTo` in interface `Part`

**Throws:**
- `IOException` - if an error occurs writing to the stream or if an error is generated by the `javax.activation` layer.
- `MessagingException`

**See Also:**
- `DataHandler.writeTo(java.io.OutputStream)`

---

```java
public void writeTo(java.io.OutputStream os, String[] ignoreList) throws java.io.IOException, javax.activation.MessagingException, MessagingException
```

Output the message as an RFC 822 format stream, without specified headers. If the `saved` flag is not set, the `saveChanges` method is called. If the `modified` flag is not set and the `content` array is not null, the `content` array is written directly, after writing the appropriate message headers.

**Throws:**
public String[] getHeader(String name) throws MessagingException

header_name US-ASCII RFC 2047

headers InternetHeaders

name return

Throws MessagingException:

See also javax.mail.internet.MimeUtility

getHeader

public String[] getHeader(String name)

throws MessagingException

Get all the headers for this header_name. Note that certain headers may be encoded as per RFC 2047 if they contain non US-ASCII characters and these should be decoded.

This implementation obtains the headers from the headers InternetHeaders object.

Specified by:
getHeader in interface Part

Parameters:
name - name of header

Returns:
array of headers

Throws:
MessagingException
public String getHeader(String name, String delimiter) throws MessagingException

String delimiter

name
delimiter

return

Throws MessagingException:

getHeader

public String getHeader(String name, String delimiter) throws MessagingException

Get all the headers for this header name, returned as a single String, with headers separated by the delimiter. If the delimiter is null, only the first header is returned.

Specified by: getHeader in interface MimePart

Parameters:
  name - the name of this header
delimiter - separator between values

Returns:
  the value fields for all headers with this name

Throws:
  MessagingException

public void setHeader(String name, String value) throws MessagingException

header_name RFC 822  US-ASCII
US-ASCII  RFC 2047
Throws IllegalArgumentException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:
See also javax.mail.internet.MimeUtility

**setHeader**

```java
public void setHeader(String name, String value)
  throws MessagingException
```

Set the value for this header_name. Replaces all existing header values with this new value. Note that RFC 822 headers must contain only US-ASCII characters, so a header that contains non US-ASCII characters must have been encoded by the caller as per the rules of RFC 2047.

**Specified by:**
setHeader in interface Part

**Parameters:**
- name - header name
- value - header value

**Throws:**
- IllegalArgumentException - if the underlying implementation does not support modification
- IllegalStateException - if this message is obtained from a READ_ONLY folder.
- MessagingException

**See Also:**
MimeUtility

---

```java
public void addHeader(String name, String value) throws MessagingException
```

header_name RFC 822 US-ASCII US-
**addHeader**

```java
class Part {

    public void addHeader(String name, String value) throws MessagingException {
        // Implementation
    }
}
```

Add this value to the existing values for this header name. Note that RFC 822 headers must contain only US-ASCII characters, so a header that contains non US-ASCII characters must have been encoded as per the rules of RFC 2047.

**Specified by:**
- `addHeader` in interface `Part`

**Parameters:**
- `name` - header name
- `value` - header value

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this message is obtained from a READ_ONLY folder.
- `MessagingException` - if this message is obtained from a READ_ONLY folder.

**See Also:**
- `javax.mail.internet.MimeUtility`

---

```java
class Part {

    public void removeHeader(String name) throws MessagingException {
        // Implementation
    }
}
```
Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:

removeHeader

public void removeHeader(String name)
throws MessagingException

Remove all headers with this name.

Specified by:
removeHeader in interface Part
Parameters:
name - the name of this header
Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

public java.utilEnumeration{} getAllHeaders() throws MessagingException
Header Message

US-ASCII RFC 2047

    headers InternetHeaders
    return
    Throws MessagingException:
    See also javax.mail.internet.MimeUtility

getAllHeaders
public Enumeration getAllHeaders() throws MessagingException

Return all the headers from this Message as an enumeration of Header objects.

Note that certain headers may be encoded as per RFC 2047 if they contain non US-ASCII characters and these should be decoded.

This implementation obtains the headers from the headers InternetHeaders object.

Specified by:
getAllHeaders in interface Part

Returns:
array of header objects

Throws:
MessagingException

See Also:
MimeUtility

---

public java.util Enumeration<E> getMatchingHeaders(String[] names) throws MessagingException

Return matching headers from this Message as an Enumeration of Header objects. This implementation obtains the headers from the headers InternetHeaders object.

Specified by:
getMatchingHeaders in interface Part

Returns:
enumeration of Header objects

Throws:
MessagingException

public java.util.Enumeration<E>
getNonMatchingHeaders(String[] names) throws MessagingException

Header Message

Throws
MessagingException

getNonMatchingHeaders

public Enumeration getNonMatchingHeaders(String[] names) throws MessagingException

Return non-matching headers from this Message as an Enumeration of Header objects. This implementation obtains the header from the headers InternetHeaders object.

Specified by:
getNonMatchingHeaders in interface Part

Returns:
enumeration of Header objects

Throws:
MessagingException

public void addHeaderLine(String line) throws MessagingException

RFC 822

Throws
IllegalWriteException:

Throws
IllegalStateException: READ_ONLY

Throws
MessagingException:
addHeaderLine

public void addHeaderLine(String line)
    throws MessagingException

Add a raw RFC 822 header-line.

Specified by: 
    addHeaderLine in interface MimePart

Throws:
    IllegalWriteException - if the underlying implementation does not support modification
    IllegalStateException - if this message is obtained from a READ_ONLY folder.
    MessagingException

public java.util.Enumeration&lt;E&gt; getAllHeaderLines()
throws MessagingException
String Header  RFC 822  "name"  "value"

Throws: 
    MessagingException: 

getAllHeaderLines

public Enumeration getAllHeaderLines()
    throws MessagingException

Get all header lines as an Enumeration of Strings. A Header line is a raw RFC 822 header-line, containing both the "name" and "value" field.

Specified by: 
    getAllHeaderLines in interface MimePart

Throws: 
    MessagingException
public java.util.Enumeration<E> getMatchingHeaderLines(String[] names) throws MessagingException
String Header  RFC 822  "name"  "value"

Throws  

MessagingException:

getMatchingHeaderLines

public Enumeration getMatchingHeaderLines(String[] names)
throws MessagingException

Get matching header lines as an Enumeration of Strings. A Header line is a raw RFC 822 header-line, containing both the "name" and "value" field.

Specified by:
    getMatchingHeaderLines in interface MimePart

Throws:
    MessagingException

public java.util.Enumeration<E> getNonMatchingHeaderLines(String[] names) throws MessagingException
String Header  RFC 822  "name"  "value"

Throws  

MessagingException:

getNonMatchingHeaderLines

public Enumeration getNonMatchingHeaderLines(String[] names)
throws MessagingException

Get non-matching header lines as an Enumeration of Strings. A
Header line is a raw RFC 822 header-line, containing both the "name" and "value" field.

**Specified by:**
getNonMatchingHeaderLines in interface MimePart

**Throws:**
MessagingException

```java
public Flags getFlags() throws MessagingException

Flags
```

**Flags Flags**

```java
return Flags
Throws MessagingException:
See also javax.mail.Flags
```

getFlags

```java
public Flags getFlags() throws MessagingException

Return a Flags object containing the flags for this message.

Note that a clone of the internal Flags object is returned, so modifying the returned Flags object will not affect the flags of this message.

**Specified by:**
getFlags in class Message

**Returns:**
Flags object containing the flags for this message

**Throws:**
MessagingException

**See Also:**
Flags
public boolean isSet(Flags.Flag flag) throws MessagingException

flag

flag

return

Throws: MessagingException:
See javax.mail.Flags.Flag, ANSWERED, DELETED, DRAFT,
also FLAGGED, RECENT, SEEN

isSet

public boolean isSet(Flags.Flag flag)
throws MessagingException

Check whether the flag specified in the flag argument is set in this message.

This implementation checks this message's internal flags object.

Overrides:
iset in class Message

Parameters:
flag - the flag

Returns:
value of the specified flag for this message

Throws: MessagingException

See Also:
Flags.Flag, Flags.Flag.ANSWERED, Flags.Flag.DELETED,
Flags.Flag.DRAFT, Flags.Flag.FLAGGED, Flags.Flag.RECENT,
Flags.Flag.SEEN

public void setFlags(Flags flag, boolean set) throws
MessagingException

flags

Throws IllegalWriteException:
Throws IllegalStateException: READ_ONLY
Throws MessagingException:

setFlags

public void setFlags(Flags flag, boolean set)
throws MessagingException

Set the flags for this message.

This implementation modifies the flags field.

Specified by:
setFlags in class Message

Parameters:
flag - Flags object containing the flags to be set
set - the value to be set

Throws:
IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this message is obtained from a READ_ONLY folder.
MessagingException

See Also:
MessageChangedEvent

public void saveChanges() throws MessagingException
saveChanges

**READ ONLY**  saveChanges

| modified | true | save | true | updateHeaders |

**Throws**

- `IllegalWriteException`
- `IllegalStateException`: READ_ONLY
- `MessagingException`:

**saveChanges**

```java
public void saveChanges()
    throws MessagingException
```

Updates the appropriate header fields of this message to be consistent with the message's contents. If this message is contained in a Folder, any changes made to this message are committed to the containing folder.

If any part of a message's headers or contents are changed, `saveChanges` must be called to ensure that those changes are permanent. Otherwise, any such modifications may or may not be saved, depending on the folder implementation.

Messages obtained from folders opened READ_ONLY should not be modified and saveChanges should not be called on such messages.

This method sets the modified flag to true, the save flag to true, and then calls the updateHeaders method.

**Specified by:**

`saveChanges` in class `Message`

**Throws:**

- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this message is obtained from a READ_ONLY folder.
protected void updateMessageID() throws MessagingException

**updateMessageID**

protected void updateMessageID()

throws MessagingException

Update the Message-ID header. This method is called by the updateHeaders and allows a subclass to override only the algorithm for choosing a Message-ID.

**Throws:**

- **MessagingException**

**Since:**

JavaMail 1.4

protected void updateHeaders() throws MessagingException

**updateHeaders**

protected void updateHeaders()

throws MessagingException

Throws **IllegalWriteException:**

Throws **IllegalStateException:** READ_ONLY

Throws **MessagingException:**

protected void updateHeaders()
Called by the `saveChanges` method to actually update the MIME headers. The implementation here sets the `Content-Transfer-Encoding` header (if needed and not already set), the `MIME-Version` header and the `Message-ID` header. Also, if the content of this message is a `MimeMultipart`, its `updateHeaders` method is called.

**Throws:**

- `IllegalArgumentException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this message is obtained from a `READ_ONLY` folder.
- `MessagingException`
is - the InputStream to read the headers from

**Throws:**
MessagingException

**Since:**
JavaMail 1.2

---

**protected** **MimeMessage** createMimeMessage**(Session session)** throws **MessagingException**

MimeMessage  MimeMessage
MimeMessage  Session  MimeMessage
session  Session
return  MimeMessage
since  JavaMail 1.4

**createMimeMessage**

**protected** **MimeMessage** createMimeMessage**(Session session)** throws **MessagingException**

Create and return a MimeMessage object. The reply method uses this method to create the MimeMessage object that it will return. Subclasses can override this method to return a subclass of MimeMessage. This implementation simply constructs and returns a MimeMessage object using the supplied Session.

**Parameters:**

session - the Session to use for the new message

**Returns:**

the new MimeMessage object

**Throws:**

MessagingException

**Since:**
JavaMail 1.4
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail.internet  Class MimeMessage.RecipientType

java.lang.Object
   ▼ javax.mail.Message.RecipientType
      ▼ javax.mail.internet.MimeMessage.RecipientType

All Implemented Interfaces:
   Serializable

Enclosing class:
   MimeMessage

public static class MimeMessage.RecipientType
extends Message.RecipientType

Extends: Message.RecipientType
Contained within: MimeMessage

javax.mail.Message.RecipientType RecipientType
RecipientType NEWSGROUPS

See also  javax.mail.Message.RecipientType

This inner class extends the javax.mail.Message.RecipientType class to add additional RecipientTypes. The one additional RecipientType currently defined here is NEWSGROUPS.

See Also:
   Message.RecipientType, Serialized Form

---

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static MimeMessage.RecipientType NEWSGROUPS The &quot;Newsgroup&quot; (Usenet news) recipients.</td>
</tr>
</tbody>
</table>
Fields inherited from class javax.mail.Message.RecipientType
BCC, CC, TO, type

Constructor Summary
protected MimeMessage.RecipientType(String type)

Method Summary
protected Object readResolve()
When deserializing a RecipientType, we need to make sure to return only one of the known static final instances defined in this class.

Methods inherited from class javax.mail.Message.RecipientType
toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

NEWSGROUPS

public static final MimeMessage.RecipientType NEWSGROUPS

The "Newsgroup" (Usenet news) recipients.

Constructor Detail
protected MimeMessage.RecipientType(String type)

MimeMessage.RecipientType

protected MimeMessage.RecipientType(String type)

Method Detail

protected Object readResolve() throws java.io.ObjectStreamException

readResolve

protected Object readResolve() throws ObjectStreamException

Description copied from class: Message.RecipientType
When deserializing a RecipientType, we need to make sure to return only one of the known static final instances defined in this class. Subclasses must implement their own readResolve method that checks for their known instances before calling this super method.

Overrides:
readResolve in class Message.RecipientType

Throws:
ObjectStreamException

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
The MimeMultipart class is an implementation of the abstract Multipart class that uses MIME conventions for the multipart data.
A MimeMultipart is obtained from a MimePart whose primary type is "multipart" (by invoking the part's `getContentType()` method) or it can be created by a client as part of creating a new MimeMessage.

The default multipart subtype is "mixed". The other multipart subtypes, such as "alternative", "related", and so on, can be implemented as subclasses of MimeMultipart with additional methods to implement the additional semantics of that type of multipart content. The intent is that service providers, mail JavaBean writers and mail clients will write many such subclasses and their Command Beans, and will install them into the JavaBeans Activation Framework, so that any JavaMail implementation and its clients can transparently find and use these classes. Thus, a MIME multipart handler is treated just like any other type handler, thereby decoupling the process of providing multipart handlers from the JavaMail API. Lacking these additional MimeMultipart subclasses, all subtypes of MIME multipart data appear as MimeMultipart objects.

An application can directly construct a MIME multipart object of any subtype by using the `MimeMultipart(String subtype)` constructor. For example, to create a "multipart/alternative" object, use `new MimeMultipart("alternative")`.

The `mail.mime.multipart.ignoreMissingEndBoundary` property may be set to `false` to cause a MessagingException to be thrown if the multipart data does not end with the required end boundary line. If this property is set to `true` or not set, missing end boundaries are not considered an error and the final body part ends at the end of the data.

The `mail.mime.multipart.ignoreMissingBoundaryParameter` System property may be set to `false` to cause a MessagingException to be thrown if the Content-Type of the MimeMultipart does not include a `boundary` parameter. If this property is set to `true` or not set, the multipart parsing code will look for a line that looks like a boundary line and use that as the boundary separating the parts.

**Version:**

1.48, 07/05/15

**Author:**

John Mani, Bill Shannon, Max Spivak
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected ds</td>
<td>DataSource</td>
<td>The DataSource supplying our InputStream.</td>
</tr>
<tr>
<td>protected parsed</td>
<td>boolean</td>
<td>Have we parsed the data from our InputStream yet?</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.mail.Multipart

- contentType
- parent
- parts

### Constructor Summary

- **MimeMultipart()**
  
  Default constructor.

- **MimeMultipart(DataSource ds)**
  
  Constructs a MimeMultipart object and its bodyparts from the given DataSource.

- **MimeMultipart(String subtype)**
  
  Construct a MimeMultipart object of the given subtype.

### Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>void addBodyPart(BodyPart part)</td>
<td>Adds a Part to the multipart.</td>
</tr>
<tr>
<td></td>
<td>void addBodyPart(BodyPart part, int index)</td>
<td>Adds a BodyPart at position index.</td>
</tr>
<tr>
<td>protected</td>
<td>createInternetHeaders(InputStream is)</td>
<td>Create and return an InternetHeaders object that loads the headers from the given InputStream.</td>
</tr>
<tr>
<td>protected</td>
<td>createMimeBodyPart(InputStream is)</td>
<td>Create and return a MimeBodyPart object to represent a body part parsed from the InputStream.</td>
</tr>
<tr>
<td>protected</td>
<td>createMimeBodyPart(InternetHeaders headers, byte[] content)</td>
<td>Create and return a MimeBodyPart object to</td>
</tr>
</tbody>
</table>
represent a body part parsed from the InputStream.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getBodyPart(int index)</code></td>
<td>Get the specified BodyPart.</td>
</tr>
<tr>
<td><code>getBodyPart(String CID)</code></td>
<td>Get the MimeBodyPart referred to by the given ContentID (CID).</td>
</tr>
<tr>
<td><code>getCount()</code></td>
<td>Return the number of enclosed BodyPart objects.</td>
</tr>
<tr>
<td><code>getPreamble()</code></td>
<td>Get the preamble text, if any, that appears before the first body part of this multipart.</td>
</tr>
<tr>
<td><code>isComplete()</code></td>
<td>Return true if the final boundary line for this multipart was seen.</td>
</tr>
<tr>
<td><code>parse()</code></td>
<td>Parse the InputStream from our DataSource, constructing the appropriate MimeBodyParts.</td>
</tr>
<tr>
<td><code>removeBodyPart(BodyPart part)</code></td>
<td>Remove the specified part from the multipart message.</td>
</tr>
<tr>
<td><code>removeBodyPart(int index)</code></td>
<td>Remove the part at specified location (starting from 0).</td>
</tr>
<tr>
<td><code>setPreamble(String preamble)</code></td>
<td>Set the preamble text to be included before the first body part.</td>
</tr>
<tr>
<td><code>setSubType(String subtype)</code></td>
<td>Set the subtype.</td>
</tr>
<tr>
<td><code>updateHeaders()</code></td>
<td>Update headers.</td>
</tr>
<tr>
<td><code>writeTo(OutputStream os)</code></td>
<td>Iterates through all the parts and outputs each MIME part separated by a boundary.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.Multipart
Methods inherited from class java.lang.\texttt{Object}
\texttt{clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait}

Field Detail

\verb|ds|

\texttt{protected DataSource ds}

The DataSource supplying our InputStream.

\verb|parsed|

\texttt{protected boolean parsed}

Have we parsed the data from our InputStream yet? Defaults to true; set to false when our constructor is given a DataSource with an InputStream that we need to parse.

Constructor Detail

\texttt{public MimeMultipart()}

\texttt{MimeMultipart "multipart/mixed" contentType "boundary"}

\texttt{MimeBodyParts}
MimeMultipart

public MimeMultipart()

Default constructor. An empty MimeMultipart object is created. Its content type is set to "multipart/mixed". A unique boundary string is generated and this string is setup as the "boundary" parameter for the contentType field.

MimeBodyParts may be added later.

public MimeMultipart(String subtype)
MimeMultipart

MimeBodyPart

public MimeMultipart(String subtype)

Construct a MimeMultipart object of the given subtype. A unique boundary string is generated and this string is setup as the "boundary" parameter for the contentType field.

MimeBodyParts may be added later.

public MimeMultipart(DataSource ds) throws MessagingException
DataSource MimeMultipart bodypart

DataSource MultipartDataSource MultipartDataSource

MultipartDataSource Multipart
MimeMultipart

public MimeMultipart(DataSource ds) throws MessagingException

Constructs a MimeMultipart object and its bodyparts from the given DataSource.

This constructor handles as a special case the situation where the given DataSource is a MultipartDataSource object. In this case, this method just invokes the superclass (i.e., Multipart) constructor that takes a MultipartDataSource object.

Otherwise, the DataSource is assumed to provide a MIME multipart byte stream. The parsed flag is set to false. When the data for the body parts are needed, the parser extracts the "boundary" parameter from the content type of this DataSource, skips the 'preamble' and reads bytes till the terminating boundary and creates MimeBodyParts for each part of the stream.

Parameters:
  ds - DataSource, can be a MultipartDataSource

Throws:
  MessagingException

Method Detail

public void setSubType(String subtype) throws MessagingException

MimeMultipart  multipart  "mixed"
**setSubType**

```java
public void setSubType(String subtype) throws MessagingException
```

Set the subtype. This method should be invoked only on a new MimeMultipart object created by the client. The default subtype of such a multipart object is "mixed".

**Parameters:**
- `subtype` - Subtype

**Throws:**
- `MessagingException`

**getCount**

```java
public int getCount() throws MessagingException
```

BodyPart

```java
return
```

Return the number of enclosed BodyPart objects.

**Overrides:**
- `getCount` in class `Multipart`

**Returns:**
- number of parts

**Throws:**
- `MessagingException`

**See Also:**
- `Multipart.parts`
public BodyPart getBodyPart(int index) throws MessagingException

Get the specified BodyPart. BodyParts are numbered starting at 0.

Overrides: 
getBodyPart in class Multipart

Parameters:
index - the index of the desired BodyPart

Returns:
the Part

Throws:
MessagingException - if no such BodyPart exists

public BodyPart getBodyPart(String CID) throws MessagingException

ContentID (CID) MimeBodyPart null

return

getBodyPart

public BodyPart getBodyPart(String CID) throws MessagingException
Get the MimeBodyPart referred to by the given ContentID (CID). Returns null if the part is not found.

**Parameters:**
- CID - the ContentID of the desired part

**Returns:**
- the Part

**Throws:**
- MessagingException

---

```java
public boolean removeBodyPart(BodyPart part) throws MessagingException, IllegalWriteException
```

**multipart**

```
part
return part true false
```

**Throws**:
- MessagingException: Part
- IllegalWriteException:

---

**removeBodyPart**

```java
public boolean removeBodyPart(BodyPart part) throws MessagingException
```

Remove the specified part from the multipart message. Shifts all the parts after the removed part down one.

**Overrides:**
- removeBodyPart in class Multipart

**Parameters:**
- part - The part to remove

**Returns:**
- true if part removed, false otherwise

**Throws:**
- MessagingException - if no such Part exists
- IllegalWriteException - if the underlying implementation does not support modification of existing values
public void removeBodyPart(int index) throws MessagingException

Remove the part at specified location (starting from 0). Shifts all the parts after the removed part down one.

Overrides:
removeBodyPart in class Multipart

Parameters:
index - Index of the part to remove

Throws:
MessagingException
IndexOutOfBoundsException - if the given index is out of range.
IllegalWriteException - if the underlying implementation does not support modification of existing values

public void addBodyPart(BodyPart part) throws MessagingException
Part multipartBodyPart Part

Throws:
MessagingException
IllegalWriteException:
addBodyPart

public void addBodyPart(BodyPart part) throws MessagingException

Adds a Part to the multipart. The BodyPart is appended to the list of existing Parts.

Overrides:
addBodyPart in class Multipart

Parameters:
part - The Part to be appended

Throws:
MessagingException
IllegalWriteException - if the underlying implementation does not support modification of existing values

public void addBodyPart(BodyPart part, int index) throws MessagingException

Adds a BodyPart at position index. If index is not the last one in the list, the subsequent parts are shifted up. If index is larger than the number of parts present, the BodyPart is appended to the end.

Overrides:
addBodyPart in class Multipart

**Parameters:**
- part - The BodyPart to be inserted
- index - Location where to insert the part

**Throws:**
- MessagingException
- IllegalWriteException - if the underlying implementation does not support modification of existing values

---

```java
public boolean isComplete() throws MessagingException {
    multipart true multipart multipart false
    "mail.mime.multipart.ignoremissingendboundary" false MessagingException
    return true
    since JavaMail 1.4
```

**isComplete**

Return true if the final boundary line for this multipart was seen.
When parsing multipart content, this class will (by default) terminate
parsing with no error if the end of input is reached before seeing
the final multipart boundary line. In such a case, this method will return
false. (If the System property
"mail.mime.multipart.ignoremissingendboundary" is set to false,
parsing such a message will instead throw a MessagingException.)

**Returns:**
- true if the final boundary line was seen

**Throws:**
- MessagingException

**Since:**
- JavaMail 1.4
public String getPreamble() throws MessagingException

multipart IMAP

return null

since JavaMail 1.4

getPreamble

public String getPreamble()

throws MessagingException

Get the preamble text, if any, that appears before the first body part of this multipart. Some protocols, such as IMAP, will not allow access to the preamble text.

Returns:
the preamble text, or null if no preamble

Throws:
MessagingException

Since:
JavaMail 1.4

public void setPreamble(String preamble) throws MessagingException

MIME

\[ \text{preamble} \]

since JavaMail 1.4

setPreamble

public void setPreamble(String preamble)

throws MessagingException

Set the preamble text to be included before the first body part.
Applications should generally not include any preamble text. In some cases it may be helpful to include preamble text with instructions for users of pre-MIME software. The preamble text should be complete lines, including newlines.

**Parameters:**
- preamble - the preamble text

**Throws:**
- MessagingException

**Since:**
- JavaMail 1.4

```java
protected void updateHeaders() throws MessagingException
BodyPart updateHeaders
MimeMultipart boundary

Multipart Message saveChanges Message
MimeMultipart

updateHeaders

protected void updateHeaders()
throws MessagingException

Update headers. The default implementation here just calls the updateHeaders method on each of its children BodyParts.

Note that the boundary parameter is already set up when a new and empty MimeMultipart object is created.

This method is called when the saveChanges method is invoked on the Message object containing this Multipart. This is typically done as part of the Message send process, however note that a client is free to call it any number of times. So if the header updating process
is expensive for a specific MimeMultipart subclass, then it might itself want to track whether its internal state actually did change, and do the header updating only if necessary.

**Throws:**
- `MessagingException`

---

```java
public void writeTo(java.io.OutputStream os) throws java.io.IOException, MessagingException
```

**Specified by:**
- `writeTo` in class `Multipart`

**Throws:**
- `IOException` - if an IO related exception occurs
- `MessagingException`

---

```java
protected void parse() throws MessagingException
```

**since** JavaMail 1.2

---

**parse**

```java
protected void parse()
```
Parse the InputStream from our DataSource, constructing the appropriate MimeBodyParts. The parsed flag is set to true, and if true on entry nothing is done. This method is called by all other methods that need data for the body parts, to make sure the data has been parsed.

**Throws:**

- MessagingException

**Since:**

JavaMail 1.2

---

```java
protected InternetHeaders createInternetHeaders(InputStream is) throws MessagingException

InputStream InternetHeaders
InternetHeaders InternetHeaders

is InputStream

Throws

MessagingException:

since JavaMail 1.2

```

---

**createInternetHeaders**

```java
protected InternetHeaders createInternetHeaders(InputStream is) throws MessagingException

Create and return an InternetHeaders object that loads the headers from the given InputStream. Subclasses can override this method to return a subclass of InternetHeaders, if necessary. This implementation simply constructs and returns an InternetHeaders object.

**Parameters:**

- is - the InputStream to read the headers from

**Throws:**

- MessagingException:
protected MimeBodyPart createMimeBodyPart(InternetHeaders headers, byte[] content) throws MessagingException

Create and return a MimeBodyPart object to represent a body part parsed from the InputStream. Subclasses can override this method to return a subclass of MimeBodyPart, if necessary. This implementation simply constructs and returns a MimeBodyPart object.

Parameters:
- headers - the headers for the body part
- content - the content of the body part

Throws:
- MessagingException

Since:
JavaMail 1.2
createMimeBodyPart(java.io.InputStream is) throws MessagingException

MimeBodyPart  InputStream
MimeBodyPart  MimeBodyPart

is  InputStream

Throws  MessagingException:

since  JavaMail 1.2

createMimeBodyPart

protected  MimeBodyPart  createMimeBodyPart(InputStream is)

throws  MessagingException

Create and return a MimeBodyPart object to represent a body part parsed from the InputStream. Subclasses can override this method to return a subclass of MimeBodyPart, if necessary. This implementation simply constructs and returns a MimeBodyPart object.

Parameters:

is - InputStream containing the body part

Throws:

MessagingException

Since:

JavaMail 1.2

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.mail.internet Interface MimePart

All Superinterfaces:
   Part

All Known Implementing Classes:
   MimeBodyPart, MimeMessage, PreencodedMimeBodyPart

public interface MimePart
extends Part

Implements: Part
Implemented by: MimeBodyPart, MimeMessage

MimePart MIMERFC 2045Section 2.4
MimePart Part RFC822 MIME MimeMessage
MimeBodyPart

See also javax.mail.internet.MimeUtility, javax.mail.Part

The MimePart interface models an Entity as defined by MIME (RFC2045, Section 2.4).

MimePart extends the Part interface to add additional RFC822 and MIME specific semantics and attributes. It provides the base interface for the MimeMessage and MimeBodyPart classes
A note on RFC822 and MIME headers

RFC822 and MIME header fields **must** contain only US-ASCII characters. If a header contains non US-ASCII characters, it must be encoded as per the rules in RFC 2047. The MimeUtility class provided in this package can be used to achieve this. Callers of the `setHeader`, `addHeader`, and `addHeaderLine` methods are responsible for enforcing the MIME requirements for the specified headers. In addition, these header fields must be folded (wrapped) before being sent if they exceed the line length limitation for the transport (1000 bytes for SMTP). Received headers may have been folded. The application is responsible for folding and unfolding headers as appropriate.

**Author:**
John Mani

**See Also:**
MimeUtility, Part

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.mail.Part</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTACHMENT, INLINE</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>addHeaderLine(String line)</code></td>
<td>Add a raw RFC822 header-line.</td>
</tr>
<tr>
<td>Enumeration</td>
<td><code>getAllHeaderLines()</code></td>
<td>Get all header lines as an Enumeration of Strings.</td>
</tr>
<tr>
<td>String</td>
<td><code>getContentID()</code></td>
<td>Get the Content-ID of this part.</td>
</tr>
<tr>
<td>String[]</td>
<td><code>getContentLanguage()</code></td>
<td>Get the language tags specified in the Content-Language header of this MimePart.</td>
</tr>
<tr>
<td>String</td>
<td><code>getContentMD5()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>String getEncoding()</td>
<td>Get the transfer encoding of this part.</td>
<td></td>
</tr>
<tr>
<td>String getHeader(String name, String delimiter)</td>
<td>Get the values of all header fields available for this header, returned as a single String, with the values separated by the delimiter.</td>
<td></td>
</tr>
<tr>
<td>Enumeration getMatchingHeaderLines(String[] names)</td>
<td>Get matching header lines as an Enumeration of Strings.</td>
<td></td>
</tr>
<tr>
<td>Enumeration getNonMatchingHeaderLines(String[] names)</td>
<td>Get non-matching header lines as an Enumeration of Strings.</td>
<td></td>
</tr>
<tr>
<td>void setContentLanguage(String[] languages)</td>
<td>Set the Content-Language header of this MimePart.</td>
<td></td>
</tr>
<tr>
<td>void setContentMD5(String md5)</td>
<td>Set the Content-MD5 of this part.</td>
<td></td>
</tr>
<tr>
<td>void setText(String text)</td>
<td>Convenience method that sets the given String as this part's content, with a MIME type of &quot;text/plain&quot;.</td>
<td></td>
</tr>
<tr>
<td>void setText(String text, String charset)</td>
<td>Convenience method that sets the given String as this part's content, with a MIME type of &quot;text/plain&quot; and the specified charset.</td>
<td></td>
</tr>
<tr>
<td>void setText(String text, String charset, String subtype)</td>
<td>Convenience method that sets the given String as this part's content, with a primary MIME type of &quot;text&quot; and the specified MIME subtype.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.mail.Part

- addHeader
- getAllHeaders
- getContent
- getContentType
- getDataHandler
- getDescription
- getDisposition
- getFileName
- getHeader
- getInputStream
- getLineCount
- getMatchingHeaders
- getNonMatchingHeaders
- getSize
- isMimeType
- removeHeader
- setContent
- setContent
- setDataHandler
- setDescription
- setDisposition
- setFileName
- setHeader
- writeTo
public String getHeader(String name, String delimiter)
throws MessagingException

String delimiter delimiter

return

Throws MessagingException:

getHeader

String getHeader(String name, String delimiter)
throws MessagingException

Get the values of all header fields available for this header, returned as a single String, with the values separated by the delimiter. If the delimiter is null, only the first value is returned.

Parameters:
- name - the name of this header
- delimiter - delimiter between fields in returned string

Returns:
- the value fields for all headers with this name

Throws:
- MessagingException

public void addHeaderLine(String line) throws MessagingException

RFC822

Throws IllegalWriteException:

Throws IllegalStateException: Part READ_ONLY
addHeaderLine

```java
void addHeaderLine(String line) throws MessagingException
```

Add a raw RFC822 header-line.

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification
- `IllegalStateException` - if this Part is obtained from a `READ_ONLY` folder
- `MessagingException`

---

```java
public java.util.Enumeration<E> getAllHeaderLines() throws MessagingException
```

String Header RFC822 "name" "value"

---

getAllHeaderLines

```java
Enumeration getAllHeaderLines() throws MessagingException
```

Get all header lines as an Enumeration of Strings. A Header line is a raw RFC822 header-line, containing both the "name" and "value" field.

**Throws:**
- `MessagingException`

---

```java
public java.util.Enumeration<E> getMatchingHeaderLines(String[] names) throws MessagingException
```

String Header RFC822 "name" "value"
getMatchingHeaderLines

Enumeration getMatchingHeaderLines(String[] names)
throws MessagingException

Get matching header lines as an Enumeration of Strings. A Header line is a raw RFC822 header-line, containing both the "name" and "value" field.

Throws:
MessagingException

public java.utilEnumeration<E>
getNonMatchingHeaderLines(String[] names) throws MessagingException
String Header RFC822 "name" "value"

getNonMatchingHeaderLines

Enumeration getNonMatchingHeaderLines(String[] names)
throws MessagingException

Get non-matching header lines as an Enumeration of Strings. A Header line is a raw RFC822 header-line, containing both the "name" and "value" field.

Throws:
MessagingException

public String getEncoding() throws MessagingException
getEncoding

```java
String getEncoding() throws MessagingException
```

Get the transfer encoding of this part.

Returns:
content-transfer-encoding

Throws:
MessagingException

getContentTypeID

```java
public String getContentTypeID() throws MessagingException
Content-ID null
```

```java
return
```

getContentTypeMD5

```java
public String getContentTypeMD5() throws MessagingException
Content-MD5 null
```

```java
return
```

Get the transfer encoding of this part. Returns null if none present.

Returns:
content-ID

Throws:
MessagingException

Get the Content-ID of this part. Returns null if none present.

Returns:
content-ID

Throws:
MessagingException

Get the Content-MD5 of this part. Returns null if none present.

Returns:
content-MD5

Throws:
MessagingException
getContentMD5

String getContentMD5() throws MessagingException

Get the Content-MD5 digest of this part. Returns null if none present.

Returns: content-MD5
Throws: MessagingException

public void setContentMD5(String md5) throws MessagingException

Content-MD5

md5 MD5
Throws IllegalWriteException:
Throws IllegalStateException: Part READ_ONLY

setContentMD5

void setContentMD5(String md5) throws MessagingException

Set the Content-MD5 of this part.

Parameters: md5 - the MD5 value
Throws: IllegalWriteException - if the underlying implementation does not support modification
IllegalStateException - if this Part is obtained from a READ_ONLY folder
MessagingException
public String[] getContentLanguage() throws MessagingException

MimePart  Content-Language Content-Language
RFC 1766  null

getContentLanguage

String[] getContentLanguage() throws MessagingException

Get the language tags specified in the Content-Language header of this MimePart. The Content-Language header is defined by RFC 1766. Returns null if this header is not available.

Throws:
MessagingException

public void setContentLanguage(String[] languages) throws MessagingException

MimePart  Content-Language Content-Language
RFC 1766

languages

Throws
IllegalWriteException:
Throws  IllegalStateException: Part  READ_ONLY

setContentLanguage

void  setContentLanguage(String[] languages)
throws MessagingException

Set the Content-Language header of this MimePart. The Content-Language header is defined by RFC1766.

Parameters:
languages - array of language tags
Throws:

- **IllegalWriteException** - if the underlying implementation does not support modification
- **IllegalStateException** - if this Part is obtained from a READ_ONLY folder
- **MessagingException**

```java
public void setText(String text) throws MessagingException
    MIME "text/plain" String string US-ASCII "charset"

    text

    charset

    setText

    text

    Throws **MessagingException**: 
    See also **setText(String text, String charset)**
```

**setText**

```java
void setText(String text)
    throws MessagingException
```

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain". If the string contains non US-ASCII characters, it will be encoded using the platform's default charset. The charset is also used to set the "charset" parameter.

Note that there may be a performance penalty if `text` is large, since this method may have to scan all the characters to determine what charset to use.

If the charset is already known, use the `setText` method that takes the charset parameter.
public void setText(String text, String charset) throws MessagingException

MIME "text/plain" String Unicode "charset"

text
charset

Throws MessagingException:

Convenience method that sets the given String as this part's content, with a MIME type of "text/plain" and the specified charset. The given Unicode string will be charset-encoded using the specified charset. The charset is also used to set "charset" parameter.

Parameters:
  text - the text content to set
  charset - the charset to use for the text

Throws:
  MessagingException - if an error occurs


void setText(String text, String charset, String subtype)
throws MessagingException

Convenience method that sets the given String as this part's content, with a primary MIME type of "text" and the specified MIME subtype. The given Unicode string will be charset-encoded using the specified charset. The charset is also used to set the "charset" parameter.

Parameters:
  text - the text content to set
  charset - the charset to use for the text
  subtype - the MIME subtype to use (e.g., "html")

Throws:
  MessagingException - if an error occurs

Since: JavaMail 1.4
to **license terms.**

**PS:**
javax.mail.internet 

Class MimePartDataSource

java.lang.Object
  \- javax.mail.internet.MimePartDataSource

All Implemented Interfaces:
  DataSource, MessageAware

public class MimePartDataSource
extends Object
implements DataSource, MessageAware

Implements: DataSource, MessageAware

MimePart DataSource

See javax.mail.internet.MimePart,
also javax.activation.DataSource

A utility class that implements a DataSource out of a MimePart. This class is primarily meant for service providers.

Author: John Mani

See Also: MimePart, DataSource

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected MimePart part</td>
<td>The MimePart that provides the data for this DataSource.</td>
</tr>
</tbody>
</table>

Constructor Summary

MimePartDataSource(MimePart part)
Constructor, that constructs a DataSource from a MimePart.

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><code>getContentType()</code></td>
<td>Returns the content-type of this DataSource.</td>
</tr>
<tr>
<td>InputStream</td>
<td><code>getInputStream()</code></td>
<td>Returns an input stream from this MimePart.</td>
</tr>
<tr>
<td>MessageContext</td>
<td><code>getMessageContext()</code></td>
<td>Return the <code>MessageContext</code> for the current part.</td>
</tr>
<tr>
<td>String</td>
<td><code>getName()</code></td>
<td>DataSource method to return a name.</td>
</tr>
<tr>
<td>OutputStream</td>
<td><code>getOutputStream()</code></td>
<td>DataSource method to return an output stream.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

### Field Detail

**part**

protected `MimePart` **part**

The MimePart that provides the data for this DataSource.

**Since:**

JavaMail 1.4

### Constructor Detail
public MimePartDataSource(MimePart part)  
MimePart  DataSource

MimePartDataSource

public MimePartDataSource(MimePart part)

Constructor, that constructs a DataSource from a MimePart.

Method Detail

public java.io.InputStream getInputStream() throws java.io.IOException  
MimePart

MimePart  Content-Transfer-Encoding

getContentStream()  Part  MimeUtility.decode()

return

See also  getContentStream, getContentStream, decode

getInputStream

public InputStream getInputStream() throws IOException

Returns an input stream from this MimePart.

This method applies the appropriate transfer-decoding, based on the Content-Transfer-Encoding attribute of this MimePart. Thus the returned input stream is a decoded stream of bytes.

This implementation obtains the raw content from the Part using the
getContentStream() method and decodes it using the MimeUtility.decode() method.

**Specified by:**
getInputStream in interface DataSource

**Returns:**
decoded input stream

**Throws:**
IOException

**See Also:**
MimeMessage.getContentStream(),
MimeBodyPart.getContentStream(),
MimeUtility.decode(java.io.InputStream, java.lang.String)

---

public java.io.OutputStream getOutputStream() throws java.io.IOException

DataSource

UnknownServiceException

getOutputStream

public OutputStream getOutputStream() throws IOException

DataSource method to return an output stream.

This implementation throws the UnknownServiceException.

**Specified by:**
getOutputStream in interface DataSource

**Returns:**
an OutputStream

**Throws:**
IOException
public String getContentType()

MimePart getContentType

getContentType

public String getContentType()

    Returns the content-type of this DataSource.
    
    This implementation just invokes the getContentType method on the MimePart.
    
    Specified by: getContentType in interface DataSource
    
    Returns: the MIME Type


public String getName()

DataSource

getName

public String getName()

    DataSource method to return a name.
    
    This implementation just returns an empty string.
    
    Specified by:
getName in interface DataSource

Returns:
the name of the object.

Public MessageContext getMessageContext()

MessageContext
since JavaMail 1.1

getMessageContext

Public MessageContext getMessageContext()

Return the MessageContext for the current part.

Specified by:
getMessageContext in interface MessageAware

Since:
JavaMail 1.1

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.activation  **Class MimeType**

[java.lang.Object](http://docs.oracle.com/javase/8/docs/api/java/lang/Object.html)  
[javax.activation.MimeType](http://docs.oracle.com/javase/8/docs/api/javax/activation/MimeType.html)

**All Implemented Interfaces:**  
[Externalizable](http://docs.oracle.com/javase/8/docs/api/java/io/Externalizable.html), [Serializable](http://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html)

```java
public class MimeType
    extends Object
    implements Externalizable
```  

**Implements:** java.io.Externalizable  

RFC 2045  2046  Internet Multipurpose Internet Mail Extension  

**MIME**

A Multipurpose Internet Mail Extension (MIME) type, as defined in RFC 2045 and 2046.

**See Also:**  
[Serialized Form](http://docs.oracle.com/javase/8/docs/api/javax/activation/MimeType.html)

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MimeType()</code></td>
<td>Default constructor.</td>
</tr>
<tr>
<td><code>MimeType(String rawdata)</code></td>
<td>Constructor that builds a MimeType from a String.</td>
</tr>
<tr>
<td><code>MimeType(String primary, String sub)</code></td>
<td>Constructor that builds a MimeType with the given primary and sub type but has an empty parameter list.</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String</code></td>
<td><code>getBaseType()</code></td>
<td>Return a String representation of this object without the parameter list.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getParameter(String name)</code></td>
<td>Retrieve the value associated with the given name, or null if there is no current association.</td>
</tr>
<tr>
<td><code>MimeTypeParameterList</code></td>
<td><code>getParameters()</code></td>
<td>Retrieve this object's parameter list.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getPrimaryType()</code></td>
<td>Retrieve the primary type of this object.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getSubType()</code></td>
<td>Retrieve the subtype of this object.</td>
</tr>
<tr>
<td><code>boolean</code></td>
<td><code>match(MimeType type)</code></td>
<td>Determine if the primary and sub type of this object is the same as what is in the given type.</td>
</tr>
<tr>
<td><code>boolean</code></td>
<td><code>match(String rawdata)</code></td>
<td>Determine if the primary and sub type of this object is the same as the content type described in rawdata.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>readExternal(ObjectInput in)</code></td>
<td>The object implements the readExternal method to restore its contents by calling the methods of DataInput for primitive types and readObject for objects, strings and arrays.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>removeParameter(String name)</code></td>
<td>Remove any value associated with the given name.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setParameter(String name, String value)</code></td>
<td>Set the value to be associated with the given name, replacing any previous association.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setPrimaryType(String primary)</code></td>
<td>Set the primary type for this object to the given String.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setSubType(String sub)</code></td>
<td></td>
</tr>
</tbody>
</table>
void Set the subtype for this object to the given String.

String toString()
    Return the String representation of this object.

void writeExternal(ObjectOutput out)
    The object implements the writeExternal method to save its contents by calling the methods of DataOutput for its primitive values or calling the writeObject method of ObjectOutput for objects, strings and arrays.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MimeType()

MimeType

public MimeType()
    Default constructor.

public MimeType(String rawdata) throws MimeTypeParseException
String MimeType
    rawdata MIME
**MimeType**

```java
public MimeType(String rawdata) throws MimeTypeParseException
```

Constructor that builds a MimeType from a String.

**Parameters:**
- rawdata - the MIME type string

**Throws:**
- MimeTypeParseException

```java
public MimeType(String primary, String sub) throws MimeTypeParseException
```

**MimeType**

```java
primary MIME
sub MIME
```

**Throws:**
- MimeTypeParseException

**MimeType**

```java
public MimeType(String primary, String sub) throws MimeTypeParseException
```

Constructor that builds a MimeType with the given primary and sub type but has an empty parameter list.

**Parameters:**
- primary - the primary MIME type
- sub - the MIME sub-type

**Throws:**
- MimeTypeParseException - if the primary type or subtype is not a valid token

---

**Method Detail**
public String getPrimaryType()

    return MIME

getPrimaryType

public  String  getPrimaryType()

    Retrieve the primary type of this object.

    Returns:
        the primary MIME type

public void setPrimaryType(String primary) throws String

    primary MIME

    Throws: MimeTypeParseException:

setPrimaryType

public void setPrimaryType(String primary)

    throws MimeTypeParseException

    Set the primary type for this object to the given String.

    Parameters:
        primary - the primary MIME type

    Throws:
        MimeTypeParseException - if the primary type is not a valid token

public String getSubType()
getSubType

public String getSubType()

Retrieve the subtype of this object.

Returns:
the MIME subtype

setSubType

public void setSubType(String sub) throws MimeTypeParseException

String

sub

MIME

Throws:
MimeTypeParseException:

setSubType

public void setSubType(String sub)
throws MimeTypeParseException

Set the subtype for this object to the given String.

Parameters:
sub - the MIME subtype

Throws:
MimeTypeParseException - if the subtype is not a valid token

getParameterList
getParameters

public MimeTypeInfoList getParameters()

Retrieve this object's parameter list.

Returns:
  a MimeTypeInfoList object representing the parameters

getParameter

public String getParameter(String name)

Retrieve the value associated with the given name, or null if there is no current association.

Parameters:
  name - the parameter name

Returns:
  the parameter's value

setParameter

public void setParameter(String name, String value)

setParameter
Set the value to be associated with the given name, replacing any previous association.

**Parameters:**
- *name* - the parameter name
- *value* - the parameter's value

---

```java
public void removeParameter(String name)
```

Remove any value associated with the given name.

**Parameters:**
- *name* - the parameter name

---

```java
public String toString()
```

Return the String representation of this object.

**Overrides:**
- `toString` in class `Object`
public String getBaseType()

String

    return MIME

getBaseType

gpublic String getBaseType()

    Return a String representation of this object without the parameter list.

    Returns:
    the MIME type and sub-type

match

public boolean match(MimeType type)

type
    return MimeType

match

public boolean match(MimeType type)

    Determine if the primary and sub type of this object is the same as what is in the given type.

    Parameters:
    type - the MimeType object to compare with

    Returns:
    true if they match

match

public boolean match(String rawdata) throws 
MimeTypeParseException
### rawdata

```
rawdata
  rawdata MIME
  return true
```

### match

```java
public boolean match(String rawdata)
  throws MimeTypeParseException
```

Determine if the primary and sub type of this object is the same as the content type described in rawdata.

**Parameters:**
- `rawdata` - the MIME type string to compare with

**Returns:**
- `true` if they match

**Throws:**
- `MimeTypeParseException`

### writeExternal

```java
public void writeExternal(java.io.ObjectOutput out) throws java.io.IOException
```

writeExternal  DataOutput  ObjectOutput  writeObject
```

**Throws**
- `java.io.IOException`: I/O

### writeExternal

```java
public void writeExternal(ObjectOutput out)
  throws IOException
```

The object implements the writeExternal method to save its contents by calling the methods of DataOutput for its primitive values or calling the writeObject method of ObjectOutput for objects, strings and arrays.
public void readExternal(java.io.ObjectInput in) throws java.io.IOException, ClassNotFoundException
readExternal DataInput readObject
readExternal writeExternal
in ObjectInput
Throws ClassNotFoundException:

The object implements the readExternal method to restore its contents by calling the methods of DataInput for primitive types and readObject for objects, strings and arrays. The readExternal method must read the values in the same sequence and with the same types as were written by writeExternal.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.activation  Class MimeTypeParameterList

java.lang.Object
   `javax.activation.MimeTypeParameterList`

public class MimeTypeParameterList
extends Object

RFC 2045 2046 MimeType

See also `javax.activation.MimeType`

A parameter list of a MimeType as defined in RFC 2045 and 2046. The Primary type of the object must already be stripped off.

See Also:
 `MimeType`

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MimeTypeParameterList()</code></td>
<td>Default constructor.</td>
</tr>
<tr>
<td><code>MimeTypeParameterList(String parameterList)</code></td>
<td>Constructs a new MimeTypeParameterList with the passed in data.</td>
</tr>
</tbody>
</table>

---

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>get(String name)</code></td>
<td>Retrieve the value associated with the given name, or null if there is no current association.</td>
</tr>
<tr>
<td><code>getNames()</code></td>
<td>Retrieve an enumeration of all the names in this list.</td>
</tr>
</tbody>
</table>
boolean isEmpty()  
Determine whether or not this list is empty.

protected void parse(String parameterList)  
A routine for parsing the parameter list out of a String.

void remove(String name)  
Remove any value associated with the given name.

void set(String name, String value)  
Set the value to be associated with the given name, replacing any previous association.

int size()  
Return the number of name-value pairs in this list.

String toString()  
Return a string representation of this object.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MimeTypeParameterList()

Default constructor.

public MimeTypeParameterList(String parameterList)

throws MimeTypeParseException

MimeTypeParameterList
**MimeTypeParameterList**

public MimeTypeParameterList(String parameterList) throws MimeTypeParseException

Constructs a new MimeTypeParameterList with the passed in data.

**Parameters:**
- parameterList - an RFC 2045, 2046 compliant parameter list.

**Throws:**
- MimeTypeParseException

### Method Detail

**protected void parse(String parameterList) throws MimeTypeParseException**

String

```
parameterList
```

A routine for parsing the parameter list out of a String.

**Parameters:**
- parameterList - an RFC 2045, 2046 compliant parameter list.

**Throws:**
- MimeTypeParseException

**public int size()**

-
```java
public int size()
{
    return
}
```

**size**

Return the number of name-value pairs in this list.

**Returns:**
the number of parameters

```java
public boolean isEmpty()
{
    return true
}
```

**isEmpty**

Determine whether or not this list is empty.

**Returns:**
true if there are no parameters

```java
public String get(String name)
{
    return
}
```

**get**

public **String** get(String name)
Retrieve the value associated with the given name, or null if there is no current association.

**Parameters:**
- name - the parameter name

**Returns:**
- the parameter's value

---

**public void set(String name, String value)**

**set**

```java
public void set(String name, String value)
```

Set the value to be associated with the given name, replacing any previous association.

**Parameters:**
- name - the parameter name
- value - the parameter's value

---

**public void remove(String name)**

**remove**

```java
public void remove(String name)
```

Remove any value associated with the given name.
Parameters:
name - the parameter name

public java.util.Enumeration<E> getNames()

    return

getNames

public Enumeration getNames()

    Retrieve an enumeration of all the names in this list.

    Returns:
    an enumeration of all parameter names

public String toString()

toString

public String toString()

    Return a string representation of this object.

    Overrides:
    toString in class Object
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.activation  **Class MimeTypeError ParseException**

java.lang.Object  
  ⊃ java.lang.Throwable  
    ⊃ java.lang.Exception  
      ⊃ javax.activation.MimeTypeParseException

**All Implemented Interfaces:**

Serializable

```java
public class MimeTypeError ParseException
    extends Exception
```

**Extends:** Throwable > Exception

MimeType

A class to encapsulate MimeType parsing related exceptions.

**See Also:**

Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MimeTypeParseException()</strong></td>
<td>Constructs a MimeTypeError(ParseException with no specified detail message.</td>
</tr>
<tr>
<td><strong>MimeTypeParseException(String s)</strong></td>
<td>Constructs a MimeTypeError(ParseException with the specified detail message.</td>
</tr>
</tbody>
</table>

---

**Method Summary**
Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public MimeTypeErrorParseException()
MimeTypeErrorParseException

MimeTypeErrorParseException

public MimeTypeErrorParseException()

Constructs a MimeTypeErrorParseException with no specified detail message.

public MimeTypeErrorParseException(String s)
MimeTypeErrorParseException

s

MimeTypeErrorParseException

public MimeTypeErrorParseException(String s)

Constructs a MimeTypeErrorParseException with the specified detail message.

Parameters:
s - the detail message.
javax.activation  Class MimetypesFileTypeMap

java.lang.Object
   ▼ javax.activation.FileTypeMap
      ▼ javax.activation.MimetypesFileTypeMap

public class MimetypesFileTypeMap extends FileTypeMap

Extends: FileTypeMap

File Type Map

MIME

MimetypesFileTypeMap MIME MimetypesFileTypeMap
MIME MIME

1. MimetypesFileTypeMap
2. .mime.types
3. <$java.home>/lib/mime.types
4. META-INF/mime.types
5. META-INF/mimetypes.default activation.jar

MIME

# '#'
# <mime > <>
#
text/plain txt text TXT
# file.txt file.text file.TXT
# mime "text/plain"

This class extends FileTypeMap and provides data typing of files via their file extension. It uses the .mime.types format.

MIME types file search order:
The MimetypesFileTypeMap looks in various places in the user's system for MIME types file entries. When requests are made to search for MIME types in the MimetypesFileTypeMap, it searches MIME types files in the following order:

1. Programmatically added entries to the MimetypesFileTypeMap instance.
2. The file `.mime.types` in the user's home directory.
3. The file `<java.home>/lib/mime.types`.
4. The file or resources named META-INF/mime.types.
5. The file or resource named META-INF/mimetypes.default (usually found only in the activation.jar file).

**MIME types file format:**

```
# comments begin with a '#'
# the format is <mime type> <space separated file extensions>
# for example:
text/plain txt text TXT
# this would map file.txt, file.text, and file.TXT to
# the mime type "text/plain"
```

**Author:**

Bart Calder, Bill Shannon

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>MimetypesFileTypeMap()</code></td>
<td>The default constructor.</td>
</tr>
<tr>
<td><code>MimetypesFileTypeMap(InputStream is)</code></td>
<td>Construct a MimetypesFileTypeMap with programmatic entries added from the InputStream.</td>
</tr>
<tr>
<td><code>MimetypesFileTypeMap(String mimeTypesFileName)</code></td>
<td>Construct a MimetypesFileTypeMap with programmatic entries added from the named file.</td>
</tr>
</tbody>
</table>

---

**Method Summary**


**addMimeTypes**

Prepend the MIME type values to the registry.

**getContentType**

Return the MIME type of the file object.

**getContentType**

Return the MIME type based on the specified file name.

Methods inherited from class `javax.activation.FileTypeMap`

- `getDefaultFileTypeMap`, `setDefaultFileTypeMap`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Constructor Detail**

```java
public MimetypesFileTypeMap()
```

**MimetypesFileTypeMap**

```java
public MimetypesFileTypeMap()
```

The default constructor.

```java
public MimetypesFileTypeMap(String mimeTypeFileName)
```

`mimeTypeFileName`
public MimetypesFileTypeMap(String mimeTypeFileName) throws IOException

Construct a MimetypesFileTypeMap with programmatic entries added from the named file.

Parameters:
  mimeTypeFileName - the file name

Throws:
  IOException

public MimetypesFileTypeMap(java.io.InputStream is)
InputStream MimetypesFileTypeMap

MimetypesFileTypeMap

public MimetypesFileTypeMap(InputStream is)

Construct a MimetypesFileTypeMap with programmatic entries added from the InputStream.

Parameters:
  is - the input stream to read from

Method Detail

public void addMimeTypes(String mime_types)

MIME

  mime_types .mime.

addMimeTypes

public void addMimeTypes(String mime_types)
Prepend the MIME type values to the registry.

**Parameters:**
- mime_types - A .mime.types formatted string of entries.

```java
public String getContentType(java.io.File f)

MIME
getContentType(f.getName())

f
return

getContentType
```

**getContentType**

```java
public String getContentType(File f)

getContentType(f.getName())

f
return

MIME
```

**Specified by:**
- `getContentType` in class `FileTypeMap`

**Parameters:**
- f - the file

**Returns:**
- the file's MIME type

```java
public String getContentType(String filename)

MIME
"application/octet-stream"

filename
return

MIME
```

**getContentType**

```java
public String getContentType(String filename)

MIME
"application/octet-stream"

filename
return

MIME
```
Return the MIME type based on the specified file name. The MIME type entries are searched as described above under MIME types file search order. If no entry is found, the type "application/octet-stream" is returned.

**Specified by:**
getContentType in class FileTypeMap

**Parameters:**
filename - the file name

**Returns:**
the file's MIME type

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.internet Class MimeUtility

java.lang.Object
   javax.mail.internet.MimeUtility

public class MimeUtility
extends Object

MIME

RFC 2047 MIME

BASE64 QP RFC 2047

Java String 16 Unicode ASCII Unicode 0 -
127 ASCII String String US-ASCII
Unicode String

String SMTP
   byte[] bytes = string.getBytes("iso-8859-1");

MimeMessage MimeBodyPart setHeader addHeader US-
ASCII Unicode US-ASCII

MimeMessage MimeBodyPart getHeader RFC 2047
Unicode String

MIME

mail.mime.decodetext.strict MIME MIME
mail.mime.decodetext.strict "false" true

mail.mime.encodeeol.strict "text" MIME Content-Transfer-
This is a utility class that provides various MIME related functionality.

There are a set of methods to encode and decode MIME headers as per RFC 2047. A brief description on handling such headers is given below:

RFC 822 mail headers **must** contain only US-ASCII characters. Headers that contain non US-ASCII characters must be encoded so that they contain only US-ASCII characters. Basically, this process involves using either BASE64 or QP to encode certain characters. RFC 2047 describes this in detail.

In Java, Strings contain (16 bit) Unicode characters. ASCII is a subset of Unicode (and occupies the range 0 - 127). A String that contains only ASCII characters is already mail-safe. If the String contains non US-ASCII characters, it must be encoded. An additional complexity in this step is that since Unicode is not yet a widely used charset, one might want to first charset-encode the String into another charset and then do the transfer-encoding.

Note that to get the actual bytes of a mail-safe String (say, for sending over SMTP), one must do

```
byte[] bytes = string.getBytes("iso-8859-1");
```

The `setHeader` and `addHeader` methods on MimeMessage and MimeBodyPart assume that the given header values are Unicode strings that contain only US-ASCII characters. Hence the callers of those methods must insure that the values they pass do not contain non US-ASCII characters. The methods in this class help do this.
The `getHeader` family of methods on `MimeMessage` and `MimeBodyPart` return the raw header value. These might be encoded as per RFC 2047, and if so, must be decoded into Unicode Strings. The methods in this class help to do this.

Several System properties control strict conformance to the MIME spec. Note that these are not session properties but must be set globally as System properties.

The `mail.mime.decodetext.strict` property controls decoding of MIME encoded words. The MIME spec requires that encoded words start at the beginning of a whitespace separated word. Some mailers incorrectly include encoded words in the middle of a word. If the `mail.mime.decodetext.strict` System property is set to "false", an attempt will be made to decode these illegal encoded words. The default is true.

The `mail.mime.encodeeol.strict` property controls the choice of Content-Transfer-Encoding for MIME parts that are not of type "text". Often such parts will contain textual data for which an encoding that allows normal end of line conventions is appropriate. In rare cases, such a part will appear to contain entirely textual data, but will require an encoding that preserves CR and LF characters without change. If the `mail.mime.encodeeol.strict` System property is set to "true", such an encoding will be used when necessary. The default is false.

In addition, the `mail.mime.charset` System property can be used to specify the default MIME charset to use for encoded words and text parts that don't otherwise specify a charset. Normally, the default MIME charset is derived from the default Java charset, as specified in the `file.encoding` System property. Most applications will have no need to explicitly set the default MIME charset. In cases where the default MIME charset to be used for mail messages is different than the charset used for files stored on the system, this property should be set.

**Version:**
1.60, 07/05/15

**Author:**
John Mani, Bill Shannon
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int ALL</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>decoded</code></td>
<td><em>InputStream is, String encoding</em> Decode the given input stream.</td>
</tr>
<tr>
<td><code>decodeText</code></td>
<td><em>String etext</em> Decode &quot;unstructured&quot; headers, that is, headers that are defined as &quot;text&quot; as per RFC 822.</td>
</tr>
<tr>
<td><code>decodeWord</code></td>
<td><em>String eword</em> The string is parsed using the rules in RFC 2047 for parsing an &quot;encoded-word&quot;.</td>
</tr>
<tr>
<td><code>encoded</code></td>
<td><em>OutputStream os, String encoding</em> Wrap an encoder around the given output stream.</td>
</tr>
<tr>
<td><code>encodeText</code></td>
<td><em>String text</em> Encode a RFC 822 &quot;text&quot; token into mail-safe form as per RFC 2047.</td>
</tr>
<tr>
<td><code>encodeText</code></td>
<td><em>String text, String charset, String encoding</em> Encode a RFC 822 &quot;text&quot; token into mail-safe form as per RFC 2047.</td>
</tr>
<tr>
<td><code>encodeWord</code></td>
<td><em>String word</em> Encode a RFC 822 &quot;word&quot; token into mail-safe form as per RFC 2047.</td>
</tr>
<tr>
<td><code>encodeWord</code></td>
<td><em>String word, String charset, String encoding</em> Encode a RFC 822 &quot;word&quot; token into mail-safe form as per RFC 2047.</td>
</tr>
<tr>
<td><code>fold</code></td>
<td>(int used, String s)</td>
</tr>
</tbody>
</table>
Fold a string at linear whitespace so that each line is no longer than 76 characters, if possible.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <code>String</code> <code>getDefaultJavaCharset()</code></td>
<td>Get the default charset corresponding to the system's current default locale.</td>
</tr>
<tr>
<td>static <code>String</code> <code>getEncoding(DataHandler dh)</code></td>
<td>Same as <code>getEncoding(DataSource ds)</code> except that instead of reading the data from an InputStream it uses the <code>writeTo</code> method to examine the data.</td>
</tr>
<tr>
<td>static <code>String</code> <code>getEncoding(DataSource ds)</code></td>
<td>Get the content-transfer-encoding that should be applied to the input stream of this datasource, to make it mailsafe.</td>
</tr>
<tr>
<td>static <code>String</code> <code>javaCharset(String charset)</code></td>
<td>Convert a MIME charset name into a valid Java charset name.</td>
</tr>
<tr>
<td>static <code>String</code> <code>mimeCharset(String charset)</code></td>
<td>Convert a java charset into its MIME charset name.</td>
</tr>
<tr>
<td>static <code>String</code> <code>quote(String word, String specials)</code></td>
<td>A utility method to quote a word, if the word contains any characters from the specified 'specials' list.</td>
</tr>
<tr>
<td>static <code>String</code> <code>unfold(String s)</code></td>
<td>Unfold a folded header.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.`Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public static final int ALL

See Also:
Constant Field Values

Method Detail

public static String getEncoding(DataSource ds)

- "text" US-ASCII "7bit" US-ASCII "base64" US-ASCII "quoted-printable"
- "text" US-ASCII "7bit" US-ASCII "base64"

getEncoding

public static String getEncoding(DataSource ds)

Get the content-transfer-encoding that should be applied to the input stream of this datasource, to make it mailsafe.

The algorithm used here is:

- If the primary type of this datasource is "text" and if all the bytes in its input stream are US-ASCII, then the encoding is "7bit". If more than half of the bytes are non-US-ASCII, then the encoding is "base64". If less than half of the bytes are non-US-ASCII, then the encoding is "quoted-printable".
- If the primary type of this datasource is not "text", then if all the bytes of its input stream are US-ASCII, the encoding is "7bit". If
there is even one non-US-ASCII character, the encoding is "base64".

**Parameters:**
- ds - DataSource

**Returns:**
the encoding. This is either "7bit", "quoted-printable" or "base64"

---

```java
public static String getEncoding(DataHandler dh) {
    return getEncoding(DataSource) instanceof InputStream ?
        writeTo
    : MIME "text/plain" String
        InputStream DataHandler writeTo

    since JavaMail 1.2
}
```

---

### getEncoding

```java
public static String getEncoding(DataHandler dh) {
    Same as getEncoding(DataSource) except that instead of reading the data from an InputStream it uses the writeTo method to examine the data. This is more efficient in the common case of a DataHandler created with an object and a MIME type (for example, a "text/plain" String) because all the I/O is done in this thread. In the case requiring an InputStream the DataHandler uses a thread, a pair of pipe streams, and the writeTo method to produce the data.

Since: JavaMail 1.2
```

---

```java
public static java.io.InputStream decode(java.io.InputStream is, String encoding) throws MessagingException
```
**InputStream RFC 2045**

"base64" "quoted-printable" "7bit" "8bit" "binary" "uuencode"

```
is
encoding
return
```

**decode**

```java
public static InputStream decode(InputStream is, String encoding)
throws MessagingException
```

Decode the given input stream. The Input stream returned is the decoded input stream. All the encodings defined in RFC 2045 are supported here. They include "base64", "quoted-printable", "7bit", "8bit", and "binary". In addition, "uuencode" is also supported.

**Parameters:**
- `is` - input stream
- `encoding` - the encoding of the stream.

**Returns:**
- decoded input stream.

**Throws:**
- `MessagingException`

---

```java
public static java.io.OutputStream encode(java.io.OutputStream os, String encoding)
throws MessagingException
```

RFC 2045 "base64" "quoted-printable" "7bit" "8bit" "binary" "uuencode"

```
    os
    encoding
    return
```
encode

public static OutputStream encode(OutputStream os, String encoding) throws MessagingException

Wrap an encoder around the given output stream. All the encodings defined in RFC 2045 are supported here. They include "base64", "quoted-printable", "7bit", "8bit" and "binary". In addition, "uuencode" is also supported.

Parameters:
- os - output stream
- encoding - the encoding of the stream.

Returns:
output stream that applies the specified encoding.

Throws:
MessagingException

---

public static java.io.OutputStream encode(java.io.OutputStream os, String encoding, String filename) throws MessagingException

RFC 2045 "base64" "quoted-printable" "7bit" "8bit" "binary" "uuencode" filename "uuencode"

os
encoding
filename uuencode
return
since JavaMail 1.2

en

encode

public static OutputStream encode(OutputStream os, String encoding, String filename)
Wrap an encoder around the given output stream. All the encodings defined in RFC 2045 are supported here. They include "base64", "quoted-printable", "7bit", "8bit" and "binary". In addition, "uuencode" is also supported. The filename parameter is used with the "uuencode" encoding and is included in the encoded output.

**Parameters:**
- os - output stream
- encoding - the encoding of the stream.
- filename - name for the file being encoded (only used with uuencode)

**Returns:**
output stream that applies the specified encoding.

**Throws:**
- MessagingException

**Since:**
JavaMail 1.2

```java
public static String encodeText(String text) throws java.io.UnsupportedEncodingException
RFC 2047 RFC 822 "text"


""RFC 822

MimePart part = ... String rawvalue = "FooBar Mailer, Japanese version 1.1"
try {
// If we know for sure that rawvalue contains only US-ASCII characters, we can skip the encoding part
part.setHeader("X-mailer", MimeUtility.encodeText(rawvalue)
catch (UnsupportedEncodingException e) {
```
public static String encodeText(String text) throws UnsupportedEncodingException

Encode a RFC 822 "text" token into mail-safe form as per RFC 2047.

The given Unicode string is examined for non US-ASCII characters. If the string contains only US-ASCII characters, it is returned as-is. If the string contains non US-ASCII characters, it is first character-encoded using the platform's default charset, then transfer-encoded using either the B or Q encoding. The resulting bytes are then returned as a Unicode string containing only ASCII characters.

Note that this method should be used to encode only "unstructured" RFC 822 headers.

Example of usage:

MimePart part = ...
String rawvalue = "FooBar Mailer, Japanese version 1.1"
try {
    // If we know for sure that rawvalue contains only US-ASCII characters, we can skip the encoding part
    part.setHeader("X-mailer", MimeUtility.encodeText(rawvalue)
} catch (UnsupportedEncodingException e) {
    // encoding failure
} catch (MessagingException me) {
    // setHeader() failure
Parameters:
  text - Unicode string

Returns:
  Unicode string containing only US-ASCII characters

Throws:
  UnsupportedEncodingException - if the encoding fails

public static String encodeText(String text, String charset, String encoding) throws java.io.UnsupportedEncodingException

RFC 2047 RFC 822 "text"


""RFC 822
  text
  charset null
  encoding "B" "Q" null ASCII "Q" "B"
  return US-ASCII Unicode

encodeText

public static String encodeText(String text, String charset, String encoding) throws UnsupportedEncodingException

Encode a RFC 822 "text" token into mail-safe form as per RFC 2047.

The given Unicode string is examined for non US-ASCII characters. If the string contains only US-ASCII characters, it is returned as-is.
the string contains non US-ASCII characters, it is first character-encoded using the specified charset, then transfer-encoded using either the B or Q encoding. The resulting bytes are then returned as a Unicode string containing only ASCII characters.

Note that this method should be used to encode only "unstructured" RFC 822 headers.

**Parameters:**
- text - the header value
- charset - the charset. If this parameter is null, the platform's default chatset is used.
- encoding - the encoding to be used. Currently supported values are "B" and "Q". If this parameter is null, then the "Q" encoding is used if most of characters to be encoded are in the ASCII charset, otherwise "B" encoding is used.

**Returns:**
Unicode string containing only US-ASCII characters

**Throws:**
UnsupportedEncodingException

```java
public static String decodeText(String etext) throws java.io.UnsupportedEncodingException
```

""" RFC 822 "*text"

RFC 2047 6.1
UnsupportedEncodingException String RFC 2047

```java
MimePart part = ...
String rawvalue = null;
String value = null;
try {
if ((rawvalue = part.getHeader("X-mailer")[0]) != null) 
value = MimeUtility.decodeText(rawvalue);
} catch (UnsupportedEncodingException e) {
```
Don't care
value = rawvalue;
} catch (MessagingException me) { }

return value;

decodeText

public static String decodeText(String etext)
throws UnsupportedEncodingException

Decode "unstructured" headers, that is, headers that are defined as 'text' as per RFC 822.

The string is decoded using the algorithm specified in RFC 2047, Section 6.1. If the charset-conversion fails for any sequence, an UnsupportedEncodingException is thrown. If the String is not an RFC 2047 style encoded header, it is returned as-is.

Example of usage:

MimePart part = ... String rawvalue = null; String value = null; try {
if ((rawvalue = part.getHeader("X-mailer") [0]) != null)
  value = MimeUtility.decodeText(rawvalue);
} catch (UnsupportedEncodingException e) {
  // Don't care
  value = rawvalue;
} catch (MessagingException me) { }

return value;
public static String encodeWord(String word) throws java.io.UnsupportedEncodingException
RFC 2047 RFC 822 "word"


RFC 822 "phrases" InternetAddress 'phrase'

    word                        Unicode
    return                      US-ASCII Unicode
    Throws                      java.io.UnsupportedEncodingException:

encodeWord

public static String encodeWord(String word)
throws UnsupportedEncodingException

Encode a RFC 2047 "word" token into mail-safe form as per RFC 822.

The given Unicode string is examined for non US-ASCII characters. If the string contains only US-ASCII characters, it is returned as-is. If the string contains non US-ASCII characters, it is first character-encoded using the platform's default charset, then transfer-encoded using either the B or Q encoding. The resulting bytes are then returned as a Unicode string containing only ASCII characters.

This method is meant to be used when creating RFC 822 "phrases". The InternetAddress class, for example, uses this to encode it's
'phrase' component.

**Parameters:**
- **word** - Unicode string

**Returns:**
- Array of Unicode strings containing only US-ASCII characters.

**Throws:**
- `UnsupportedEncodingException` - if the encoding fails

```java
public static String encodeWord(String word, String charset, String encoding) throws java.io.UnsupportedEncodingException
RFC 2047 RFC 822 "word"


| word     | Unicode |
| charset  | MIME    |
| encoding | "B" "Q" null ASCII "Q" "B"
| return   | US-ASCII Unicode |

Throws: java.io.UnsupportedEncodingException:
```

**encodeWord**

```java
public static String encodeWord(String word, String charset, String encoding) throws UnsupportedEncodingException

Encode a RFC 822 "word" token into mail-safe form as per RFC 2047.

The given Unicode string is examined for non US-ASCII characters. If the string contains only US-ASCII characters, it is returned as-is. If
the string contains non US-ASCII characters, it is first character-encoded using the specified charset, then transfer-encoded using either the B or Q encoding. The resulting bytes are then returned as a Unicode string containing only ASCII characters.

**Parameters:**
- `word` - Unicode string
- `charset` - the MIME charset
- `encoding` - the encoding to be used. Currently supported values are "B" and "Q". If this parameter is null, then the "Q" encoding is used if most of characters to be encoded are in the ASCII charset, otherwise "B" encoding is used.

**Returns:**
- Unicode string containing only US-ASCII characters

**Throws:**
- `UnsupportedEncodingException` - if the encoding fails

```java
public static String decodeWord(String eword) throws 
ParseException, java.io.UnsupportedEncodingException
```

**eword**

**Throws**
- `ParseException`: RFC 2047 encoded-word
- `java.io.UnsupportedEncodingException`:

**decodeWord**

```java
public static String decodeWord(String eword) 
throws ParseException, 
UnsupportedEncodingException
```

The string is parsed using the rules in RFC 2047 for parsing an "encoded-word". If the parse fails, a `ParseException` is thrown.
Otherwise, it is transfer-decoded, and then charset-converted into Unicode. If the charset-conversion fails, an UnsupportedEncodingException is thrown.

**Parameters:**
- `eword` - the encoded value

**Throws:**
- `ParseException` - if the string is not an encoded-word as per RFC 2047.
- `UnsupportedEncodingException` - if the charset conversion failed.

```java
public static String quote(String word, String specials)
' specials'
```

**HeaderTokenizer** - MIME  RFC 822

**RFC 822**  **MIME**

```java
word
specials
return
See also  MIME, RFC822
```

**quote**

```java
public static String quote(String word, String specials)
```

A utility method to quote a word, if the word contains any characters from the specified 'specials' list.

The **HeaderTokenizer** class defines two special sets of delimiters - MIME and RFC 822.

This method is typically used during the generation of RFC 822 and MIME header fields.
public static String fold(int used, String s)

Fold a string at linear whitespace so that each line is no longer than 76 characters, if possible. If there are more than 76 non-whitespace characters consecutively, the string is folded at the first whitespace after that sequence. The parameter `used` indicates how many characters have been used in the current line; it is usually the length of the header name.

Note that line breaks in the string aren't escaped; they probably should be.

**Parameters:**
- `used` - characters used in line so far
- `s` - the string to fold

**Parameters:**
- `word` - word to be quoted
- `specials` - the set of special characters

**Returns:**
the possibly quoted word

**See Also:**
HeaderTokenizer.MIME, HeaderTokenizer.RFC822
public static String unfold(String s)

Unfold a folded header. Any line breaks that aren't escaped and are followed by whitespace are removed.

Parameters:
  s - the string to unfold

Returns:
  the unfolded string

Since:
  JavaMail 1.4
public static String javaCharset(String charset)

Convert a MIME charset name into a valid Java charset name.

**Parameters:**
- charset - the MIME charset name

**Returns:**
- the Java charset equivalent. If a suitable mapping is not available, the passed in charset is itself returned.

---

public static String mimeCharset(String charset)

Java MIME

**JDK 1.2**

<table>
<thead>
<tr>
<th>charset</th>
<th>JDK</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>MIME/IANA</td>
</tr>
<tr>
<td>since</td>
<td>JavaMail 1.1</td>
</tr>
</tbody>
</table>

mimeCharset

public static String mimeCharset(String charset)

Convert a java charset into its MIME charset name.

Note that a future version of JDK (post 1.2) might provide this functionality, in which case, we may deprecate this method then.

**Parameters:**
- charset - the JDK charset

**Returns:**
- the MIME/IANA equivalent. If a mapping is not possible, the passed in charset itself is returned.

**Since:**
- JavaMail 1.1
public static String getDefaultJavaCharset()

    return Java MIME

since JavaMail 1.1

getDefaultJavaCharset

public static String getDefaultJavaCharset()

Get the default charset corresponding to the system's current default locale. If the System property mail.mime.charset is set, a system charset corresponding to this MIME charset will be returned.

Returns:
the default charset of the system's default locale, as a Java charset. (NOT a MIME charset)

Since:
JavaMail 1.1

Overview Package Tree Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.enterprise.deploy.shared Class ModuleType

javax.lang.Object
   javax.enterprise.deploy.shared.ModuleType

public class ModuleType
   extends Object

ModuleTypes  J2EE

Class ModuleTypes defines enumeration values for the J2EE module types.

Author:
   Rebecca Searls

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static ModuleType CAR  The module is an Client Application archive.</td>
</tr>
<tr>
<td>static ModuleType EAR  The module is an EAR archive.</td>
</tr>
<tr>
<td>static ModuleType EJB  The module is an Enterprise Java Bean archive.</td>
</tr>
<tr>
<td>static ModuleType RAR  The module is an Connector archive.</td>
</tr>
<tr>
<td>static ModuleType MAR  The module is an Web Application archive.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ModuleType(int value)</td>
</tr>
</tbody>
</table>
Construct a new enumeration value with the given integer value.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected ModuleType[] getEnumValueTable()</td>
<td>Returns the enumeration value table for class ModuleType</td>
</tr>
<tr>
<td>String getModuleExtension()</td>
<td>Return the file extension string for this enumeration.</td>
</tr>
<tr>
<td>static ModuleType getModuleType(int value)</td>
<td>Return an object of the specified value.</td>
</tr>
<tr>
<td>protected int getOffset()</td>
<td>Returns the lowest integer value used by this enumeration value's enumeration class.</td>
</tr>
<tr>
<td>protected String[] getStringTable()</td>
<td>Returns the string table for class ModuleType</td>
</tr>
<tr>
<td>int getValue()</td>
<td>Returns this enumeration value's integer value.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Return the string name of this ModuleType or the integer value if outside the bounds of the table</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Field Detail

EAR
public static final ModuleType EAR

   The module is an EAR archive.

---

EJB

public static final ModuleType EJB

   The module is an Enterprise Java Bean archive.

---

CAR

public static final ModuleType CAR

   The module is an Client Application archive.

---

RAR

public static final ModuleType RAR

   The module is an Connector archive.

---

WAR

public static final ModuleType WAR

   The module is an Web Application archive.
Constructor Detail

protected ModuleType(int value)

value

ModuleType

protected ModuleType(int value)

Construct a new enumeration value with the given integer value.

Parameters:
value - Integer value.

Method Detail

public int getValue()

return

getValue

public int getValue()

Returns this enumeration value's integer value.

Returns:
the value

protected String[] getStringTable()
getStringTable

protected String[] getStringTable()

Returns the string table for class ModuleType

protected ModuleType[] getEnumValueTable()

ModuleType

gEnumValueTable

protected ModuleType[] getEnumValueTable()

Returns the enumeration value table for class ModuleType

gEnumValueTable

public String getModuleExtension()

getModuleExtension

public String getModuleExtension()

Return the file extension string for this enumeration.

gEnumValueTable

public static ModuleType getModuleType(int value)

value

gEnumValueTable

public static ModuleType getModuleType(int value)
Return an object of the specified value.

**Parameters:**
- value - a designator for the object.

```java
public String toString()

ModuleType

toString
```

```java
public String toString()

Return the string name of this ModuleType or the integer value if outside the bounds of the table

**Overrides:**
- toString in class Object
```

```java
protected int getOffset()

0 EnumSyntax  0

return
```

```java
getOffset

protected int getOffset()

Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0. If the enumeration class (a subclass of class EnumSyntax) uses integer values starting at other
than 0, override this method in the subclass.

**Returns:**
the offset of the lowest enumeration value.
javax.mail  Class Multipart

javax.lang.Object
  ↳ javax.mail.Multipart

Direct Known Subclasses:
  MimeMultipart

public abstract class Multipart
extends Object

Extended by: MimeMultipart

Multipart Multipart

Multipart  Multipart DataContentHandler  "multipart/signed"
  DataHandler  getContent()  Multipart

  Multipart MIME
"alternative""mixed""related""parallel""signed"

Multipart

  version  1.16, 07/05/04

Multipart is a container that holds multiple body parts. Multipart provides methods to retrieve and set its subparts.

Multipart also acts as the base class for the content object returned by most Multipart DataContentHandlers. For example, invoking getContent() on a DataHandler whose source is a "multipart/signed" data source may return an appropriate subclass of Multipart.

Some messaging systems provide different subtypes of Multiparts. For example, MIME specifies a set of subtypes that include "alternative", "mixed", "related", "parallel", "signed", etc.
Multipart is an abstract class. Subclasses provide actual implementations.

**Version:**
1.16, 07/05/04

**Author:**
John Mani

---

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>contentType</code></td>
<td><code>protected String</code></td>
<td>This field specifies the content-type of this multipart object.</td>
</tr>
<tr>
<td><code>parent</code></td>
<td><code>protected Part</code></td>
<td>The Part containing this Multipart, if known.</td>
</tr>
<tr>
<td><code>parts</code></td>
<td><code>protected Vector</code></td>
<td>Vector of BodyPart objects.</td>
</tr>
</tbody>
</table>

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Multipart()</code></td>
<td>Default constructor.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addBodyPart(BodyPart part)</code></td>
<td>Adds a Part to the multipart.</td>
</tr>
<tr>
<td><code>addBodyPart(BodyPart part, int index)</code></td>
<td>Adds a BodyPart at position index.</td>
</tr>
<tr>
<td><code>getBodyPart(int index)</code></td>
<td>Get the specified Part.</td>
</tr>
<tr>
<td><code>getContentType()</code></td>
<td>Return the content-type of this Multipart.</td>
</tr>
<tr>
<td><code>getCount()</code></td>
<td>Return the number of enclosed BodyPart objects.</td>
</tr>
</tbody>
</table>
### Methods

- **getParent()**
  Return the Part that contains this Multipart object, or null if not known.

- **boolean removeBodyPart(BodyPart part)**
  Remove the specified part from the multipart message.

- **void removeBodyPart(int index)**
  Remove the part at specified location (starting from 0).

- **protected void setMultipartDataSource(MultipartDataSource mp)**
  Setup this Multipart object from the given MultipartDataSource.

- **void setParent(Part parent)**
  Set the parent of this Multipart to be the specified Part.

- **abstract void writeTo(OutputStream os)**
  Output an appropriately encoded bytestream to the given OutputStream.

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

---

### Field Detail

- **parts**

  protected Vector parts

  Vector of BodyPart objects.

- **contentType**
protected `String` `contentType`

This field specifies the content-type of this multipart object. It defaults to "multipart/mixed".

---

parent

protected `Part` `parent`

The `Part` containing this `Multipart`, if known.

**Since:**
JavaMail 1.1

**Constructor Detail**

protected `Multipart()`
`Multipart`

Default constructor. An empty `Multipart` object is created.

**Method Detail**

protected `void` `setMultipartDataSource`(`MultipartDataSource` `mp`) throws `MessagingException`
`MultipartDataSource` `Multipart`
MultipartDataSource BodyPart Multipart
Multipart contentType MultipartDataSource

Multipart IMAP multipart Multipart

\( mp \)          Multipart

**setMultipartDataSource**

protected void setMultipartDataSource(MultipartDataSource mp) throws MessagingException

Set up this Multipart object from the given MultipartDataSource.

The method adds the MultipartDataSource's BodyPart objects into this Multipart. This Multipart's contentType is set to that of the MultipartDataSource.

This method is typically used in those cases where one has a multipart data source that has already been pre-parsed into the individual body parts (for example, an IMAP datasource), but needs to create an appropriate Multipart subclass that represents a specific multipart subtype.

**Parameters:**

- \( mp \) - Multipart datasource

**Throws:**

- MessagingException

---

**public String getContentType()**

Multipart

contentType

return

See also contentType
**getContentType**

```java
public String getContentType()
```

Return the content-type of this Multipart.

This implementation just returns the value of the `contentType` field.

**Returns:**
- `contentType`

**See Also:**
- `contentType`

---

**public int getCount() throws MessagingException**

**BodyPart**

```java
return
```

See also

**getCount**

```java
public int getCount()
```

Throws: `MessagingException`

Return the number of enclosed BodyPart objects.

**Returns:**
- number of parts

**Throws:**
- `MessagingException`

**See Also:**
- `parts`
public **BodyPart** getBodyPart(int index) throws **MessagingException**

**Part**

index	Part
return	Part
Throws	**IndexOutOfBoundsException**:
Throws	**MessagingException**:

---

getBodyPart

public **BodyPart** getBodyPart(int index) throws **MessagingException**

Get the specified Part. Parts are numbered starting at 0.

**Parameters:**

index - the index of the desired Part

**Returns:**

the Part

**Throws:**

**IndexOutOfBoundsException** - if the given index is out of range.
**MessagingException**

---

public boolean removeBodyPart(**BodyPart** part) throws **MessagingException**

**multipart**

part
return	part true false
Throws	**MessagingException**: Part
Throws	**IllegalWriteException**:

---

removeBodyPart

public boolean removeBodyPart(**BodyPart** part)
Remove the specified part from the multipart message. Shifts all the parts after the removed part down one.

**Parameters:**
- **part** - The part to remove

**Returns:**
- true if part removed, false otherwise

**Throws:**
- **MessagingException** - if no such Part exists
- **IllegalWriteException** - if the underlying implementation does not support modification of existing values

---

```java
public void removeBodyPart(int index) throws MessagingException
```

Remove the part at specified location (starting from 0). Shifts all the parts after the removed part down one.

**Parameters:**
- **index** - Index of the part to remove

**Throws:**
- **MessagingException**
- **IndexOutOfBoundsException**
- **IllegalWriteException**

---

```java
public void removeBodyPart(int index) throws MessagingException
```
public void addBodyPart(BodyPart part) throws MessagingException

Part multipartBodyPart Part

    part   Part
     Throws    MessagingException:
    Throws    IllegalWriteException:

addBodyPart

public void addBodyPart(BodyPart part)
    throws MessagingException

    Adds a Part to the multipart. The BodyPart is appended to the list of existing Parts.

Parameters:
    part - The Part to be appended

Throws:
    MessagingException
    IllegalWriteException - if the underlying implementation does not support modification of existing values

addBodyPart

public void addBodyPart(BodyPart part, int index) throws MessagingException

index BodyPart index
BodyPart

    part   BodyPart
    index
     Throws    MessagingException:
    Throws    IllegalWriteException:

addBodyPart
public void addBodyPart(BodyPart part, int index) throws MessagingException

Adds a BodyPart at position index. If index is not the last one in the list, the subsequent parts are shifted up. If index is larger than the number of parts present, the BodyPart is appended to the end.

Parameters:
  part - The BodyPart to be inserted
  index - Location where to insert the part

Throws:
  MessagingException
  IllegalWriteException - if the underlying implementation does not support modification of existing values

abstract public void writeTo(java.io.OutputStream os) throws java.io.IOException, MessagingException

OutputStream

Throws: java.io.IOException: IO
Throws: MessagingException:

writeTo

class public abstract void writeTo(OutputStream os) throws IOException, MessagingException

OutputStream

Throws: IOException - if an IO related exception occurs
  MessagingException:

Output an appropriately encoded bytestream to the given OutputStream. The implementation subclass decides the appropriate encoding algorithm to be used. The bytestream is typically used for sending.

Throws:
  IOException - if an IO related exception occurs
  MessagingException
public Part getParent()

Return the Part that contains this Multipart object, or null if not known.

Since:
JavaMail 1.1

public void setParent(Part parent)

Set the parent of this Multipart to be the specified Part. Normally called by the Message or BodyPart setContent(Multipart) method. parent may be null if the Multipart is being removed from its containing Part.

Since:
JavaMail 1.1
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD

FRAMES  NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
javax.mail Interface MultipartDataSource

All Superinterfaces:  
[DataSource](javax.activation.DataSource)

---

```java
public interface MultipartDataSource
    extends DataSource

Implements: DataSource

MultipartDataSource
    extends DataSource
    implements Datasources, BodyParts, 
    DataContentHandlers

MultipartDataSource

    multipart IMAP
    version 1.8, 07/05/04
    See also javax.activation.DataSource
```

MultipartDataSource is a DataSource that contains body parts. This allows "mail aware" DataContentHandlers to be implemented more efficiently by being aware of such Datasources and using the appropriate methods to access BodyParts.

Note that the data of a MultipartDataSource is also available as an input stream.

This interface will typically be implemented by providers that preparse multipart bodies, for example an IMAP provider.

**Version:**
1.8, 07/05/04

**Author:**
John Mani
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BodyPart.getBodyPart(int index)</td>
<td>Get the specified Part.</td>
</tr>
<tr>
<td>int getCount()</td>
<td>Return the number of enclosed BodyPart objects.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.activation.DataSource
getContentType, getInputStream, getName, getOutputStream

Method Detail

public int getCount()
BodyPart
return

gCount

int getCount()  

Return the number of enclosed BodyPart objects.

Returns:
number of parts

public BodyPart getBodyPart(int index) throws MessagingException
Part

index
Part
return Part
Throws IndexOutOfBoundsException:
Throws MessagingException:

getBodyPart

BodyPart getBodyPart(int index)
   throws MessagingException

Get the specified Part. Parts are numbered starting at 0.

Parameters:
   index - the index of the desired Part

Returns:
   the Part

Throws:
   IndexOutOfBoundsException - if the given index is out of range.
   MessagingException

Overview Package Tree Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface Name

XML URI

<wombat:GetLastTradePrice xmlns:wombat="http://www.wombat.org/trader">

"xmlns" "XML namespace"

- getQualifiedName "prefix:LocalName" = "WOMBAT:GetLastTradePrice"
- getURI "http://www.wombat.org/trader"
- getLocalName "GetLastTracePrice"
- getPrefix "WOMBAT"

XML SOAP

Name SOAPEnvelope.createName URI Name Name

se SOAPEnvelope Name

Name name = se.createName("GetLastTradePrice", "WOMBAT", "http://www.wombat.org/trader");

element addChildElement(name);

See SOAPEnvelope.createName, SOAPFactory.createName

also
A representation of an XML name. This interface provides methods for getting the local and namespace-qualified names and also for getting the prefix associated with the namespace for the name. It is also possible to get the URI of the namespace.

The following is an example of a namespace declaration in an element.

```xml
<wombat:GetLastTradePrice xmlns:wombat="http://www.wombat.org/trader"

("xmlns" stands for "XML namespace").) The following shows what the methods in the Name interface will return.

- `getQualifiedName` will return "prefix:LocalName" = "WOMBAT:GetLastTradePrice"
- `getURI` will return "http://www.wombat.org/trader"
- `getLocalName` will return "GetLastTracePrice"
- `getPrefix` will return "WOMBAT"

XML namespaces are used to disambiguate SOAP identifiers from application-specific identifiers.

`Name` objects are created using the method `SOAPEnvelope.createName`, which has two versions. One method creates `Name` objects with a local name, a namespace prefix, and a namespace URI, and the second creates `Name` objects with just a local name. The following line of code, in which `se` is a `SOAPEnvelope` object, creates a new `Name` object with all three.

```java
Name name = se.createName("GetLastTradePrice", "WOMBAT", "http://www.wombat.org/trader");
```

The following line of code gives an example of how a `Name` object can be used. The variable `element` is a `SOAPElement` object. This code creates a new `SOAPElement` object with the given name and adds it to `element`.

```java
element.addChildElement(name);
```

The `Name` interface may be deprecated in a future release of SAAJ in favor of `javax.xml.namespace.QName`
See Also:  
SOAPEnvelope.createName, SOAPFactory.createName

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getLocalName()</code></td>
<td>Gets the local name part of the XML name that this Name object represents.</td>
</tr>
<tr>
<td><code>getPrefix()</code></td>
<td>Returns the prefix that was specified when this Name object was initialized.</td>
</tr>
<tr>
<td><code>getQualifiedName()</code></td>
<td>Gets the namespace-qualified name of the XML name that this Name object represents.</td>
</tr>
<tr>
<td><code>getURI()</code></td>
<td>Returns the URI of the namespace for the XML name that this Name object represents.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
def getLocalName()
    Name XML
    return
```

**getLocalName**

```java
String getLocalName()
```

Gets the local name part of the XML name that this Name object represents.

**Returns:**

a string giving the local name
public String get Qualified Name()

    Name  XML
    return

getQualifiedName

String getQualified Name()

Gets the namespace-qualified name of the XML name that this Name object represents.

Returns:
the namespace-qualified name as a string

public String getPrefix()

Name  Name  XML
return

getPrefix

String getPrefix()

Returns the prefix that was specified when this Name object was initialized. This prefix is associated with the namespace for the XML name that this Name object represents.

Returns:
the prefix as a string

public String getURI()

Name  XML  URI
return  URI
getURI

String getURI()

Returns the URI of the namespace for the XML name that this Name object represents.

Returns:
the URI as a string
javax.persistence Annotation Type NamedNativeQueries

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface NamedNativeQueries

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

SQL

---

Is used to specify an array of native SQL named queries. Query names are scoped to the persistence unit.

**Since:**
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>NamedNativeQuery[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Array of native SQL named queries</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public NamedNativeQuery[] value()

SQL

value
public abstract NamedNativeQuery[] value

Array of native SQL named queries
### javax.persistence Annotation Type NamedNativeQuery

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface NamedNativeQuery

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

SQL

<table>
<thead>
<tr>
<th>Required Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>query</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>hints</td>
</tr>
<tr>
<td>resultClass</td>
</tr>
<tr>
<td>resultSetMapping</td>
</tr>
</tbody>
</table>
### Element Detail

**abstract public String name()**

The **name** of a `SqlResultSetMapping`, as defined in metadata.

*EntityManager*

#### name

```java
public abstract String name
```

Is used to refer to the query when using the `EntityManager` methods that create query objects.

**abstract public String query()**

The **SQL** query string

#### query

```java
public abstract String query
```

**abstract public QueryHint[] hints()**

#### hints

```java
public abstract QueryHint[] hints
```

Vendor-specific query hints
abstract public Class<T> resultClass()

resultClass

public abstract Class resultClass

The class of the result

Default:

void.class

abstract public String resultSetMapping()

SqlResultSetMapping

resultSetMapping

public abstract String resultSetMapping

The name of a SqlResultSetMapping, as defined in metadata

Default:

""
to license terms.

PS:
javax.persistence Annotation Type NamedQueries

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
public @interface NamedQueries

**Implements**: Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

Java Persistence  

**since** Java Persistence 1.0

Specifies an array of named Java Persistence query language queries. Query names are scoped to the persistence unit.

**Since**:  
Java Persistence 1.0

---

**Required Element Summary**

| NamedQuery[] value | An array of named Java Persistence query language queries. |

---

**Element Detail**

abstract public **NamedQuery[] value**()  
Java Persistence

value
public abstract NamedQuery[] value

An array of named Java Persistence query language queries.
javax.persistence  **Annotation Type NamedQuery**

```java
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface NamedQuery

Implementes: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Java Persistence

Java Persistence

```java
@NamedQuery(
    name="findFirstWithMatchingName",
    query="SELECT c FROM Customer c WHERE c.name LIKE :custName"
)
```

@PersistenceContext
public EntityManager em;
...

customers = em.createNamedQuery("findFirstWithMatchingName")
    .setParameter("custName", "Smith")
    .getResultList();

since  Java Persistence 1.0  en

Is used to specify a named query in the Java Persistence query language, which is a static query expressed in metadata. Query names are scoped to the persistence unit.

The following is an example of the definition of a named query in the Java Persistence query language:

```java
@NamedQuery(
name="findAllCustomersWithName",
query="SELECT c FROM Customer c WHERE c.name LIKE :custName"
)

The following is an example of the use of a named query:

```java
@PersistenceContext
public EntityManager em;
...
customers = em.createNamedQuery("findAllCustomersWithName")
    .setParameter("custName", "Smith")
    .getResultList();
```

Since:
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Refers to the query when using the <a href="https://docs.oracle.com/javaee/7/api/javax/persistence/EntityManager.html">EntityManager</a> methods that create query objects.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>query</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The query string in the Java Persistence query language</td>
</tr>
</tbody>
</table>

### Optional Element Summary

<table>
<thead>
<tr>
<th>QueryHint[]</th>
<th>hints</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Vendor-specific query hints</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String name()

[EntityManager](https://docs.oracle.com/javaee/7/api/javax/persistence/EntityManager.html)
name

public abstract String name

   Refers to the query when using the EntityManager methods that create query objects.

abstract public String query()
Java Persistence

query

public abstract String query

   The query string in the Java Persistence query language

abstract public QueryHint[] hints()

hints

public abstract QueryHint[] hints

   Vendor-specific query hints

   Default:
   {}

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.stream.events Interface Namespace

All Superinterfaces:
   Attribute, XMLEvent, XMLStreamConstants

public interface Namespace
extends Attribute

Implements: Attribute

StartElement

version 1.0

See also javax.xml.stream.eventsStartElement

An interface that contains information about a namespace. Namespaces are accessed from a StartElement.

Version:
   1.0

Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

See Also:
   StartElement

---

Field Summary

Fields inherited from interface
javax.xml.stream.XMLStreamConstants

ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT,
END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE,
NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE,
START_DOCUMENT, START_ELEMENT
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getNameURI()</td>
<td>Gets the uri bound to the prefix of this namespace</td>
</tr>
<tr>
<td>String getPrefix()</td>
<td>Gets the prefix, returns &quot;&quot; if this is a default namespace declaration.</td>
</tr>
<tr>
<td>boolean isDefaultNamespaceDeclaration()</td>
<td>returns true if this attribute declares the default namespace</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.stream.events.Attribute
- getDTDType, getName, getValue, isSpecified

Methods inherited from interface javax.xml.stream.events.XMLEvent
- asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public String getPrefix()

""

getPrefix

String getPrefix()

    Gets the prefix, returns "" if this is a default namespace declaration.
public String getNamespaceURI()

getNamespaceURI

String getNamespaceURI()

Gets the uri bound to the prefix of this namespace

public boolean isDefaultNamespaceDeclaration()

true

isDefaultNamespaceDeclaration

boolean isDefaultNamespaceDeclaration()

returns true if this attribute declares the default namespace

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Summary</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
<tr>
<td>Frames</td>
<td>No frames</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
</tr>
<tr>
<td>No Frames</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.rpc  Class NamespaceConstants

java.lang.Object
   javax.xml.rpc.NamespaceConstants

public class NamespaceConstants

   extends Object

   URI JAX-RPC

   version 1.0

Constants used in JAX-RPC for namespace prefixes and URIs

Version:
   1.0
Author:
   Rahul Sharma

---

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>NSPREFIX_SCHEMA_XSD</td>
<td>Namespace prefix for XML schema XSD</td>
</tr>
<tr>
<td>static String</td>
<td>NSPREFIX_SCHEMA_XSI</td>
<td>Namespace prefix for XML Schema XSI</td>
</tr>
<tr>
<td>static String</td>
<td>NSPREFIX_SOAP_ENCODING</td>
<td>Namespace prefix for SOAP Encoding</td>
</tr>
<tr>
<td>static String</td>
<td>NSPREFIX_SOAP_ENVELOPE</td>
<td>Namespace prefix for SOAP Envelope</td>
</tr>
<tr>
<td>static String</td>
<td>NSURI_SCHEMA_XSD</td>
<td>Namespace URI for XML Schema XSD</td>
</tr>
<tr>
<td>static String</td>
<td>NSURI_SCHEMA_XSI</td>
<td>Namespace URI for XML Schema XSI</td>
</tr>
</tbody>
</table>
Constructor Summary

NamespaceConstants()

Method Summary

Methods inherited from class java.lang.Object
cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

NSPREFIX_SOAP_ENVELOPE

public static final String NSPREFIX_SOAP_ENVELOPE

Namespace prefix for SOAP Envelope

See Also:
Constant Field Values
NSPREFIX_SOAP_ENCODING

public static final String NSPREFIX_SOAP_ENCODING

Namespace prefix for SOAP Encoding

See Also:
  Constant Field Values

NSPREFIX_SCHEMA_XSD

public static final String NSPREFIX_SCHEMA_XSD

Namespace prefix for XML schema XSD

See Also:
  Constant Field Values

NSPREFIX_SCHEMA_XSI

public static final String NSPREFIX_SCHEMA_XSI

Namespace prefix for XML Schema XSI

See Also:
  Constant Field Values

NSURI_SOAP_ENVELOPE

public static final String NSURI_SOAP_ENVELOPE
Nameapace URI for SOAP 1.1 Envelope

See Also:
Constant Field Values

NSURI_SOAP_ENCODING

public static final String NSURI_SOAP_ENCODING

Nameapace URI for SOAP 1.1 Encoding

See Also:
Constant Field Values

NSURI_SOAP_NEXT_ACTOR

public static final String NSURI_SOAP_NEXT_ACTOR

Nameapace URI for SOAP 1.1 next actor role

See Also:
Constant Field Values

NSURI_SCHEMA_XSD

public static final String NSURI_SCHEMA_XSD

Namespace URI for XML Schema XSD

See Also:
Constant Field Values
public static final String NSURI_SCHEMA_XSI

Namespace URI for XML Schema XSI

See Also:
Constant Field Values

Constructor Detail

public NamespaceConstants()

NamespaceConstants

public NamespaceConstants()
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES |
SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |
javax.faces.component  Interface NamingContainer

All Known Implementing Classes:
   HtmlDataTable, HtmlForm, UIData, UIForm, UINamingContainer

public interface NamingContainer

Implemented by: UIData, UIForm, UINamingContainer

NamingContainer  UICOMPONENT  UICOMPONENT#findComponent
UICOMPONENT#getClientId

NamingContainer is an interface that must be implemented by any
UICOMPONENT that wants to be a naming container. Naming containers
affect the behavior of the UICOMPONENT.findComponent(java.lang.String)
and UICOMPONENT.getClientId(javax.faces.context.FacesContext)
methods; see those methods for further information.

---

Field Summary

| SEPARATOR_CHAR | static char | The separator character used in component identifiers
to demarcate navigation to a child naming container. |

Field Detail

SEPARATOR_CHAR
static final char SEPARATOR_CHAR
The separator character used in component identifiers to demarcate navigation to a child naming container.

See Also:

Constant Field Values
<table>
<thead>
<tr>
<th>Summary</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | Constructor | Method |
FRAMES  NO FRAMES
DETAIL: FIELD | Constructor | Method |
A **NavigationHandler** is passed the outcome string returned by an application action invoked for this application, and will use this (along with related state information) to choose the view to be displayed next.

A default implementation of **NavigationHandler** must be provided by the JSF implementation, which will be utilized unless **setNavigationHandler()** is called to establish a different one. This default instance will compare the view identifier of the current view, the specified action binding, and the specified outcome against any navigation rules provided in **faces-config.xml** file(s). If a navigation case matches, the current view will be changed by a call to **FacesContext.setViewRoot()**. Note that a null outcome value will never match any navigation rule, so it can be used as an indicator that the current view should be redisplayed.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>handleNavigation</td>
<td><strong>abstract</strong> <code>void</code> handleNavigation(FacesContext context, String fromAction, String outcome)</td>
<td>Perform navigation processing based on the state information in the specified FacesContext, plus the outcome string returned by an executed application action.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
- wait

Constructor Detail

public NavigationHandler()

NavigationHandler

public NavigationHandler()

Method Detail

abstract public void handleNavigation(FacesContext context, String fromAction, String outcome)

Throws

- NullPointerException: context null

Other methods:
- FacesContext
  - `context`
  - `fromAction`
  - `outcome`
handleNavigation

public abstract void handleNavigation(FacesContext context, String fromAction, String outcome)

Perform navigation processing based on the state information in the specified FacesContext, plus the outcome string returned by an executed application action.

Parameters:
  context - The FacesContext for the current request
  fromAction - The action binding expression that was evaluated to retrieve the specified outcome, or null if the outcome was acquired by some other means
  outcome - The logical outcome returned by a previous invoked application action (which may be null)

Throws:
  NullPointerException - if context is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.internet Class NewsAddress

java.lang.Object
   ↓ javax.mail.Address
   ↓ javax.mail.internet.NewsAddress

All Implemented Interfaces:
   Serializable

public class NewsAddress
    extends Address

Extends: Address

RFC1036

This class models an RFC1036 newsgroup address.

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected String host</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String newsgroup</td>
</tr>
</tbody>
</table>

Constructor Summary

NewsAddress()
   Default constructor.
### NewsAddress

**NewsAddress(String newsgroup)**  
Construct a NewsAddress with the given newsgroup.

**NewsAddress(String newsgroup, String host)**  
Construct a NewsAddress with the given newsgroup and host.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object a)</td>
<td>The equality operator.</td>
</tr>
<tr>
<td>String getHost()</td>
<td>Get the host.</td>
</tr>
<tr>
<td>String getNewsgroup()</td>
<td>Get the newsgroup.</td>
</tr>
<tr>
<td>String getType()</td>
<td>Return the type of this address.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Compute a hash code for the address.</td>
</tr>
<tr>
<td>static NewsAddress[] parse(String newsgroups)</td>
<td>Parse the given comma separated sequence of newsgroup into NewsAddress objects.</td>
</tr>
<tr>
<td>void setHost(String host)</td>
<td>Set the host.</td>
</tr>
<tr>
<td>void setNewsgroup(String newsgroup)</td>
<td>Set the newsgroup.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Convert this address into a RFC 1036 address.</td>
</tr>
<tr>
<td>static String toString(Address[] addresses)</td>
<td>Convert the given array of NewsAddress objects into a comma separated sequence of address strings.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

*clone, finalize, getClass, notify, notifyAll, wait, wait, wait*
Field Detail

newsgroup

protected String newsgroup

host

protected String host

Constructor Detail

public NewsAddress()

NewsAddress

public NewsAddress()

    Default constructor.

public NewsAddress(String newsgroup)

    NewsAddress

    newsgroup

NewsAddress

public NewsAddress(String newsgroup)
Construct a NewsAddress with the given newsgroup.

**Parameters:**
- newsgroup - the newsgroup

```
public NewsAddress(String newsgroup, String host)
```

```
NewsAddress
```

Construct a NewsAddress with the given newsgroup and host.

**Parameters:**
- newsgroup - the newsgroup
- host - the host

---

**Method Detail**

```
public String getType()
```

```
NewsAddress "news"
```

**getType**

```
public String getType()
```

Return the type of this address. The type of a NewsAddress is "news".

**Specified by:**
public void setNewsgroup(String newsgroup)

    newsgroup

setNewsgroup

public void setNewsgroup(String newsgroup)

    Set the newsgroup.

    Parameters:
    newsgroup - the newsgroup

public String getNewsgroup()

    return

getNewsgroup

public String getNewsgroup()

    Get the newsgroup.

    Returns:
    newsgroup

public void setHost(String host)
public void setHost(String host)

Set the host.

Parameters:
  host - the host

public String getHost()

return

public String toString()

RFC 1036

return

public String toString()
Convert this address into a RFC 1036 address.

**Specified by:**
`toString` in class `Address`

**Returns:**
newsgroup

---

**public boolean equals(Object a)**

equals

code: public boolean equals(Object a)

The equality operator.

**Specified by:**
equals in class `Address`

**Parameters:**
a - Address object

---

**public int hashCode()**

hashCode

code: public int hashCode()

Compute a hash code for the address.

**Overrides:**
hashCode in class `Object`
public static String toString(Address[] addresses)
NewsAddress US-ASCII

dates

Throws  ClassCastException, RuntimeException

return

toString

public static String toString(Address[] addresses)

Convert the given array of NewsAddress objects into a comma separated sequence of address strings. The resulting string contains only US-ASCII characters, and hence is mail-safe.

Parameters:
  addresses - array of NewsAddress objects

Returns:
  comma separated address strings

Throws:
  ClassCastException, - if any address object in the given array is not a NewsAddress objects. Note that this is a RuntimeException.

public static NewsAddress[] parse(String newsgroups)
throws AddressException

NewsAddress

newsgroups

return

Throws AddressException:

parse

public static NewsAddress[] parse(String newsgroups)
throws AddressException

Parse the given comma separated sequence of newsgroup into NewsAddress objects.

**Parameters:**

newsgroups - comma separated newsgroup string

**Returns:**

array of NewsAddress objects

**Throws:**

AddressException - if the parse failed

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://example.com/license).

PS :
public interface Node
extends Node

Implements: org.w3c.dom.Node
Implemented by: SOAPElement, SOAPPart, Text

XML DOM Node

A representation of a node (element) in an XML document. This interface extends the standard DOM Node interface with methods for getting and setting the value of a node, for getting and setting the parent of a node, and for removing a node.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void detachNode()</code></td>
<td>Removes this Node object from the tree.</td>
</tr>
<tr>
<td><code>SOAPElement getParentElement()</code></td>
<td>Returns the parent element of this Node object.</td>
</tr>
<tr>
<td><code>String getValue()</code></td>
<td>Returns the value of this node if this is a Text node or the value of the immediate child of this node otherwise.</td>
</tr>
<tr>
<td><code>void recycleNode()</code></td>
<td>Notifies the implementation that this Node object is no longer being used by the application and that the implementation is free to reuse this object for nodes that may be created later.</td>
</tr>
<tr>
<td><code>void setParentElement(SOAPElement parent)</code></td>
<td>Sets the parent of this Node object to the given SOAPElement object.</td>
</tr>
<tr>
<td><code>void setValue(String value)</code></td>
<td>If this is a Text node then this method will set its value, otherwise it sets the value of the immediate (Text) child of this node.</td>
</tr>
</tbody>
</table>

## Methods inherited from interface org.w3c.dom.Node

appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData

## Method Detail
public String getValue()

String getValue()

Returns the value of this node if this is a Text node or the value of the immediate child of this node otherwise. If there is an immediate child of this Node that it is a Text node then it's value will be returned. If there is more than one Text node then the value of the first Text Node will be returned. Otherwise null is returned.

Returns:

- a String with the text of this node if this is a Text node or the text contained by the first immediate child of this Node object that is a Text object if such a child exists; null otherwise.

public void setValue(String value)

void setValue(String value)

If this is a Text node then this method will set its value, otherwise it sets the value of the immediate (Text) child of this node. The value of the immediate child of this node can be set only if, there is one child node and that node is a Text node, or if there are no children in which case a child Text node will be created.
Throws:

[IllegalStateException](#) - if the node is not a [Text](#) node and either has more than one child node or has a child node that is not a Text node.

Since:

SAAJ 1.2

---

**public void setParentElement(**[SOAPElement](#) parent)**

throws [SOAPException](#)

Sets the parent of this Node object to the given SOAPElement object.

**Parameters:**

parent - the SOAPElement object to be set as the parent of this Node object

**Throws:**

[SOAPException](#) - if there is a problem in setting the parent to the given element

**See Also:**

getParentElement()
**getParentElement**

`SOAPElement getParentElement()`

Returns the parent element of this `Node` object. This method can throw an `UnsupportedOperationException` if the tree is not kept in memory.

**Returns:**
the `SOAPElement` object that is the parent of this `Node` object or `null` if this `Node` object is root

**Throws:**
`UnsupportedOperationException` - if the whole tree is not kept in memory

**See Also:**
`setParentElement(javax.xml.soap.SOAPElement)`

---

**public void detachNode()**

`Node`

**detachNode**

`void detachNode()`

Removes this `Node` object from the tree.

---

**public void recycleNode()**

`Node`

`recycleNode` `detachNode`
recycleNode

void recycleNode()

Notifies the implementation that this Node object is no longer being used by the application and that the implementation is free to reuse this object for nodes that may be created later.

Calling the method recycleNode implies that the method detachNode has been called previously.
javax.persistence  **Class NonUniqueResultException**

java.lang.Object  
  └ java.lang.Throwable  
    └ java.lang.Exception  
      └ java.lang.RuntimeException  
        └ javax.persistence.PersistenceException  
          └ javax.persistence.NonUniqueResultException

**All Implemented Interfaces:**  
Serializable

---

```java
public class NonUniqueResultException
    extends PersistenceException
```

**Extends:** Throwable > Exception > RuntimeException > PersistenceException

**getSingleResult**

*since*  
Java Persistence 1.0

**See also**  
getSingleResult()

Thrown by the persistence provider when **getSingleResult()** is executed on a query and there is more than one result from the query. This exception will not cause the current transaction, if one is active, to be marked for roll back.

**Since:**  
Java Persistence 1.0

**See Also:**  
Query.getSingleResult(), Serialized Form

---

**Constructor Summary**

```java
NonUniqueResultException()
```
Constructs a new NonUniqueResultException exception with null as its detail message.

<table>
<thead>
<tr>
<th>NonUniqueResultException(String message)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new NonUniqueResultException exception with the specified detail message.</td>
</tr>
</tbody>
</table>

### Method Summary

#### Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

#### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

```java
class NonUniqueResultException {
    public NonUniqueResultException() {
        null NonUniqueResultException
    }

    public NonUniqueResultException() {
        Constructs a new NonUniqueResultException exception with null as its detail message.
    }

    public NonUniqueResultException(String message) {
        NonUniqueResultException
    }
}
```
NonUniqueResultException

public NonUniqueResultException(String message)

Constructs a new NonUniqueResultException exception with the specified detail message.

Parameters:
message - the detail message.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Class NoResultException

java.lang.Object
   ▼ java.lang.Throwable
      ▼ java.lang.Exception
         ▼ java.lang.RuntimeException
            ▼ javax.persistence.PersistenceException
                ▼ javax.persistence.NoResultException

All Implemented Interfaces:
   Serializable

public class NoResultException
extends PersistenceException

Extends: Throwable > Exception > RuntimeException > PersistenceException

getsingleResult()

since Java Persistence 1.0

See also

getsingleResult()

Thrown by the persistence provider when getsingleResult() is executed on a query and there is no result to return. This exception will not cause the current transaction, if one is active, to be marked for roll back.

Since:
   Java Persistence 1.0

See Also:
   Query.getSingleResult(), Serialized Form

Constructor Summary

NoResultException()

Constructs a new NoResultException exception with null as its
detail message.

**NoResultException(String message)**
Constructs a new NoResultException exception with the specified detail message.

### Method Summary

#### Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

#### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

**public NoResultException()**

null NoResultException

**NoResultException**

**public NoResultException()**

Constructs a new NoResultException exception with null as its detail message.

**public NoResultException(String message)**

NoResultException

message
public NoResultException(String message)

Constructs a new NoResultException exception with the specified detail message.

Parameters:
message - the detail message.
javax.xml.bind.annotation.adapters

Class

NormalizedStringAdapter

java.lang.Object
  ↓ javax.xml.bind.annotation.adapters.XmlAdapter<String, String>
    ↓ javax.xml.bind.annotation.adapters.NormalizedStringAdapter

public final class NormalizedStringAdapter
extends XmlAdapter<String, String>

Extends: "(XmlAdapter

xs:normalizedString    XmlAdapter

"    CR  LF
  since

JAXB 2.0

XmlAdapter to handle xs:normalizedString.

This adapter removes leading and trailing whitespaces, then replace any tab, CR, and LF by a whitespace character '\'.

Since:
  JAXB 2.0

Author:
  Kohsuke Kawaguchi

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NormalizedStringAdapter()</td>
<td></td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected static boolean</td>
<td>isWhiteSpaceExceptSpace(char ch) Returns true if the specified char is a white space character but not 0x20.</td>
</tr>
<tr>
<td>String marshal(String s)</td>
<td>No-op.</td>
</tr>
<tr>
<td>String unmarshal(String text)</td>
<td>Removes leading and trailing whitespaces of the string given as the parameter, then replace any tab, CR, and LF by a whitespace character ''.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public NormalizedStringAdapter()

NormalizedStringAdapter

public NormalizedStringAdapter()

Method Detail

public String unmarshal(String text)

" "CR LF

unmarshal

public String unmarshal(String text)
Removes leading and trailing whitespaces of the string given as the parameter, then replace any tab, CR, and LF by a whitespace character ".

**Specified by:**

`unmarshal` in class `XmlAdapter<String, String>`

**Parameters:**

- `text` - The value to be converted. Can be null.

---

```java
public String marshal(String s)
```

`marshal`

```java
public String marshal(String s)
```

No-op. Just return the same string given as the parameter.

**Specified by:**

`marshal` in class `XmlAdapter<String, String>`

**Parameters:**

- `s` - The value to be converted. Can be null.

---

```java
protected static boolean isWhiteSpaceExceptSpace(char ch)
```

`isWhiteSpaceExceptSpace`

```java
protected static boolean isWhiteSpaceExceptSpace(char ch)
```

Returns true if the specified char is a white space character but not 0x20.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  Class NoSuchEJBException

java.lang.Object  
  ↑  java.lang.Throwable  
  ↑  java.lang.Exception  
  ↑  java.lang.RuntimeException  
  ↑  javax.ejb.EJBException  
  ↑  javax.ejb.NoSuchEJBException

All Implemented Interfaces:  Serializable

public class NoSuchEJBException

extends EJBException

Extends:  Throwable > Exception > RuntimeException > EJBException

NoSuchEJBException

A NoSuchEJBException is thrown if an attempt is made to invoke a method on an object that no longer exists.

See Also:  Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>NoSuchEJBException</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>()</td>
<td>Constructs a NoSuchEJBException with no detail message.</td>
<td></td>
</tr>
<tr>
<td>(String message)</td>
<td>Constructs a NoSuchEJBException with the specified detail message.</td>
<td></td>
</tr>
<tr>
<td>(String message, Exception ex)</td>
<td>Constructs a NoSuchEJBException with the specified detail message and a nested exception.</td>
<td></td>
</tr>
</tbody>
</table>
Method Summary

Methods inherited from class javax.ejb.EJBException
getCausedByException, getMessage, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public NoSuchEJBException()
NoSuchEJBException

NoSuchEJBException

public NoSuchEJBException()

Constructs a NoSuchEJBException with no detail message.

public NoSuchEJBException(String message)
NoSuchEJBException

NoSuchEJBException
public NoSuchEJBException(String message)

Constructs a NoSuchEJBException with the specified detail message.

public NoSuchEJBException(String message, Exception ex)
NoSuchEJBException

NoSuchEJBException

public NoSuchEJBException(String message, Exception ex)

Constructs a NoSuchEJBException with the specified detail message and a nested exception.
javax.ejb

Class NoSuchEntityException

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
      └ java.lang.RuntimeException
        └ javax.ejb.EJBException
          └ javax.ejb.NoSuchEntityException

All Implemented Interfaces:
  Serializable

public class NoSuchEntityException extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

Bean NoSuchEntityException

Bean Bean ejbLoad ejbStore

The NoSuchEntityException exception is thrown by an Entity Bean instance to its container to report that the invoked business method or callback method could not be completed because of the underlying entity was removed from the database.

This exception may be thrown by the bean class methods that implement the business methods defined in the bean's component interface; and by the ejbLoad and ejbStore methods.

See Also:
  Serialized Form

Constructor Summary
NoSuchEntityException()  
    Constructs a NoSuchEntityException with no detail message.

NoSuchEntityException( Exception ex)  
    Constructs a NoSuchEntityException that embeds the originally thrown exception.

NoSuchEntityException( String message)  
    Constructs a NoSuchEntityException with the specified detailed message.

Method Summary

Methods inherited from class javax.ejb.EJBException  
getCausedByException, getMessage, printStackTrace, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable  
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object  
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public NoSuchEntityException()  
    NoSuchEntityException

NoSuchEntityException  

public NoSuchEntityException()  
    Constructs a NoSuchEntityException with no detail message.
public NoSuchEntityException(String message)

NoSuchEntityException

public NoSuchEntityException(String message)

Constructs a NoSuchEntityException with the specified detailed message.

public NoSuchEntityException(Exception ex)

NoSuchEntityException

public NoSuchEntityException(Exception ex)

Constructs a NoSuchEntityException that embeds the originally thrown exception.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.ejb Class NoSuchObjectLocalException

java.lang.Object
   └ java.lang.Throwable
       └ java.lang.Exception
           └ java.lang.RuntimeException
               └ javax.ejb.EJBException
                   └ javax.ejb.NoSuchObjectLocalException

All Implemented Interfaces:
   Serializable

public class NoSuchObjectLocalException

extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

NoSuchObjectLocalException

A NoSuchObjectLocalException is thrown if an attempt is made to invoke a method on an object that no longer exists.

See Also:
   Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>NoSuchObjectLocalException()</td>
</tr>
<tr>
<td>Constructs a NoSuchObjectLocalException with no detail message.</td>
</tr>
<tr>
<td>NoSuchObjectLocalException(String message)</td>
</tr>
<tr>
<td>Constructs a NoSuchObjectLocalException with the specified detail message.</td>
</tr>
<tr>
<td>NoSuchObjectLocalException(String message, Exception ex)</td>
</tr>
<tr>
<td>Constructs a NoSuchObjectLocalException with the specified message and exception.</td>
</tr>
</tbody>
</table>
detail message and a nested exception.

Method Summary

Methods inherited from class javax.ejb.EJBEException
getCausedByException, getMessage, printStackTrace, printStackTrace

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public NoSuchObjectLocalException()
NoSuchObjectLocalException

NoSuchObjectLocalException

public NoSuchObjectLocalException()

Constructs a NoSuchObjectLocalException with no detail message.

public NoSuchObjectLocalException(String message)
NoSuchObjectLocalException

NoSuchObjectLocalException
public NoSuchObjectLocalException(String message)

    Constructs a NoSuchObjectLocalException with the specified detail message.

public NoSuchObjectLocalException(String message, Exception ex)
NoSuchObjectLocalException

NoSuchObjectLocalException

public NoSuchObjectLocalException(String message, Exception ex)

    Constructs a NoSuchObjectLocalException with the specified detail message and a nested exception.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail Class NoSuchProviderException

java.lang.Object
   └ java.lang.Throwable
       └ java.lang.Exception
           └ javax.mail.MessagingException
               └ javax.mail.NoSuchProviderException

All Implemented Interfaces:
   Serializable

public class NoSuchProviderException

extends MessagingException

Extends: Throwable > Exception > MessagingException

Session Provider

This exception is thrown when Session attempts to instantiate a Provider that doesn't exist.

Author:
   Max Spivak

See Also:
   Serialized Form

Constructor Summary

NoSuchProviderException() Constructor.

NoSuchProviderException(String message) Constructor.

Method Summary
Methods inherited from class javax.mail.
\texttt{MessagingException}
\begin{itemize}
\item \texttt{getCause}, \texttt{getNextException}, \texttt{setNextException}, \texttt{toString}
\end{itemize}

Methods inherited from class java.lang.\texttt{Throwable}
\begin{itemize}
\item \texttt{fillInStackTrace}, \texttt{getLocalizedMessage}, \texttt{getMessage}, \texttt{getStackTrace}, \texttt{initCause}, \texttt{printStackTrace}, \texttt{printStackTrace}, \texttt{printStackTrace}, \texttt{setStackTrace}
\end{itemize}

Methods inherited from class java.lang.\texttt{Object}
\begin{itemize}
\item \texttt{clone}, \texttt{equals}, \texttt{finalize}, \texttt{getClass}, \texttt{hashCode}, \texttt{notify}, \texttt{notifyAll}, \texttt{wait}, \texttt{wait}, \texttt{wait}
\end{itemize}

\section*{Constructor Detail}

\begin{verbatim}
public NoSuchProviderException()

NoSuchProviderException

class NoSuchProviderException

public NoSuchProviderException()

    Constructor.

public NoSuchProviderException(String message)

    message

NoSuchProviderException

public NoSuchProviderException(String message)
Constructor.

**Parameters:**
message - The detailed error message

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.xml.stream.events Interface NotationDeclaration

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface NotationDeclaration
extends XMLEvent

Implements: XMLEvent

   publicId   systemId   null
            version     1.0

An interface for handling Notation Declarations Receive notification of a notation declaration event. It is up to the application to record the notation for later reference, At least one of publicId and systemId must be non-null. There is no guarantee that the notation declaration will be reported before any unparsed entities that use it.

Version:
   1.0
Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

Fields inherited from interface
ejavax.xml.stream.XMLStreamConstants

ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getName()</td>
<td>The notation name.</td>
</tr>
<tr>
<td>getPublicId()</td>
<td>The notation's public identifier, or null if none was given.</td>
</tr>
<tr>
<td>getSystemId()</td>
<td>The notation's system identifier, or null if none was given.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.stream.events.XMLEvent

asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public String getName()

getName

String getName()

   The notation name.

public String getPublicId()

   getPublicId

   null
**String** getPublicId()

The notation's public identifier, or null if none was given.

**public String getSystemId()**
null

**getSystemId**

**String** getSystemId()

The notation's system identifier, or null if none was given.
**javax.xml.bind**  
**Interface NotIdentifiableEvent**

**All Superinterfaces**:  
*ValidationEvent*

**All Known Implementing Classes**:  
*NotIdentifiableEventImpl*

```java
public interface NotIdentifiableEvent extends ValidationEvent

Implements: ValidationEvent  
Implemented by: NotIdentifiableEventImpl
```

**ID/IDREF**

- **version**  
  $Revision: 1.1 $

- **since**  
  JAXB1.0

- **See**  
  [javax.xml.bind.Validator](http://docs.oracle.com/javase/7/docs/api/javax/xml/bind/Validator.html),  
  [javax.xml.bind.ValidationEventHandler](http://docs.oracle.com/javase/7/docs/api/javax/xml/bind/ValidationEventHandler.html)

**This event indicates that a problem was encountered resolving an ID/IDREF.**

**Since:**  
JAXB1.0

**Version:**  
$Revision: 1.1 $

**Author:**

- Ryan Shoemaker, Sun Microsystems, Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems, Inc.

**See Also:**  
[Validator](http://docs.oracle.com/javase/7/docs/api/javax/xml/bind/Validator.html), [ValidationEventHandler](http://docs.oracle.com/javase/7/docs/api/javax/xml/bind/ValidationEventHandler.html)
### Field Summary

Fields inherited from interface javax.xml.bind.**ValidationEvent**

ERROR, FATAL_ERROR, WARNING

### Method Summary

Methods inherited from interface javax.xml.bind.**ValidationEvent**

getLinkedException, getLocator, getMessage, getSeverity

---

**Overview** | **Package** | **Tree** | **Deprecated** | **Index** | **Help**
---|---|---|---|---|---
PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.xml.bind.helpers Class NotIdentifiableEventImpl

java.lang.Object
   ▼javax.xml.bind.helpers.ValidationEventImpl
      ▼javax.xml.bind.helpers.NotIdentifiableEventImpl

All Implemented Interfaces:
   NotIdentifiableEvent, ValidationEvent

public class NotIdentifiableEventImpl

extends ValidationEventImpl
implements NotIdentifiableEvent

Extends: ValidationEventImpl
Implements: NotIdentifiableEvent

NotIdentifiableEvent

JAXB ValidationEvent

version $Revision: 1.1 $ en
since JAXB1.0

javax.xml.bind.NotIdentifiableEvent,
javax.xml.bind.Validator,
javax.xml.bind.ValidationEventHandler,
javax.xml.bind.ValidationEvent,
javax.xml.bind.ValidationEventLocator

Default implementation of the NotIdentifiableEvent interface.

JAXB providers are allowed to use whatever class that implements the ValidationEvent interface. This class is just provided for a convenience.

Since:
   JAXB1.0
**Field Summary**

Fields inherited from interface javax.xml.bind.**ValidationEvent**

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR, FATAL_ERROR, WARNING</td>
</tr>
</tbody>
</table>

**Constructor Summary**

**NotIdentifiableEventImpl**(int _severity, String _message, ValidationEventLocator _locator)

Create a new NotIdentifiableEventImpl.

**NotIdentifiableEventImpl**(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)

Create a new NotIdentifiableEventImpl.

**Method Summary**

Methods inherited from class javax.xml.bind.helpers.**ValidationEventImpl**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>getLinkedException, getLocator, getMessage, getSeverity, setLinkedException, setLocator, setMessage, setSeverity, toString</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.bind.**ValidationEvent**
public NotIdentifiableEventImpl(int _severity, String _message, ValidationEventLocator _locator)  
NotIdentifiableEventImpl

_severity     severity  ValidationEvent.WARNING
_message      ValidationEvent.ERROR  ValidationEvent.FATAL_ERROR
_locator      Locator  null

Throws  IllegalArgument Exception: severity

Create a new NotIdentifiableEventImpl.

Parameters:
_severity - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR
_message - The text message for this event - may be null.
_locator - The locator object for this event - may be null.

Throws:  IllegalArgument Exception - if an illegal severity field is supplied

public NotIdentifiableEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)  
NotIdentifiableEventImpl
NotIdentifiableEventImpl

public NotIdentifiableEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)

Create a new NotIdentifiableEventImpl.

Parameters:

_severity - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR

_message - The text message for this event - may be null.

_locator - The locator object for this event - may be null.

_linkedException - An optional linked exception that may provide additional information about the event - may be null.

Throws:

IllegalArgumentException - if an illegal severity field is supplied
PS:
**Class** NotSupportedException

---

```
javax.resource
java.lang.Object
   ↘ java.lang.Throwable
      ↘ java.lang.Exception
         ↘ javax.resource.ResourceException
            ↘ javax.resource.NotSupportedException
```

**All Implemented Interfaces:**

* Serializable

---

```
public class NotSupportedException
extends ResourceException
```

**Extends:** Throwable > Exception > ResourceException

---

<table>
<thead>
<tr>
<th>NotSupportedException</th>
<th>NO_TRANSACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>ManagedConnection</td>
<td>getXAResource</td>
</tr>
<tr>
<td></td>
<td>version</td>
</tr>
</tbody>
</table>

---

A `NotSupportedException` is thrown to indicate that callee (resource adapter or application server for system contracts) cannot execute an operation because the operation is not a supported feature. For example, if the transaction support level for a resource adapter is `NO_TRANSACTION`, an invocation of `getXAResource` method on a `ManagedConnection` object should throw a `NotSupportedException` exception.

---

**Version:**

1.0

**Author:**

Rahul Sharma, Ram Jeyaraman

**See Also:**

* Serialized Form
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>NotSupportedException()</code></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><code>NotSupportedException(String message)</code></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><code>NotSupportedException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and error code.</td>
</tr>
<tr>
<td><code>NotSupportedException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>NotSupportedException( Throwable cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class `javax.resource.ResourceException`
- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`

### Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

```java
public NotSupportedException()
```

Constructs a new instance with null as its detail message.
null

NotSupportedException

public RuntimeException()

Constructs a new instance with null as its detail message.

public RuntimeException(String message)

message

NotSupportedException

public RuntimeException(String message)

message

Constructs a new instance with the specified detail message.

Parameters:

message - the detail message.

public RuntimeException(Throwable cause)

cause throwable

cause

NotSupportedException

public RuntimeException(Throwable cause)

cause

Constructs a new throwable with the specified cause.

Parameters:

cause - a chained exception of type Throwable.
public NotSupportedException(String message, Throwable cause)
        cause throwable
        message
        cause
        Throwable

NotSupportedException

public NotSupportedException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public NotSupportedException(String message, String errorCode)
        Throwable
        message
        errorCode

NotSupportedException

public NotSupportedException(String message, String errorCode)

Constructs a new throwable with the specified detail message and error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
javax.transaction Class NotSupportedException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
           └ javax.transaction.NotSupportedException

All Implemented Interfaces:
  Serializable

public class NotSupportedException
  extends Exception

Extends: Throwable > Exception

NotSupportedException Transaction Manager

NotSupportedException exception indicates that the request cannot be executed because the operation is not a supported feature. For example, because nested transactions are not supported, the Transaction Manager throws this exception when a calling thread that is already associated with a transaction attempts to start a new transaction. (A nested transaction occurs when a thread is already associated with one transaction and attempts to start a second transaction.)

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotSupportedException()</td>
</tr>
<tr>
<td>NotSupportedException(String msg)</td>
</tr>
</tbody>
</table>
Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public NotSupportedException()

NotSupportedException

public NotSupportedException()

public NotSupportedException(String msg)

NotSupportedException

public NotSupportedException(String msg)
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.search Class NotTerm

java.lang.Object
   \ javax.mail.search.SearchTerm
   \ javax.mail.search.NotTerm

All Implemented Interfaces:
   Serializable

public final class NotTerm
extends searchTerm

Extends: SearchTerm

NEGATION

This class implements the logical NEGATION operator.

Author:
   Bill Shannon, John Mani

See Also:
   Serialized Form

Field Summary

*protected searchTerm term*
   The search term to negate.

Constructor Summary

*NotTerm*(SearchTerm t)
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>SearchTerm getTerm()</td>
<td>Return the term to negate.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean match(Message msg)</td>
<td>This method applies a specific match criterion to the given message and returns the result.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

- clone, finalize, getClass, notify, notifyAll, toString, wait, wait

## Field Detail

protected `SearchTerm` term

The search term to negate.

## Constructor Detail

public `NotTerm(SearchTerm t)`
public SearchTerm getTerm()

getTerm

public SearchTerm getTerm()

    Return the term to negate.

public boolean match(Message msg)

match

public boolean match(Message msg)

    Description copied from class: SearchTerm
    This method applies a specific match criterion to the given message and returns the result.

    Specified by:
        match in class SearchTerm
    Parameters:
        msg - The match criterion is applied on this message
    Returns:
        true, if the match succeeds, false if the match fails

public boolean equals(Object obj)

equals
public boolean equals(Object obj)

Equality comparison.

**Overrides:**

equals in class Object

---

public int hashCode()

hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

**Overrides:**

hashCode in class Object

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.convert Class NumberConverter

java.lang.Object
  | javax.faces.convert.NumberConverter

All Implemented Interfaces:
  StateHolder, Converter

public class NumberConverter
  extends Object
  implements Converter, StateHolder

  Implements: Converter, StateHolder

java.lang.Number   Converter
getAsObject() String    java.lang.Double    java.lang.Long

  • String  null    null
  • String  0       null
  • locale null   Locale   UIViewRoot    Locale
  • pattern java.text.DecimalFormat    type
  • pattern type java.text.NumberFormat
    getCurrencyInstance()    getNumberInstance()    getPercentInstance()
    Locale
  • integerOnly true String    java.text.NumberFormat
    setParseIntegerOnly()    JavaDoc

getAsString() java.lang.Number    String

  • null  0    String
  • String
  • locale null   Locale   FacesContext    Locale
  • pattern java.text.DecimalFormat    type
  • pattern type java.text.NumberFormat
    getCurrencyInstance()    getNumberInstance()    getPercentInstance()
Locale
- groupingUsed true
- setGroupingUsed(true)
- maxFractionDigits
- maxIntegerDigits
- minFractionDigits
- minIntegerDigits
- currency
- currencyCode
- currencySymbol
- currencyCode  JDK 1.4JVM
- currencySymbol

**Converter** implementation for `java.lang.Number` values.

The `getAsObject()` method parses a String into an `java.lang.Double` or `java.lang.Long`, according to the following algorithm:

- If the specified String is null, return a `null`. Otherwise, trim leading and trailing whitespace before proceeding.
- If the specified String - after trimming - has a zero length, return `null`.
- If the `locale` property is not null, use that `Locale` for managing parsing. Otherwise, use the `Locale` from the `UIViewRoot`.
- If a pattern has been specified, its syntax must conform the rules specified by `java.text.DecimalFormat`. Such a pattern will be used to parse, and the `type` property will be ignored.
- If a pattern has not been specified, parsing will be based on the `type` property, which expects a currency, a number, or a percent. The parse pattern for currencies, numbers, and percentages is determined by calling the `getCurrencyInstance()`, `getNumberInstance()`, or `getPercentInstance()` method of the `java.text.NumberFormat` class, passing in the selected `Locale`.
- If the `integerOnly` property has been set to true, only the integer portion of the String will be parsed. See the JavaDocs for the `setParseIntegerOnly()` method of the `java.text.NumberFormat` class for more information.

The `getAsString()` method expects a value of type `java.lang.Number` (or a subclass), and creates a formatted String according to the following algorithm:
If the specified value is null, return a zero-length String.
If the specified value is a String, return it unmodified.
If the locale property is not null, use that Locale for managing formatting. Otherwise, use the Locale from the FacesContext.
If a pattern has been specified, its syntax must conform the rules specified by java.text.DecimalFormat. Such a pattern will be used to format, and the type property (along with related formatting options described in the next paragraph) will be ignored.
If a pattern has not been specified, formatting will be based on the type property, which formats the value as a currency, a number, or a percent. The format pattern for currencies, numbers, and percentages is determined by calling the percentages is determined by calling the getCurrencyInstance(), getNumberInstance(), or getPercentInstance() method of the java.text.NumberFormat class, passing in the selected Locale. In addition, the following properties will be applied to the format pattern, if specified:
  - If the groupingUsed property is true, the setGroupingUsed(true) method on the corresponding NumberFormat instance will be called.
  - The minimum and maximum number of digits in the integer and fractional portions of the result will be configured based on any values set for the maxFractionDigits, maxIntegerDigits, minFractionDigits, and minIntegerDigits properties.
  - If the type is set to currency, it is also possible to configure the currency symbol to be used, using either the currencyCode or currencySymbol properties. If both are set, the value for currencyCode takes precedence on a JDK 1.4 (or later) JVM; otherwise, the value for currencySymbol takes precedence.

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CONVERTER_ID</strong></td>
<td>The standard converter id for this converter.</td>
</tr>
<tr>
<td><strong>CURRENCY_ID</strong></td>
<td>The message identifier of the FacesMessage to be created if the conversion to Number fails.</td>
</tr>
<tr>
<td><strong>NUMBER_ID</strong></td>
<td></td>
</tr>
</tbody>
</table>
static String PATTERN_ID
The message identifier of the FacesMessage to be created if the conversion to Number fails.

static String PERCENT_ID
The message identifier of the FacesMessage to be created if the conversion to Number fails.

static String STRING_ID
The message identifier of the FacesMessage to be created if the conversion of the Number Value to String fails.

Constructor Summary
NumberConverter()

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object getAsObject(FacesContext, UIComponent, String)</td>
<td>Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.</td>
</tr>
<tr>
<td>String getAsString(FacesContext, UIComponent, Object)</td>
<td>Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.</td>
</tr>
<tr>
<td>String getCurrencyCode()</td>
<td>Return the ISO 4217 currency code used by getAsString() with a type of currency.</td>
</tr>
<tr>
<td>String getCurrencySymbol()</td>
<td>Return the currency symbol used by getAsString() with a type of currency.</td>
</tr>
<tr>
<td>Method Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getLocale()</code></td>
<td>Return the <code>Locale</code> to be used when parsing numbers.</td>
</tr>
<tr>
<td><code>getMaxFractionDigits()</code></td>
<td>Return the maximum number of digits <code>getAsString()</code> should render in the fraction portion of the result.</td>
</tr>
<tr>
<td><code>getMaxIntegerDigits()</code></td>
<td>Return the maximum number of digits <code>getAsString()</code> should render in the integer portion of the result.</td>
</tr>
<tr>
<td><code>getMinFractionDigits()</code></td>
<td>Return the minimum number of digits <code>getAsString()</code> should render in the fraction portion of the result.</td>
</tr>
<tr>
<td><code>getMinIntegerDigits()</code></td>
<td>Return the minimum number of digits <code>getAsString()</code> should render in the integer portion of the result.</td>
</tr>
<tr>
<td><code>getPattern()</code></td>
<td>Return the format pattern to be used when formatting and parsing numbers.</td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Return the number type to be used when formatting and parsing numbers.</td>
</tr>
<tr>
<td><code>isGroupingUsed()</code></td>
<td>Return true if <code>getAsString</code> should include grouping separators if necessary.</td>
</tr>
<tr>
<td><code>isIntegerOnly()</code></td>
<td>Return true if only the integer portion of the given value should be returned from <code>getAsObject()</code>.</td>
</tr>
<tr>
<td><code>isTransient()</code></td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td><code>restoreState(FacesContext context, Object state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>saveState(FacesContext context)</code></td>
<td>Gets the state of the instance as a <code>Serializable Object</code>.</td>
</tr>
</tbody>
</table>
| `setCurrencyCode(String currencyCode)` | }
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void</code></td>
<td>Set the ISO 4217 currency code used by <code>getAsString()</code> with a type of currency.</td>
</tr>
<tr>
<td><code>setCurrencySymbol(String currencySymbol)</code></td>
<td>Set the currency symbol used by <code>getAsString()</code> with a type of currency.</td>
</tr>
<tr>
<td><code>setGroupingUsed(boolean groupingUsed)</code></td>
<td>Set the flag indicating whether <code>getAsString()</code> should include grouping separators if necessary.</td>
</tr>
<tr>
<td><code>setIntegerOnly(boolean integerOnly)</code></td>
<td>Set to true if only the integer portion of the given value should be returned from <code>getAsObject()</code>.</td>
</tr>
<tr>
<td><code>setLocale(Locale locale)</code></td>
<td>Set the Locale to be used when parsing numbers.</td>
</tr>
<tr>
<td><code>setMaxFractionDigits(int maxFractionDigits)</code></td>
<td>Set the maximum number of digits <code>getAsString()</code> should render in the fraction portion of the result.</td>
</tr>
<tr>
<td><code>setMaxIntegerDigits(int maxIntegerDigits)</code></td>
<td>Set the maximum number of digits <code>getAsString()</code> should render in the integer portion of the result.</td>
</tr>
<tr>
<td><code>setMinFractionDigits(int minFractionDigits)</code></td>
<td>Set the minimum number of digits <code>getAsString()</code> should render in the fraction portion of the result.</td>
</tr>
<tr>
<td><code>setMinIntegerDigits(int minIntegerDigits)</code></td>
<td>Set the minimum number of digits <code>getAsString()</code> should render in the integer portion of the result.</td>
</tr>
<tr>
<td><code>setPattern(String pattern)</code></td>
<td>Set the format pattern to be used when formatting and parsing numbers.</td>
</tr>
<tr>
<td><code>setTransient(boolean transientFlag)</code></td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
<tr>
<td><code>setType(String type)</code></td>
<td>Set the number type to be used when formatting and parsing numbers.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

CONVERTER_ID

public static final String CONVERTER_ID

The standard converter id for this converter.

See Also:
Constant Field Values

CURRENCY_ID

public static final String CURRENCY_ID

The message identifier of the FacesMessage to be created if the conversion to Number fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also:
Constant Field Values
NUMBER_ID

public static final String NUMBER_ID

The message identifier of the FacesMessage to be created if the conversion to Number fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also: Constant Field Values

PATTERN_ID

public static final String PATTERN_ID

The message identifier of the FacesMessage to be created if the conversion to Number fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a String whose value is the label of the input component that produced this message.

See Also: Constant Field Values
PERCENT_ID

public static final String PERCENT_ID

    The message identifier of the FacesMessage to be created if the conversion to Number fails. The message format string for this message may optionally include the following placeholders:

    • {0} replaced by the unconverted value.
    • {1} replaced by an example value.
    • {2} replaced by a String whose value is the label of the input component that produced this message.

See Also:
    Constant Field Values

STRING_ID

public static final String STRING_ID

    The message identifier of the FacesMessage to be created if the conversion of the Number value to String fails. The message format string for this message may optionally include the following placeholders:

    • {0} replaced by the unconverted value.
    • {1} replaced by a String whose value is the label of the input component that produced this message.

See Also:
    Constant Field Values

<table>
<thead>
<tr>
<th>Constructor Detail</th>
</tr>
</thead>
</table>

    public NumberConverter()
NumberConverter

class NumberConverter

Method Detail

public String getCurrencyCode()

type currency getAsString() ISO 4217 Locale

getCurrencyCode

public String getCurrencyCode()

Return the ISO 4217 currency code used by getAsString() with a type of currency. If not set, the value used will be based on the formatting Locale.

public void setCurrencyCode(String currencyCode)

type currency getAsString() ISO 4217

currencyCode

setCurrencyCode

public void setCurrencyCode(String currencyCode)

Set the ISO 4217 currency code used by getAsString() with a type of currency.

Parameters:
getCurrencySymbol

public String getCurrencySymbol()

    type currency getAsString() Locale

getCurrencySymbol

public String getCurrencySymbol()

    Return the currency symbol used by getAsString() with a type of currency. If not set, the value used will be based on the formatting Locale.

setCurrencySymbol

public void setCurrencySymbol(String currencySymbol)

    type currency getAsString()

                currencySymbol

setCurrencySymbol

public void setCurrencySymbol(String currencySymbol)

    Set the currency symbol used by getAsString() with a type of currency.

    Parameters:
                currencySymbol - The new currency symbol

public boolean isGroupingUsed()
isGroupingUsed

public boolean isGroupingUsed()

Return true if getAsString should include grouping separators if necessary. If not modified, the default value is true.

public void setGroupingUsed(boolean groupingUsed)

gAsString()

  groupingUsed

setGroupingUsed

public void setGroupingUsed(boolean groupingUsed)

  Set the flag indicating whether getAsString() should include grouping separators if necessary.

  Parameters:
    groupingUsed - The new grouping used flag

public boolean isIntegerOnly()

  getAsObject() true false

isIntegerOnly
public boolean isIntegerOnly()

    Return true if only the integer portion of the given value should be returned from getAsObject(). If not modified, the default value is false.

public void setIntegerOnly(boolean integerOnly)

    getAsObject() true

    integerOnly integer-only

setIntegerOnly

public void setIntegerOnly(boolean integerOnly)

    Set to true if only the integer portion of the given value should be returned from getAsObject().

    Parameters:
        integerOnly - The new integer-only flag

public int getMaxFractionDigits()

    getAsString()

getMaxFractionDigits

public int getMaxFractionDigits()

    Return the maximum number of digits getAsString() should render in the fraction portion of the result.
public void setMaxFractionDigits(int maxFractionDigits)

    getMaxString()

    maxFractionDigits

setMaxFractionDigits

public void setMaxFractionDigits(int maxFractionDigits)

    Set the maximum number of digits getMaxString() should render in the fraction portion of the result. If not set, the number of digits depends on the value being converted.

    Parameters:
    maxFractionDigits - The new limit

public int getMaxIntegerDigits()

    getMaxString()

getMaxIntegerDigits

public int getMaxIntegerDigits()

    Return the maximum number of digits getMaxString() should render in the integer portion of the result.

public void setMaxIntegerDigits(int maxIntegerDigits)

    getMaxString()

    maxIntegerDigits
setMaxIntegerDigits

public void setMaxIntegerDigits(int maxIntegerDigits)

Set the maximum number of digits getAsString() should render in the integer portion of the result. If not set, the number of digits depends on the value being converted.

Parameters:
maxIntegerDigits - The new limit

public int getMinFractionDigits()

getAsString()

getMinFractionDigits

public int getMinFractionDigits()  

Return the minimum number of digits getAsString() should render in the fraction portion of the result.

public void setMinFractionDigits(int minFractionDigits)

getAsString()

   minFractionDigits

setMinFractionDigits

public void setMinFractionDigits(int minFractionDigits)
Set the minimum number of digits `getAsString()` should render in the fraction portion of the result. If not set, the number of digits depends on the value being converted.

**Parameters:**

- `minFractionDigits` - The new limit

---

### public int getMinIntegerDigits()

`getAsString()`

---

#### getMinIntegerDigits

**public int getMinIntegerDigits()**

Return the minimum number of digits `getAsString()` should render in the integer portion of the result.

---

### public void setMinIntegerDigits(int minIntegerDigits)

`getAsString()`

#### setMinIntegerDigits

**public void setMinIntegerDigits(int minIntegerDigits)**

Set the minimum number of digits `getAsString()` should render in the integer portion of the result. If not set, the number of digits depends on the value being converted.

**Parameters:**
public java.util.Locale getLocale()

Locale null javax.faces.component.UIViewRoot Locale

getLocale

public Locale getLocale()

Return the Locale to be used when parsing numbers. If this value is null, the Locale stored in the UIViewRoot for the current request will be utilized.

public void setLocale(java.util.Locale locale)

Locale null javax.faces.component.UIViewRoot Locale

locale Locale null

setLocale

public void setLocale(Locale locale)

Set the Locale to be used when parsing numbers. If set to null, the Locale stored in the UIViewRoot for the current request will be utilized.

Parameters:
locale - The new Locale (or null)

public String getPattern()
**getPattern**

public String getPattern()

    Return the format pattern to be used when formatting and parsing numbers.

---

**public void setPattern(String pattern)**

    java.text.DecimalFormat getAsObject()
    getAsString() ConverterException

    pattern

**setPattern**

public void setPattern(String pattern)

    Set the format pattern to be used when formatting and parsing numbers. Valid values are those supported by java.text.DecimalFormat. An invalid value will cause a ConverterException when getAsObject() or getAsString() is called.

    Parameters:
    pattern - The new format pattern

---

**public String getType()**

    number
**getType**

public `String` `getType()`

Return the number type to be used when formatting and parsing numbers. If not modified, the default type is `number`.

**public void setType(String type)**

currenycurrency number percent getAsObject() getAsString()

`ConverterException`

`type`

**setType**

public void `setType(String type)`

Set the number type to be used when formatting and parsing numbers. Valid values are `currency`, `number`, or `percent`. An invalid value will cause a `ConverterException` when `getAsObject()` or `getAsString()` is called.

**Parameters:**

- `type` - The new number style

**public Object getAsObject(FacesContext context, UIComponent component, String value)**

Throws `ConverterException`: `NullPointerException`

Throws `NullPointerException`: `NullPointerException` context

`component null`

**getAsObject**
public Object getAsObject(FacesContext context, 
    UIComponent component, 
    String value)

Description copied from interface: Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

Specified by: 
    getAsObject in interface Converter

Parameters:
    context - FacesContext for the request being processed
    component - UIComponent with which this model object value is associated
    value - String value to be converted (may be null)

Returns:
    null if the value to convert is null, otherwise the result of the conversion

Throws:
    ConverterException - if conversion cannot be successfully performed
    NullPointerException - if context or component is null

public String getAsString(FacesContext context, 
    UIComponent component, Object value)

Throws ConverterException: NullPointeException

Throws NullPointeException: null

getAsString

public String getAsString(FacesContext context, 
    UIComponent component, Object value)
Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by:
getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context Or component is null

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the
state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by: saveState in interface StateHolder

public void restoreState(FacesContext context, Object state)

restoreState

public void restoreState(FacesContext context,
Object state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: restoreState in interface StateHolder
public boolean isTransient()

isTransient

public boolean isTransient()

Description copied from interface: StateHolder

If true, the Object implementing this interface must not participate in state saving or restoring.

Specified by:
    isTransient in interface StateHolder

public void setTransient(boolean transientFlag)

setTransient

public void setTransient(boolean transientFlag)

Description copied from interface: StateHolder

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

Specified by:
    setTransient in interface StateHolder

Parameters:
    transientFlag - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.holders  Class ObjectHolder

java.lang.Object
   ↓  javax.xml.rpc.holders.ObjectHolder

All Implemented Interfaces:
   Holder

public final class ObjectHolder
extends Object
implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>Object</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ObjectHolder()</td>
</tr>
<tr>
<td></td>
<td>ObjectHolder(Object value)</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
### Field Detail

**value**

```java
public Object value
```

### Constructor Detail

**public ObjectHolder()**

**ObjectHolder**

```java
public ObjectHolder()
```

**public ObjectHolder(Object value)**

**ObjectHolder**

```java
public ObjectHolder(Object value)
```
PS:
javax.jms Interface ObjectMessage

All Superinterfaces: Message

public interface ObjectMessage

extends Message

Implements: Message

ObjectMessage implements Java "Java" Message
Serializable Java

Java JDK 1.2 Collection

ObjectMessage Message Not Writeable Exception clearBody

version 1.0 - 6 August 1998

See also en
createObjectMessage(), createObjectMessage(Serializable),

An ObjectMessage object is used to send a message that contains a serializable object in the Java programming language ("Java object"). It inherits from the Message interface and adds a body containing a single reference to an object. Only Serializable Java objects can be used.

If a collection of Java objects must be sent, one of the Collection classes provided since JDK 1.2 can be used.

When a client receives an ObjectMessage, it is in read-only mode. If a client attempts to write to the message at this point, a
MessageNotWriteableException is thrown. If clearBody is called, the message can now be both read from and written to.

Version:
1.0 - 6 August 1998

Author:
Mark Hapner, Rich Burridge

See Also:
Session.createObjectMessage(),
Session.createObjectMessage(Serializable), BytesMessage,
MapMessage, Message, StreamMessage, TextMessage

---

Field Summary

Field Summary

Fields inherited from interface javax.jms.Message

DEFAULT_DELIVERY_MODE, DEFAULT_PRIORITY, DEFAULT_TIME_TO_LIVE

Method Summary

getObject() Gets the serializable object containing this message's data.

setObject(Serializable object) Sets the serializable object containing this message's data.

Methods inherited from interface javax.jms.Message

acknowledge, clearBody, clearProperties, getBooleanProperty, getByteProperty, getDoubleProperty, setFloatProperty, getIntProperty, getJMSCorrelationID, getJMSCorrelationIDAsBytes, getJMSDeliveryMode, getJMSDestination, getJMSExpiration, getJMSMessageID, getJMSPriority, getJMSRedelivered, getJMSReplyTo, getJMSTimestamp, getJMSType, getLongProperty, setObjectProperty, getPropertyNames, getShortProperty, getStringProperty, propertyExists, setBooleanProperty, setByteProperty, setDoubleProperty, setFloatProperty, setIntProperty, setJMSCorrelationID, setJMSCorrelationIDAsBytes, setJMSDeliveryMode, setJMSDestination, setJMSExpiration,
public void setObject(java.io.Serializable object) throws JMSException

    setObject()

object

Throws JMSException: JMS
Throws MessageFormatException: 
Throws MessageNotWriteableException:

setObject

void setObject(Serializable object)
    throws JMSException

Sets the serializable object containing this message's data. It is important to note that an objectMessage contains a snapshot of the object at the time setObject() is called; subsequent modifications of the object will have no effect on the ObjectMessage body.

Parameters:
    object - the message's data

Throws:
    JMSException - if the JMS provider fails to set the object due to some internal error.
    MessageFormatException - if object serialization fails.
    MessageNotWriteableException - if the message is in read-only mode.
public java.io.Serializable getObject() throws JMSException
null

return

Throws JMSException: JMS
Throws MessageFormatException:

getObject

Serializable getObject()
throws JMSException

Gets the serializable object containing this message's data. The default value is null.

Returns: the serializable object containing this message's data

Throws: JMSException - if the JMS provider fails to get the object due to some internal error.
MessageFormatException - if object deserialization fails.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  

Class ObjectNotFoundException

java.lang.Object
  ↓ java.lang.Throwable
      ↓ java.lang.Exception
          ↓ javax.ejb.FinderException
              ↓ javax.ejb.ObjectNotFoundException

All Implemented Interfaces:
  Serializable

public class ObjectNotFoundException

extends FinderException

Extends: Throwable > Exception > FinderException

ObjectNotFoundException finder EJB

finder EJB finder EJB

The ObjectNotFoundException exception is thrown by a finder method to indicate that the specified EJB object does not exist.

Only the finder methods that are declared to return a single EJB object use this exception. This exception should not be thrown by finder methods that return a collection of EJB objects (they should return an empty collection instead).

See Also:
  Serialized Form

---

Constructor Summary

ObjectNotFoundException()
Constructs an `ObjectNotFoundException` with no detail message.

```java
ObjectNotFoundException(String message)
```

Constructs an `ObjectNotFoundException` with the specified detail message.

## Method Summary

**Methods inherited from class java.lang.Throwable**

- `fillInStackTrace`
- `getCause`
- `getLocalizedMessage`
- `getMessage`
- `getStackTrace`
- `initCause`
- `printStackTrace`
- `printStackTrace`
- `printStackTrace`
- `setStackTrace`
- `toString`

**Methods inherited from class java.lang.Object**

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `wait`
- `wait`
- `wait`

## Constructor Detail

**public ObjectNotFoundException()**

`ObjectNotFoundException`

**public ObjectNotFoundException(String message)**

`ObjectNotFoundException`

```java

Constructs an ObjectNotFoundException with no detail message.
```
public ObjectNotFoundException(String message)

Constructs an ObjectNotFoundException with the specified detail message.
javax.persistence  **Annotation Type OneToMany**

```java
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface OneToMany

**Implements:** Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Collection  generic
1  generic
Customer
   @OneToMany(cascade=ALL, mappedBy="customer")
   public Set getOrders() { return orders; }
Order
   @ManyToOne
   @JoinColumn(name="CUST_ID", nullable=false)
   public Customer getCustomer() { return customer; }
2  generic
Customer
   @OneToMany(targetEntity=com.acme.Order.class, cascade=ALL,
     mappedBy="customer")
   public Set getOrders() { return orders; }
Order
   @ManyToOne
   @JoinColumn(name="CUST_ID", nullable=false)
   public Customer getCustomer() { return customer; }
```

since  Java Persistence 1.0  en
Defines a many-valued association with one-to-many multiplicity.

If the collection is defined using generics to specify the element type, the associated target entity type need not be specified; otherwise the target entity class must be specified.

Example 1: One-to-Many association using generics

In Customer class:

```java
@OneToMany(cascade=ALL, mappedBy="customer")
public Set getOrders() { return orders; }
```

In Order class:

```java
@ManyToOne
@JoinColumn(name="CUST_ID", nullable=false)
public Customer getCustomer() { return customer; }
```

Example 2: One-to-Many association without using generics

In Customer class:

```java
@OneToMany(targetEntity=com.acme.Order.class, cascade=ALL,
    mappedBy="customer")
public Set getOrders() { return orders; }
```

In Order class:

```java
@ManyToOne
@JoinColumn(name="CUST_ID", nullable=false)
public Customer getCustomer() { return customer; }
```

Since:
Java Persistence 1.0

---

**Optional Element Summary**

<table>
<thead>
<tr>
<th>CascadeType[]</th>
<th>cascade</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) The operations that must be cascaded to the target of the association.</td>
<td></td>
</tr>
<tr>
<td>FetchType</td>
<td>fetch</td>
</tr>
<tr>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>String</td>
<td>mappedBy</td>
</tr>
<tr>
<td>Class</td>
<td>targetEntity</td>
</tr>
</tbody>
</table>

abstract public Class<T> targetEntity()
Java generic Collection
generic Collection
targetEntity

public abstract Class targetEntity

(Optional) The entity class that is the target of the association. Optional only if the collection property is defined using Java generics. Must be specified otherwise.

Defaults to the parameterized type of the collection when defined using generics.

Default:
void.class

abstract public CascadeType[] cascade()
**cascade**

public abstract CascadeType[] cascade

(Optional) The operations that must be cascaded to the target of the association.

Defaults to no operations being cascaded.

Default:

{}  

---

**abstract public FetchType fetch()**

public abstract FetchType fetch

EAGER     LAZY

**fetch**

public abstract FetchType fetch

(Optional) Whether the association should be lazily loaded or must be eagerly fetched. The EAGER strategy is a requirement on the persistence provider runtime that the associated entities must be eagerly fetched. The LAZY strategy is a hint to the persistence provider runtime.

Default:

LAZY  

---

**abstract public String mappedBy()**

**mappedBy**
public abstract String mappedBy

The field that owns the relationship. Required unless the relationship is unidirectional.

Default: ""
javax.persistence Annotation Type OneToOne

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface OneToOne

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

1

Customer

    @OneToOne(optional=false)
    @JoinColumn(
        name="CUSTREC_ID", unique=true, nullable=false, updatable=false
    )
    public CustomerRecord getCustomerRecord() { return customerRecord; }

CustomerRecord

    @OneToOne(optional=false, mappedBy="customerRecord")
    public Customer getCustomer() { return customer; }

2

Employee

    @Entity
    public class Employee {
        @Id Integer id;

        @OneToOne @PrimaryKeyJoinColumn
        EmployeeInfo info;
        ...
    }

EmployeeInfo

    @Entity
    public class EmployeeInfo {
        @Id Integer id;
        ...
    }
This annotation defines a single-valued association to another entity that has one-to-one multiplicity. It is not normally necessary to specify the associated target entity explicitly since it can usually be inferred from the type of the object being referenced.

Example 1: One-to-one association that maps a foreign key column

On Customer class:

```java
@OneToOne(optional=false)
@JoinColumn(
    name="CUSTREC_ID", unique=true, nullable=false, updatable=false
)
public CustomerRecord getCustomerRecord() { return customerRecord; }
```

On CustomerRecord class:

```java
@OneToOne(optional=false, mappedBy="customerRecord")
public Customer getCustomer() { return customer; }
```

Example 2: One-to-one association that assumes both the source a

On Employee class:

```java
@Entity
public class Employee {
    @Id Integer id;

    @OneToOne @PrimaryKeyJoinColumn
    EmployeeInfo info;

    ...
}
```

On EmployeeInfo class:

```java
@Entity
public class EmployeeInfo {
    @Id Integer id;

    ...
}
```
Since:
Java Persistence 1.0

Optional Element Summary

<table>
<thead>
<tr>
<th>Optional Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CascadeType[]</td>
<td>Optional) The operations that must be cascaded to the target of the association.</td>
</tr>
<tr>
<td>fetch</td>
<td>(Optional) Whether the association should be lazily loaded or must be eagerly fetched.</td>
</tr>
<tr>
<td>mappedBy</td>
<td>(Optional) The field that owns the relationship.</td>
</tr>
<tr>
<td>optional</td>
<td>(Optional) Whether the association is optional.</td>
</tr>
<tr>
<td>Class targetEntity</td>
<td>(Optional) The entity class that is the target of the association.</td>
</tr>
</tbody>
</table>

abstract public Class<T> targetEntity()

targetEntity

public abstract Class targetEntity

(Optional) The entity class that is the target of the association.

Defaults to the type of the field or property that stores the association.
abstract public `CascadeType[] cascade()`

`cascade`

```java
public abstract `CascadeType[] cascade`
```

(Optional) The operations that must be cascaded to the target of the association.

By default no operations are cascaded.

Default:
```
{}  
```

abstract public ` FetchType fetch()`

`fetch`

```java
public abstract ` FetchType fetch`
```

(Optional) Whether the association should be lazily loaded or must be eagerly fetched. The `EAGER` strategy is a requirement on the persistence provider runtime that the associated entity must be eagerly fetched. The `LAZY` strategy is a hint to the persistence provider runtime.
abstract public boolean optional()  
false null

optional

public abstract boolean optional

(Optional) Whether the association is optional. If set to false then a non-null relationship must always exist.

Default:
true

abstract public String mappedBy()

mappedBy

public abstract String mappedBy

(Optional) The field that owns the relationship. This element is only specified on the inverse (non-owning) side of the association.

Default:

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jws Annotation Type Oneway

@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface Oneway

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=METHOD)

@WebMethod @Oneway Holder 181

Indicates that the given @WebMethod has only an input message and no output. Typically, a oneway method returns the thread of control to the calling application prior to executing the actual business method. A 181 processor should report an error if an operation marked @Oneway has a return value or Holder parameters, or declares any checked exceptions.

Author:
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.
javax.enterprise.deploy.spi.exceptions Class
OperationUnsupportedException

does not inherit from java.lang.Object

All Implemented Interfaces:
Serializable

public class OperationUnsupportedException
extends Exception

Extends: Throwable > Exception

This exception is to report that the method called is not supported by this implementation.

See Also:
Serialized Form

Constructor Summary

| OperationUnsupportedException(String s) |
| Creates an new OperationUnsupportedException object. |

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage,
getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public OperationUnsupportedException(String s)

OperationUnsupportedException

public OperationUnsupportedException(String s)

Creates an new OperationUnsupportedException object.

Parameters:

s - a string indicating what was wrong with the target.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Class OptimisticLockException

java.lang.Object  
  ▼ java.lang.Throwable  
    ▼ java.lang.Exception  
      ▼ java.lang.RuntimeException  
        ▼ javax.persistence.PersistenceException  
          ▼ javax.persistence.OptimisticLockException

All Implemented Interfaces:  
  Serializable

public class OptimisticLockException  
  extends PersistenceException

Extends: Throwable > Exception > RuntimeException > PersistenceException

API

since Java Persistence 1.0  

Thrown by the persistence provider when an optimistic locking conflict occurs. This exception may be thrown as part of an API call, a flush or at commit time. The current transaction, if one is active, will be marked for rollback.

Since:  
      Java Persistence 1.0
See Also:  
      Serialized Form

Constructor Summary

OptimisticLockException()  
  Constructs a new OptimisticLockException exception with null
as its detail message.

**OptimisticLockException**(Object entity)  
Constructs a new OptimisticLockException exception with the specified entity.

**OptimisticLockException**(String message)  
Constructs a new OptimisticLockException exception with the specified detail message.

**OptimisticLockException**(String message, Throwable cause)  
Constructs a new OptimisticLockException exception with the specified detail message and cause.

**OptimisticLockException**(String message, Throwable cause, Object entity)  
Constructs a new OptimisticLockException exception with the specified detail message, cause, and entity.

**OptimisticLockException**(Throwable cause)  
Constructs a new OptimisticLockException exception with the specified cause.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object getEntity()</strong></td>
<td>Returns the entity that caused this exception.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

- fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Constructor Detail
public OptimisticLockException()
    null OptimisticLockException

OptimisticLockException

public OptimisticLockException()

    Constructs a new OptimisticLockException exception with null as its detail message.

public OptimisticLockException(String message)

    Constructs a new OptimisticLockException exception with the specified detail message.

    Parameters:
    message - the detail message.

public OptimisticLockException(String message, Throwable cause)

    OptimisticLockException
    message
    cause

OptimisticLockException
public OptimisticLockException(String message, Throwable cause)

Constructs a new OptimisticLockException exception with the specified detail message and cause.

Parameters:
  - message - the detail message.
  - cause - the cause.

public OptimisticLockException(Throwable cause)

OptimisticLockException cause

OptimisticLockException

public OptimisticLockException(Throwable cause)

Constructs a new OptimisticLockException exception with the specified cause.

Parameters:
  - cause - the cause.

public OptimisticLockException(Object entity)

OptimisticLockException entity

OptimisticLockException

public OptimisticLockException(Object entity)

Constructs a new OptimisticLockException exception with the specified entity.
public OptimisticLockException(String message, Throwable cause, Object entity)

Constructs a new OptimisticLockException exception with the specified detail message, cause, and entity.

Parameters:
message - the detail message.
cause - the cause.
entity - the entity.

Method Detail

public Object getEntity()

return

gEntity

public Object getEntity()
Returns the entity that caused this exception.

**Returns:**
the entity.
javax.persistence  Annotation Type OrderBy

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface OrderBy

**Implements:** Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

Collection

value orderby_list

orderby_list::= orderby_item [,orderby_item] *
orderby_item::= property_or_field_name [ASC | DESC]

    ASC    DESC    ASC

@Entity public class Course {
    ...
    @ManyToMany
    @OrderBy("lastname ASC")
    public List getStudents() {...};
    ...
}

@Entity public class Student {
    ...
    @ManyToMany(mappedBy="students")
    @OrderBy // PK is assumed
    public List getCourses() {...};
    ...
}
This annotation specifies the ordering of the elements of a collection valued association at the point when the association is retrieved.

The syntax of the value ordering element is an orderby_list, as follows:

```
orderby_list ::= orderby_item [,orderby_item] *
orderby_item ::= property_or_field_name [ASC | DESC]
```

If ASC or DESC is not specified, ASC (ascending order) is assumed.

If the ordering element is not specified, ordering by the primary key of the associated entity is assumed.

The property or field name must correspond to that of a persistent property or field of the associated class. The properties or fields used in the ordering must correspond to columns for which comparison operators are supported.

Example:

```java
@Entity
public class Course {
  ...
  @ManyToMany
  @OrderBy("lastname ASC")
  public List getStudents() {...};
  ...
}

@Entity
public class Student {
  ...
  @ManyToMany(mappedBy="students")
  @OrderBy // PK is assumed
  public List getCourses() {...};
  ...
}
```

**Since:**
Java Persistence 1.0
An orderby_list, specified as follows:

```
orderby_list ::= orderby_item [,orderby_item] *
orderby_item ::= property_or_field_name [ASC | DESC]
```

If ASC or DESC is not specified, ASC (ascending order) is assumed.

If the ordering element is not specified, ordering by the primary key of the associated entity is assumed.

**Default:**

"""
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.registry.infomodel Interface Organization

All Superinterfaces:
   ExtensibleObject, RegistryObject

public interface Organization
extends RegistryObject

Implements: RegistryObject

Organization  Submitting Organization Organization
Organization Organization

See also  javax.xml.registry.infomodel.Service

Organization instances provide information on organizations such as a Submitting Organization. Each Organization instance may have a reference to a parent Organization. In addition it may have a contact attribute defining the primary contact within the organization. An Organization also has an address attribute.

Author:  Farrukh S. Najmi
See Also:  Service

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addChildOrganization(Organization organization)</td>
</tr>
<tr>
<td>Adds a child Organization.</td>
</tr>
<tr>
<td>void addChildOrganizations(Collection organizations)</td>
</tr>
<tr>
<td>Adds a Collection of Organization children.</td>
</tr>
<tr>
<td>void addService(Service service)</td>
</tr>
<tr>
<td>Adds a child Service.</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><code>void addServices(Collection services)</code></td>
</tr>
<tr>
<td><code>void addUser(User user)</code></td>
</tr>
<tr>
<td><code>void addUsers(Collection users)</code></td>
</tr>
<tr>
<td><code>int getChildOrganizationCount()</code></td>
</tr>
<tr>
<td><code>Collection getChildOrganizations()</code></td>
</tr>
<tr>
<td><code>Collection getDescendantOrganizations()</code></td>
</tr>
<tr>
<td><code>Organization getParentOrganization()</code></td>
</tr>
<tr>
<td><code>PostalAddress getPostalAddress()</code></td>
</tr>
<tr>
<td><code>User getPrimaryContact()</code></td>
</tr>
<tr>
<td><code>Organization getRootOrganization()</code></td>
</tr>
<tr>
<td><code>Collection getServices()</code></td>
</tr>
<tr>
<td><code>Collection getTelephoneNumbers(String phoneType)</code></td>
</tr>
<tr>
<td><code>Collection getUsers()</code></td>
</tr>
<tr>
<td><code>void removeChildOrganization(Organization organization)</code></td>
</tr>
<tr>
<td><code>void removeChildOrganizations(Collection organizations)</code></td>
</tr>
<tr>
<td><code>void removeService(Service service)</code></td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>---------------------------------------------</td>
</tr>
<tr>
<td><code>void removeServices(Collection services)</code></td>
</tr>
<tr>
<td><code>void removeUser(User user)</code></td>
</tr>
<tr>
<td><code>void removeUsers(Collection users)</code></td>
</tr>
<tr>
<td><code>void setPostalAddress(PostalAddress address)</code></td>
</tr>
<tr>
<td><code>void setPrimaryContact(User primaryContact)</code></td>
</tr>
<tr>
<td><code>void setTelephoneNumbers(Collection phoneNumbers)</code></td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`

- `addAssociation`, `addAssociations`, `addClassification`, `addClassifications`, `addExternalIdentifier`, `addExternalIdentifiers`, `addExternalLink`, `addExternalLinks`, `getAssociatedObjects`, `getAssociations`, `getAuditTrail`, `getClassifications`, `getDescription`, `getExternalIdentifiers`, `getExternalLinks`, `getKey`, `getLifeCycleManager`, `getName`, `getObjectType`, `getRegistryPackages`, `getSubmittingOrganization`, `removeAssociation`, `removeAssociations`, `removeClassification`, `removeClassifications`, `removeExternalIdentifier`, `removeExternalIdentifiers`, `removeExternalLink`, `removeExternalLinks`, `setAssociations`, `setClassifications`, `setDescription`, `setExternalIdentifiers`, `setExternalLinks`, `setKey`, `setName`, `toXML`

Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`

- `addSlot`, `addSlots`, `getSlot`, `getSlots`, `removeSlot`, `removeSlots`

**Method Detail**

```java
public PostalAddress getPostalAddress() throws
```
JAXRException
Organization

1

return Organization PostalAddress
Throws JAXRException: JAXR
supplierCardinality directed
link associates
aggregationByValue

getPostalAddress

PostalAddress getPostalAddress()
throws JAXRException

Gets the Address for this Organization.

Capability Level: 1

Returns:
the PostalAddress for this Organization

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setPostalAddress(PostalAddress address)
throws JAXRException
Organization

address Organization PostalAddress
Throws JAXRException: JAXR
**setPostalAddress**

```java
void setPostalAddress(PostalAddress address)
                throws JAXRException
```

Sets the address for this Organization.

**Capability Level: 1**

**Parameters:**
- `address` - the `PostalAddress` for this Organization

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public User getPrimaryContact() throws JAXRException**

```java
Organization Organization User
Organization getUsers
```

```java
return Organization Organization User null
Throws JAXRException: JAXR
```

**supplierCardinality**
- `directed` 1

**link**
- `aggregationByValue primaryContact` en
- `associates` en

**getPrimaryContact**

```java
User getPrimaryContact()
                throws JAXRException
```
Gets the primary Contact for this Organization. The primary contact is one of the Users of the Organization, as returned by the getUsers call on an Organization instance.

**Capability Level: 0**

**Returns:**
the User that is the primary Contact for this Organization. Must not be null

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

---

```java
public void setPrimaryContact(User primaryContact) throws JAXRException
Organization User Collection Collection
User Collection User Collection User
Collection User

0

primaryContact Organization User

Throws JAXRException: JAXR

setPrimaryContact

void setPrimaryContact(User primaryContact) throws JAXRException
```

Sets the primary contact for this Organization. If the primary contact is not in the existing Collection of User instances for this object then it must be added to that Collection. If the primary contact is already in the existing Collection of User instances for this object, then it should not be added to the Users collection as a duplicate entry. In either case, the primary contact is a distinguished User instance within the Collection of User instances for this Object.
Capability Level: 0

Parameters:
   primaryContact - the User that is the primary Contact for this Organization

Throws:
   JAXRException - If the JAXR provider encounters an internal error

```java
public void addUser(User user) throws JAXRException

User

user

Throws
   JAXRException: JAXR
```

**addUser**

```java
void addUser(User user)
   throws JAXRException

Adds a User.

Capability Level: 0

Parameters:
   user - the User being added to this object

Throws:
   JAXRException - If the JAXR provider encounters an internal error
```

```java
public void addUsers(java.util.Collection<E> users) throws JAXRException

User  Collection
```
0

users  User Collection
Throws     JAXRException: JAXR

addUsers

void addUsers(Collection users) throws JAXRException

Adds a Collection of Users.

Capability Level: 0

Parameters:
users - the Collection of Users being added to this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

0

User

Throws     JAXRException: JAXR

removeUser

void removeUser(User user) throws JAXRException

Removes a User.
public void removeUsers(java.util.Collection<E> users) throws JAXRException

Remove a Collection of Users.

Capability Level: 0

Parameters:
users - the Collection of Users being removed from this object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getUsers() throws JAXRException

Organization User Collection
getUsers

```java
Collection getUsers() throws JAXRException
```

Gets the Collection of Users affiliated with this Organization. One of these users is designated as the primary contact.

**Capability Level: 0**

**Returns:**
Collection of User instances. The Collection may be empty but not null.

**Throws:**

JAXRException - If the JAXR provider encounters an internal error

**See Also:**
User

public java.util.Collection<E> getTelephoneNumbers(String phoneType) throws JAXRException
```java
User phoneType null telephoneNumber
```
getTelephoneNumbers

```java
Collection getTelephoneNumbers(String phoneType)
throws JAXRException
```

Gets the telephone numbers for this User that match the specified telephone number type. If phoneType is null return all telephoneNumbers.

**Capability Level:** 0

**Parameters:**
- `phoneType` - a String that specifies the type of phone numbers desired

**Returns:**
- Collection of TelephoneNumber instances. The Collection may be empty but not null.

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**See Also:**
- `TelephoneNumber`

```
public void setTelephoneNumbers(java.util.Collection<E> phoneNumbers) throws JAXRException
```
**setTelephoneNumbers**

```java
void setTelephoneNumbers(Collection phoneNumbers)
throws JAXRException
```

Sets the various telephone numbers for this user.

**Capability Level:** 0

**Parameters:**
- `phoneNumbers` - the TelephoneNumbers being set for this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**public void addService(Service service) throws JAXRException**

```java
service
Throws JAXRException: JAXR
```

**addService**

```java
void addService(Service service)
throws JAXRException
```
Adds a child Service.

**Capability Level:** 0

**Parameters:**
- `service` - the Service being added to this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

```java
public void addServices(java.util.Collection<E> services)
    throws JAXRException
```

**Service Collection**

**0**

- `services` - Service Collection
- **Throws**
  - `JAXRException`: JAXR

---

**addServices**

```java
void addServices(Collection services)
    throws JAXRException
```

Adds a Collection of Service children.

**Capability Level:** 0

**Parameters:**
- `services` - the Collection of Services being added to this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

```java
public void removeService(Service service) throws
```
**JAXRException**

**Service**

0

```java
service Service
```

Throws **JAXRException**: JAXR

**removeService**

```java
void removeService(Service service)
```

Removes a Service from this object.

Capability Level: 0

**Parameters:**

- service: the Service being removed from this object

**Throws:**

- **JAXRException**: If the JAXR provider encounters an internal error

```java
public void removeServices(java.util.Collection<E> services) throws JAXRException
```

Service Collection

0

```java
services Service Collection
```

Throws **JAXRException**: JAXR

**removeServices**
void removeServices(Collection services)
   throws JAXRException

   Removes a Collection of children Services from this object.

   **Capability Level:** 0

   **Parameters:**
   services - the Collection of Services being removed from this object

   **Throws:**
   JAXRException - If the JAXR provider encounters an internal error

---

public java.util.Collection<E> getServices() throws JAXRException

Service

0

getServices

Collection getServices() throws JAXRException

Gets all children Services.

**Capability Level:** 0
Returns:
Collection of Service instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
Service

public void addChildOrganization(Organization organization) throws JAXRException

Adds a child Organization.

Capability Level: 1

Parameters:
organization - the child Organization being added

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void addChildOrganizations(java.util.Collection<E> organizations) throws JAXRException
Organization Collection

1

organizations Organization Collection
Throws JAXRException: JAXR

addChildOrganizations

void addChildOrganizations(Collection organizations)

throws JAXRException

Adds a Collection of Organization children.

Capability Level: 1

Parameters:
organizations - the Collection of child Organizations being added

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void removeChildOrganization(Organization organization) throws JAXRException

Organization

1

organization Organization
Throws JAXRException: JAXR

removeChildOrganization

void removeChildOrganization(Organization organization)
Removes a child Organization.

**Capability Level: 1**

**Parameters:**
- `organization` - the child Organization being removed

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

```java
public void removeChildOrganizations(java.util.Collection<E> organizations) throws JAXRException
Organization Collection

1

organizations Organization Collection
Throws JAXRException: JAXR

removeChildOrganizations

void removeChildOrganizations(Collection organizations) throws JAXRException

Removes a Collection of children Organizations.

**Capability Level: 1**

**Parameters:**
- `organizations` - the Collection of child Organizations being removed

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error
public int getChildOrganizationCount() throws JAXRException

1

    return Organization
    Throws JAXRException: JAXR

getChildOrganizationCount

int getChildOrganizationCount() throws JAXRException

    Gets number of children.

    Capability Level: 1

    Returns:
    the number of children Organizations

    Throws:
    JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getChildOrganizations() throws JAXRException

Organization

1

    return Organization CollectionCollection null
    Throws JAXRException: JAXR
    See also javax.xml.registry.infomodel.Organization
**getChildOrganizations**

```java
Collection getChildOrganizations() throws JAXRException
```

Gets all immediate children Organizations.

**Capability Level: 1**

**Returns:**
Collection of Organization instances. The Collection may be empty but not null.

**Throws:**
 JAXRException - If the JAXR provider encounters an internal error

**See Also:**
Organization

---

**public java.util.Collection<E> getDescendantOrganizations() throws JAXRException**

**Organization**

1

```java
return Organization CollectionCollection null
```

**Throws**
 JAXRException: JAXR

**See also**
javax.xml.registry.infomodel.Organization

---

**getDescendantOrganizations**

```java
Collection getDescendantOrganizations() throws JAXRException
```

Gets all descendant Organizations.

**Capability Level: 1**
Returns:
Collection of Organization instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
Organization

public Organization getParentOrganization() throws JAXRException

Organization

1
return Organization Organization null
Throws JAXRException: JAXR

getParentOrganization

Organization getParentOrganization() throws JAXRException

Gets the parent (container) organization.

Capability Level: 1

Returns:
the parent Organization or null if object has no parent Organization

Throws:
JAXRException - If the JAXR provider encounters an internal error

public Organization getRootOrganization() throws
JAXRException
Organization

1

    return Organization Organization null

Throws    JAXRException:  JAXR

getRootOrganization

Organization getRootOrganization() throws JAXRException

Gets the root organization.

Capability Level: 1

Returns: the root Organization or null if object has no parent Organization

Throws:    JAXRException - If the JAXR provider encounters an internal error

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.search  Class OrTerm

java.lang.Object
   javax.mail.search.SearchTerm
   javax.mail.search.OrTerm

All Implemented Interfaces:
   Serializable

public final class OrTerm
extends SearchTerm

Extends: SearchTerm

SearchTerm OR

This class implements the logical OR operator on individual SearchTerms.

Author:  Bill Shannon, John Mani

See Also:  Serialized Form

Field Summary

| protected searchTerm[] terms | The array of terms on which the OR operator should be applied. |

Constructor Summary

OrTerm(SearchTerm[] t)  Constructor that takes an array of SearchTerms.
**OrTerm**(SearchTerm t1, SearchTerm t2)

Constructor that takes two operands.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>SearchTerm[]</td>
<td>getTerms()</td>
<td>Return the search terms.</td>
</tr>
<tr>
<td>int</td>
<td>hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean</td>
<td>match(Message msg)</td>
<td>The OR operation.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, toString, wait, wait

### Field Detail

```java
protected SearchTerm[] terms
```

The array of terms on which the OR operator should be applied.

### Constructor Detail

```java
public OrTerm(SearchTerm t1, SearchTerm t2)
```
OrTerm

public OrTerm(SearchTerm t1, SearchTerm t2)

Constructor that takes two operands.

Parameters:
  t1 - first term
  t2 - second term

public OrTerm(SearchTerm[] t)

SearchTerm

  t

OrTerm

public OrTerm(SearchTerm[] t)

Constructor that takes an array of SearchTerms.

Parameters:
  t - array of search terms

Method Detail

public SearchTerm[] getTerms()
Return the search terms.

```java
public boolean match(Message msg)
```

The OR operation.

The terms specified in the constructor are applied to the given object and the OR operator is applied to their results.

**Specified by:**
```
match in class searchTerm
```

**Parameters:**

- `msg`: The specified SearchTerms are applied to this Message and the OR operator is applied to their results.

**Returns:**

- true if the OR succeeds, otherwise false

```java
public boolean equals(Object obj)
```

```java
public boolean equals(Object obj)
```
Equality comparison.

Overrides:
  equals in class Object

public int hashCode()
hashCode

hashCode

public int hashCode()
  Compute a hashCode for this object.

Overrides:
  hashCode in class Object
javax.servlet.jsp  Class PageContext

java.lang.Object
  ▼ javax.servlet.jsp.JspContext
    ▼ javax.servlet.jsp.PageContext

public abstract class PageContext
extends JspContext

Extends: JspContext

PageContext  JspContext  Servlet  JSP
PageContext  JSP  pageContext
PageContext  JSP  PageContext  JSP
JspFactory.getPageContext()  JspFactory.releasePageContext()

JSP Page Implementation  PageContextJspFactory

PageContext /

  •  API
  •  API
  •  JspWriter
  •
  •
  •  errorpage

JSP  JSP

  initialize()  release()
JspWriter  pushBody()

JSP

getException() getPage() getRequest() getResponse() getSession() getServletContext()

forward() include() handlePageException()

PageContext extends JspContext to provide useful context information for when JSP technology is used in a Servlet environment.

A PageContext instance provides access to all the namespaces associated with a JSP page, provides access to several page attributes, as well as a layer above the implementation details. Implicit objects are added to the pageContext automatically.

The PageContext class is an abstract class, designed to be extended to provide implementation dependent implementations thereof, by conformant JSP engine runtime environments. A PageContext instance is obtained by a JSP implementation class by calling the JspFactory.getPageContext() method, and is released by calling JspFactory.releasePageContext().

An example of how PageContext, JspFactory, and other classes can be used within a JSP Page Implementation object is given elsewhere.

The PageContext provides a number of facilities to the page/component author and page implementor, including:

- a single API to manage the various scoped namespaces
- a number of convenience API's to access various public objects
- a mechanism to obtain the JspWriter for output
- a mechanism to manage session usage by the page
- a mechanism to expose page directive attributes to the scripting environment
- mechanisms to forward or include the current request to other active
components in the application

- a mechanism to handle errorpage exception processing

**Methods Intended for Container Generated Code**

Some methods are intended to be used by the code generated by the container, not by code written by JSP page authors, or JSP tag library authors.

The methods supporting **lifecycle** are `initialize()` and `release()`

The following methods enable the **management of nested** JspWriter streams to implement Tag Extensions: `pushBody()`

**Methods Intended for JSP authors**

The following methods provide **convenient access** to implicit objects: `getException()`, `getPage()`, `getRequest()`, `getResponse()`, `getSession()`, `getServletConfig()` and `getServletContext()`.

The following methods provide support for **forwarding, inclusion and error handling**: `forward()`, `include()`, and `handlePageException()`.

---

**Field Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>String</td>
<td><strong>APPLICATION</strong> Name used to store ServletContext in PageContext name table.</td>
</tr>
<tr>
<td>static</td>
<td>int</td>
<td><strong>APPLICATION_SCOPE</strong> Application scope: named reference remains available in the ServletContext until it is reclaimed.</td>
</tr>
<tr>
<td>static</td>
<td>String</td>
<td><strong>CONFIG</strong> Name used to store ServletConfig in PageContext name table.</td>
</tr>
<tr>
<td>static</td>
<td>String</td>
<td><strong>EXCEPTION</strong> Name used to store uncaught exception in ServletRequest attribute list and PageContext name table.</td>
</tr>
<tr>
<td>static String OUT</td>
<td>Name used to store current JspWriter in PageContext name table.</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>static String PAGE</td>
<td>Name used to store the Servlet in this PageContext's nametables.</td>
<td></td>
</tr>
<tr>
<td>static int PAGE_SCOPE</td>
<td>Page scope: (this is the default) the named reference remains available in this PageContext until the return from the current Servlet.service() invocation.</td>
<td></td>
</tr>
<tr>
<td>static String PAGECONTEXT</td>
<td>Name used to store this PageContext in it's own name table.</td>
<td></td>
</tr>
<tr>
<td>static String REQUEST</td>
<td>Name used to store ServletRequest in PageContext name table.</td>
<td></td>
</tr>
<tr>
<td>static int REQUEST_SCOPE</td>
<td>Request scope: the named reference remains available from the ServletRequest associated with the Servlet until the current request is completed.</td>
<td></td>
</tr>
<tr>
<td>static String RESPONSE</td>
<td>Name used to store ServletResponse in PageContext name table.</td>
<td></td>
</tr>
<tr>
<td>static String SESSION</td>
<td>Name used to store HttpSession in PageContext name table.</td>
<td></td>
</tr>
<tr>
<td>static int SESSION_SCOPE</td>
<td>Session scope (only valid if this page participates in a session): the named reference remains available from the HttpSession (if any) associated with the Servlet until the HttpSession is invalidated.</td>
<td></td>
</tr>
</tbody>
</table>

**Constructor Summary**

```java
PageContext()  
Sole constructor.
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract void forward(String relativeUrlPath)</td>
<td>This method is used to re-direct, or &quot;forward&quot; the current ServletRequest and ServletResponse to another active component in the application.</td>
</tr>
<tr>
<td>ErrorData.getErrorData()</td>
<td>Provides convenient access to error information.</td>
</tr>
<tr>
<td>abstract Exception.getException()</td>
<td>The current value of the exception object (an Exception).</td>
</tr>
<tr>
<td>abstract Object.getPage()</td>
<td>The current value of the page object (In a Servlet environment, this is an instance of javax.servlet.Servlet).</td>
</tr>
<tr>
<td>abstract ServletRequest.getRequest()</td>
<td>The current value of the request object (a ServletRequest).</td>
</tr>
<tr>
<td>abstract ServletResponse.getResponse()</td>
<td>The current value of the response object (a ServletResponse).</td>
</tr>
<tr>
<td>abstract ServletConfig.getServletConfig()</td>
<td>The ServletConfig instance.</td>
</tr>
<tr>
<td>abstract ServletContext.getServletContext()</td>
<td>The ServletContext instance.</td>
</tr>
<tr>
<td>abstract HttpSession.getSession()</td>
<td>The current value of the session object (an HttpSession).</td>
</tr>
<tr>
<td>abstract void handlePageException(Exception e)</td>
<td>This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP.</td>
</tr>
<tr>
<td>abstract void handlePageException(Throwable t)</td>
<td>This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP.</td>
</tr>
</tbody>
</table>
## Methods

**abstract void include(String relativeUrlPath)**

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread.

**abstract void include(String relativeUrlPath, boolean flush)**

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread.

**abstract void initialize(Servlet servlet, ServletRequest request, ServletResponse response, String errorPageURL, boolean needsSession, int bufferSize, boolean autoFlush)**

The initialize method is called to initialize an uninitialized PageContext so that it may be used by a JSP Implementation class to service an incoming request and response within its `_jspService()` method.

**BodyContent pushBody()**

Return a new BodyContent object, save the current "out" JspWriter, and update the value of the "out" attribute in the page scope attribute namespace of the PageContext.

**abstract void release()**

This method shall "reset" the internal state of a PageContext, releasing all internal references, and preparing the PageContext for potential reuse by a later invocation of initialize().

### Methods inherited from class javax.servlet.jsp.JspContext

- `findAttribute`, `getAttribute`, `getAttributeNamesInScope`, `getAttributeScope`, `getELContext`, `getExpressionEvaluator`, `getOut`, `getVariableResolver`, `popBody`, `pushBody`, `removeAttribute`, `setAttribute`, `removeAttribute`, `setAttribute`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
Field Detail

PAGE_SCOPE

public static final int PAGE_SCOPE

Page scope: (this is the default) the named reference remains available in this PageContext until the return from the current Servlet.service() invocation.

See Also:
Constant Field Values

REQUEST_SCOPE

public static final int REQUEST_SCOPE

Request scope: the named reference remains available from the ServletRequest associated with the Servlet until the current request is completed.

See Also:
Constant Field Values

SESSION_SCOPE

public static final int SESSION_SCOPE

Session scope (only valid if this page participates in a session): the named reference remains available from the HttpSession (if any) associated with the Servlet until the HttpSession is invalidated.
APPLICATION_SCOPE

public static final int APPLICATION_SCOPE

Application scope: named reference remains available in the ServletContext until it is reclaimed.

See Also: Constant Field Values

PAGE

public static final String PAGE

Name used to store the Servlet in this PageContext's nametables.

See Also: Constant Field Values

PAGECONTEXT

public static final String PAGECONTEXT

Name used to store this PageContext in it's own name table.

See Also: Constant Field Values
REQUEST

public static final String REQUEST

Name used to store ServletRequest in PageContext name table.

See Also:  
Constant Field Values

RESPONSE

public static final String RESPONSE

Name used to store ServletResponse in PageContext name table.

See Also:  
Constant Field Values

CONFIG

public static final String CONFIG

Name used to store ServletConfig in PageContext name table.

See Also:  
Constant Field Values

SESSION
public static final String SESSION

Name used to store HttpSession in PageContext name table.

See Also:
Constant Field Values

OUT

public static final String OUT

Name used to store current JspWriter in PageContext name table.

See Also:
Constant Field Values

APPLICATION

public static final String APPLICATION

Name used to store ServletContext in PageContext name table.

See Also:
Constant Field Values

EXCEPTION

public static final String EXCEPTION

Name used to store uncaught exception in ServletRequest attribute list and PageContext name table.
See Also:
*Constant Field Values*

## Constructor Detail

```java
public PageContext()
```

**PageContext**

```java
public PageContext()
```

Sole constructor. (For invocation by subclass constructors, typically implicit.)

## Method Detail

```java
abstract public void initialize(Servlet servlet,
ServletRequest request, ServletResponse response, String errorPageURL,
boolean needsSession, int bufferSize,
boolean autoFlush) throws java.io.IOException,
IllegalStateException, IllegalArgumentException
```

initialize PageContext JSP _jspService()

JspFactory.getPageContext()

JspWriter "out"

```java

<table>
<thead>
<tr>
<th>servlet</th>
<th>PageContext</th>
<th>Servlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>request</td>
<td></td>
<td>Servlet</td>
</tr>
<tr>
<td>response</td>
<td></td>
<td>Servlet</td>
</tr>
<tr>
<td>errorPageURL</td>
<td>errorpage</td>
<td>null</td>
</tr>
</tbody>
</table>
```
initialize

public abstract void initialize(Servlet servlet,
ServletRequest request,
ServletResponse response,
String errorPageURL,
boolean needsSession,
int bufferSize,
boolean autoFlush)
throws IOException,
IllegalStateException,
IllegalArgumentException

The initialize method is called to initialize an uninitialized PageContext so that it may be used by a JSP Implementation class to service an incoming request and response within it's _jspService() method.

This method is typically called from JspFactory.getPageContext() in order to initialize state.

This method is required to create an initial JspWriter, and associate the "out" name in page scope with this newly created object.

This method should not be used by page or tag library authors.

Parameters:

&bullet; servlet - The Servlet that is associated with this PageContext
&bullet; request - The currently pending request for this Servlet
&bullet; response - The currently pending response for this Servlet
&bullet; errorPageURL - The value of the errorpage attribute from the page directive or null
needsSession - The value of the session attribute from the page
directive
bufferSize - The value of the buffer attribute from the page
directive
autoFlush - The value of the autoflush attribute from the page
directive

**Throws:**

- `IOException` - during creation of JspWriter
- `IllegalStateException` - if out not correctly initialized
- `IllegalArgumentException` - If one of the given parameters is invalid

---

**abstract public void release()**

```java
"""PageContext initialize() PageContext

JspFactory.releasePageContext()"
```

---

**release**

**public abstract void release()**

This method shall "reset" the internal state of a PageContext,
releasing all internal references, and preparing the PageContext for
potential reuse by a later invocation of `initialize()`. This method is
typically called from `JspFactory.releasePageContext()`.

Subclasses shall envelope this method.

This method should not be used by page or tag library authors.
abstract public HttpSession getSession()
HttpSession
    return PageContext HttpSession null

getSession

public abstract HttpSession getSession()

    The current value of the session object (an HttpSession).

    Returns:
        the HttpSession for this PageContext or null

abstract public Object getPage()
Servlet javax.servlet.Servlet
    return PageContext

ggetPage

public abstract Object getPage()

    The current value of the page object (In a Servlet environment, this is an instance of javax.servlet.Servlet).

    Returns:
        the Page implementation class instance associated with this PageContext

abstract public ServletRequest getRequest()
ServletRequest
    return PageContext ServletRequest
**getRequest**

```java
public abstract ServletRequest getRequest()
```

The current value of the request object (a ServletRequest).

**Returns:**
The ServletRequest for this PageContext

---

**getResponse**

```java
abstract public ServletResponse getResponse()
```

```java
ServletResponse return PageContext ServletResponse
```

---

**getException**

```java
abstract public Exception getException()
```

```java
Exception return errorpage PageContext
```

---

**getException**

```java
public abstract Exception getException()
```

The current value of the exception object (an Exception).

**Returns:**
any exception passed to this as an errorpage
abstract public ServletConfig getServletConfig()
ServletConfig return PageContext ServletConfig

getServletConfig

public abstract ServletConfig getServletConfig()

The ServletConfig instance.

Returns:
the ServletConfig for this PageContext

abstract public ServletContext getServletContext()
ServletContext return PageContext ServletContext

getServletContext

public abstract ServletContext getServletContext()

The ServletContext instance.

Returns:
the ServletContext for this PageContext

abstract public void forward(String relativeUrlPath) throws ServletException, java.io.IOException

HttpServletRequest  HttpServletResponse ""

relativeUrlPath  "/" URL JSP ServletException
forward

public abstract void forward(String relativeUrlPath)
throws ServletException, IOException

This method is used to re-direct, or "forward" the current ServletRequest and ServletResponse to another active component in the application.

If the relativeUrlPath begins with a "/" then the URL specified is calculated relative to the DOCROOT of the ServletContext for this JSP. If the path does not begin with a "/" then the URL specified is calculated relative to the URL of the request that was mapped to the calling JSP.

It is only valid to call this method from a Thread executing within a _jspService(...) method of a JSP.

Once this method has been called successfully, it is illegal for the calling Thread to attempt to modify the ServletResponse object. Any such attempt to do so, shall result in undefined behavior. Typically, callers immediately return from _jspService(...) after calling this method.

Parameters:
relativeUrlPath - specifies the relative URL path to the target resource as described above

Throws:
- `IllegalStateException` - if `ServletResponse` is not in a state where a forward can be performed
- `ServletException` - if the page that was forwarded to throws a `ServletException`
- `IOException` - if an I/O error occurred while forwarding

```java
abstract public void include(String relativeUrlPath) throws ServletException, java.io.IOException
```

Thread ServletRequest ServletResponse
ServletResponse

```java
include  JSP  JspWriter "out"
```

```java
relativeUrlPath  "/"  URL  JSP  ServletContext
DOCROOT  "/"  URL  JSP  URL
```

```java
JSP  _jspService(...)  Thread
```

```java
relativeUrlPath  URL
Throws  ServletException:  ServletException
Throws  java.io.IOException:  I/O
```

```java
include
```

```java
public abstract void include(String relativeUrlPath) throws ServletException,
IOException
```

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread. The output of the target resources processing of the request
is written directly to the ServletResponse output stream.

The current JspWriter "out" for this JSP is flushed as a side-effect of this call, prior to processing the include.

If the `relativeUrlPath` begins with a "/" then the URL specified is calculated relative to the DOCROOT of the ServletContext for this JSP. If the path does not begin with a "/" then the URL specified is calculated relative to the URL of the request that was mapped to the calling JSP.

It is only valid to call this method from a Thread executing within a `_jspService(...)` method of a JSP.

**Parameters:**
- `relativeUrlPath` - specifies the relative URL path to the target resource to be included

**Throws:**
- `ServletException` - if the page that was forwarded to throws a ServletException
- `IOException` - if an I/O error occurred while forwarding

```java
abstract public void include(String relativeUrlPath, boolean flush) throws ServletException, java.io.IOException
```

Thread ServletRequest ServletResponse
getOut() JspWriter

flush true include JSP JspWriter
"out" JspWriter "out"

```java
relativeUrlPath "/" URL JSP ServletContext
DOCROOT "/" URL JSP URL
```

JSP _jspService(...) Thread

```java
relativeUrlPath URL
```
**include**

```java
public abstract void include(String relativeUrlPath,
                              boolean flush)
    throws ServletException,
            IOException
```

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread. The output of the target resources processing of the request is written directly to the current JspWriter returned by a call to `getOut()`.

If flush is true, The current JspWriter "out" for this JSP is flushed as a side-effect of this call, prior to processing the include. Otherwise, the JspWriter "out" is not flushed.

If the `relativeUrlPath` begins with a "/" then the URL specified is calculated relative to the DOCROOT of the ServletContext for this JSP. If the path does not begin with a "/" then the URL specified is calculated relative to the URL of the request that was mapped to the calling JSP.

It is only valid to call this method from a Thread executing within a `_jspService(...)` method of a JSP.

**Parameters:**
- `relativeUrlPath` - specifies the relative URL path to the target resource to be included
- `flush` - True if the JspWriter is to be flushed before the include, or false if not.

**Throws:**
- `ServletException` - if the page that was forwarded to throws a ServletException
IOException - if an I/O error occurred while forwarding

Since:
    JSP 2.0

abstract public void handlePageException(Exception e)
throws ServletException, java.io.IOException

JSP """"

servlet

JSP  ServletResponse

PageContext.handlePageException(Throwable)

Throws ServletException:
Throws java.io.IOException: I/O
Throws NullPointerException: null
See also handlePageException(Throwable)

handlePageException

public abstract void handlePageException(Exception e)
throws ServletException, IOException

This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP. If forwarding is not possible (for example because the response has already been committed), an implementation dependent mechanism should be used to invoke the error page (e.g. "including" the error page instead).

If no error page is defined in the page, the exception should be rethrown so that the standard servlet error handling takes over.
A JSP implementation class shall typically clean up any local state prior to invoking this and will return immediately thereafter. It is illegal to generate any output to the client, or to modify any ServletResponse state after invoking this call.

This method is kept for backwards compatibility reasons. Newly generated code should use PageContext.handlePageException(Throwable).

**Parameters:**
- e - the exception to be handled

**Throws:**
- ServletException - if an error occurs while invoking the error page
- IOException - if an I/O error occurred while invoking the error page
- NullPointerException - if the exception is null

**See Also:**
handlePageException(Throwable)

```java
abstract public void handlePageException(Throwable t)
throws ServletException, java.io.IOException

JSP """

servlet

JSP ""

JSP  ServletResponse

<table>
<thead>
<tr>
<th>t</th>
<th>throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>ServletException:</td>
</tr>
<tr>
<td>Throws</td>
<td>java.io.IOException: I/O</td>
</tr>
<tr>
<td>Throws</td>
<td>NullPointerException: null</td>
</tr>
<tr>
<td>See also</td>
<td>handlePageException(Exception)</td>
</tr>
</tbody>
</table>
```
public abstract void handlePageException(Throwable t) throws ServletException, IOException

This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP. If forwarding is not possible (for example because the response has already been committed), an implementation dependent mechanism should be used to invoke the error page (e.g. "including" the error page instead).

If no error page is defined in the page, the exception should be rethrown so that the standard servlet error handling takes over.

This method is intended to process an unhandled "page" level exception by redirecting the exception to either the specified error page for this JSP, or if none was specified, to perform some implementation dependent action.

A JSP implementation class shall typically clean up any local state prior to invoking this and will return immediately thereafter. It is illegal to generate any output to the client, or to modify any ServletResponse state after invoking this call.

Parameters:
   t - the throwable to be handled

Throws:
   ServletException - if an error occurs while invoking the error page
   IOException - if an I/O error occurred while invoking the error page
   NullPointerException - if the exception is null

See Also:
   handlePageException(Exception)
public BodyContent pushBody()
BodyContent "out" JspWriter PageContext "out"

    return BodyContent

pushBody

public BodyContent pushBody()

    Return a new BodyContent object, save the current "out" JspWriter, and update the value of the "out" attribute in the page scope attribute namespace of the PageContext.

    Returns:
    the new BodyContent

public ErrorData getErrorData()

    return ErrorData Servlet isErrorPage "true"
    since 2.0

getErrorData

public ErrorData getErrorData()

    Provides convenient access to error information.

    Returns:
    an ErrorData instance containing information about the error, as obtained from the request attributes, as per the Servlet specification. If this is not an error page (that is, if the isErrorPage attribute of the page directive is not set to "true"), the information is meaningless.

    Since:
public abstract class PageData extends Object

JSP JSP XML

JSP TagLibraryValidator

Translation-time information on a JSP page. The information corresponds to the XML view of the JSP page.

Objects of this type are generated by the JSP translator, e.g. when being passed to a TagLibraryValidator instance.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>PageData()</td>
</tr>
<tr>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract InputStream getInputStream()&gt;Returns an input stream on the XML view of a JSP page.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Constructor Detail

public PageData()

PageData

public PageData()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

Method Detail

abstract public java.io.InputStream getInputStream()

JSP XML UTF-8 JSP XML include

    return

getInputStream

public abstract InputStream getInputStream()

    Returns an input stream on the XML view of a JSP page. The stream is encoded in UTF-8. Recall that the XML view of a JSP page has the include directives expanded.

    Returns: An input stream on the document.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
This class holds MIME parameters (attribute-value pairs). The `mail.mime.encodeparameters` and `mail.mime.decodeparameters` System properties control whether encoded parameters, as specified by RFC 2231, are supported. By default, such encoded parameters are not supported.

Also, in the current implementation, setting the System property `mail.mime.decodeparameters.strict` to "true" will cause a `ParseException` to be thrown for errors detected while decoding encoded parameters. By default, if any decoding errors occur, the original (undecoded) string is used.

**Version:**
1.17, 07/05/04

**Author:**
John Mani, Bill Shannon
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ParameterList()</code></td>
<td>No-arg Constructor.</td>
</tr>
<tr>
<td><code>ParameterList(String s)</code></td>
<td>Constructor that takes a parameter-list string.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>get(String name)</code></td>
<td>Returns the value of the specified parameter.</td>
</tr>
<tr>
<td><code>getNames()</code></td>
<td>Return an enumeration of the names of all parameters in this list.</td>
</tr>
<tr>
<td><code>remove(String name)</code></td>
<td>Removes the specified parameter from this ParameterList.</td>
</tr>
<tr>
<td><code>set(String name, String value)</code></td>
<td>Set a parameter.</td>
</tr>
<tr>
<td><code>set(String name, String value, String charset)</code></td>
<td>Set a parameter.</td>
</tr>
<tr>
<td><code>size()</code></td>
<td>Return the number of parameters in this list.</td>
</tr>
<tr>
<td><code>toString()</code></td>
<td>Convert this ParameterList into a MIME String.</td>
</tr>
<tr>
<td><code>toString(int used)</code></td>
<td>Convert this ParameterList into a MIME String.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public ParameterList()

**ParameterList**

public ParameterList()

   No-arg Constructor.

---

public ParameterList(String s) throws ParseException

Constructor that takes a parameter-list string. The String is parsed and the parameters are collected and stored internally. A ParseException is thrown if the parse fails. Note that an empty parameter-list string is valid and will be parsed into an empty ParameterList.

**Parameters:**

   s - the parameter-list string.

**Throws:**

ParseException - if the parse fails.

---

**Method Detail**

public int size()
size

public int size()

    Return the number of parameters in this list.

    Returns:
    number of parameters.

get

public String get(String name)

    name
    return

    get

public void set(String name, String value)

    name
    value
public void set(String name, String value)

Set a parameter. If this parameter already exists, it is replaced by this new value.

Parameters:
  name - name of the parameter.
  value - value of the parameter.

public void set(String name, String value, String charset)

RFC 2231

  name
  value
  charset

since JavaMail 1.4

Set a parameter. If this parameter already exists, it is replaced by this new value. If the mail.mime.encodeparameters System property is true, and the parameter value is non-ASCII, it will be encoded with the specified charset, as specified by RFC 2231.

Parameters:
  name - name of the parameter.
  value - value of the parameter.
public void remove(String name)
ParameterList

remove

public void remove(String name)

Removes the specified parameter from this ParameterList. This method does nothing if the parameter is not present.

Parameters:
name - name of the parameter.

public java.util.Enumeration<E> getNames()

return

getNames

public Enumeration getNames()

Return an enumeration of the names of all parameters in this list.

Returns:
Enumeration of all parameter names in this list.
**toString**

```java
public String toString()
```

Convert this ParameterList into a MIME String. If this is an empty list, an empty string is returned.

**Overrides:**
toString in class Object

**Returns:**
String

---

```java
public String toString(int used)
```

**ParameterList** MIME String

```
used
  return
```

**toString**

```java
public String toString(int used)
```

Convert this ParameterList into a MIME String. If this is an empty list, an empty string is returned. The 'used' parameter specifies the number of character positions already taken up in the field into which the resulting parameter list is to be inserted. It's used to determine where to fold the resulting parameter list.

**Parameters:**
used - number of character positions already used, in the field into which the parameter list is to be inserted.

**Returns:**
String
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.rpc Class ParameterMode

java.lang.Object
   javax.xml.rpc.ParameterMode

public class ParameterMode
   extends Object

javax.xml.rpc.ParameterMode Call API
   version 1.0
   See also javax.xml.rpc.Call

The javax.xml.rpc.ParameterMode is a type-safe enumeration for parameter mode. This class is used in the call API to specify parameter passing modes.

**Version:**
   1.0

**Author:**
   Rahul Sharma

**See Also:**
   Call

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static ParameterMode <strong>IN</strong></td>
</tr>
<tr>
<td>static ParameterMode <strong>INOUT</strong></td>
</tr>
<tr>
<td>static ParameterMode <strong>OUT</strong></td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String toString()</td>
<td>Returns a String describing this ParameterMode object.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Field Detail

IN

public static final ParameterMode IN

IN mode for parameter passing

OUT

public static final ParameterMode OUT

OUT mode for parameter passing

INOUT

public static final ParameterMode INOUT

INOUT mode for parameter passing
public String toString()

ParameterMode String

return

toString

public String toString()

Returns a String describing this ParameterMode object.

Overrides:

toString in class Object

Returns:

A string representation of the object.
javax.xml.bind Interface ParseConversionEvent

All Superinterfaces:
   ValidationEvent

All Known Implementing Classes:
   ParseConversionEventImpl

public interface ParseConversionEvent
extends ValidationEvent

Implements: ValidationEvent
Implemented by: ParseConversionEventImpl

XML  Java
version $Revision: 1.1 $
since JAXB1.0
See javax.xml.bind.ValidationEvent,
also javax.xml.bind.ValidationEventHandler,
javax.xml.bind.Unmarshaller

This event indicates that a problem was encountered while converting a
string from the XML data into a value of the target Java data type.

Since:
   JAXB1.0
Version:
   $Revision: 1.1 $
Author:
   - Ryan Shoemaker, Sun Microsystems, Inc.
   - Kohsuke Kawaguchi, Sun Microsystems, Inc.
   - Joe Fialli, Sun Microsystems, Inc.
See Also:
   ValidationEvent, ValidationEventHandler, Unmarshaller
Field Summary

Fields inherited from interface javax.xml.bind.ValidationEvent
ERROR, FATAL_ERROR, WARNING

Method Summary

Methods inherited from interface javax.xml.bind.ValidationEvent
getLinkedException, getLocator, getMessage, getSeverity

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
Class ParseConversionEventImpl

All Implemented Interfaces:
ParseConversionEvent, ValidationEvent

public class ParseConversionEventImpl
extends ValidationEventImpl
implements ParseConversionEvent

Extends: ValidationEventImpl
Implements: ParseConversionEvent

ParseConversionEvent

JAXB ValidationEvent

version $Revision: 1.1 $ en
since JAXB1.0

See also
javax.xml.bind.ParseConversionEvent,
javax.xml.bind.Validator,
javax.xml.bind.ValidationEventHandler,
javax.xml.bind.ValidationEvent,
javax.xml.bind.ValidationEventLocator

Default implementation of the ParseConversionEvent interface.

JAXB providers are allowed to use whatever class that implements the ValidationEvent interface. This class is just provided for a convenience.

Since:
JAXB1.0
**Field Summary**

Fields inherited from interface `javax.xml.bind.ValidationEvent`

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>FATAL_ERROR, WARNING</td>
</tr>
</tbody>
</table>

**Constructor Summary**

```java
ParseConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator)  
Create a new ParseConversionEventImpl.
```

```java
ParseConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)  
Create a new ParseConversionEventImpl.
```

**Method Summary**

Methods inherited from class `javax.xml.bind.helpers.ValidationEventImpl`

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getLinkedException</td>
<td>getLocator, getMessage, getSeverity, setLinkedException, setLocator, setMessage, setSeverity, toString</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone</td>
<td>equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.bind.ValidationEvent`
**Constructor Detail**

```java
public ParseConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator)
```

**ParseConversionEventImpl**

- `_severity` - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR
- `_message` - The text message for this event - may be null.
- `_locator` - The locator object for this event - may be null.

**Throws:**

IllegallegalArgumentException - if an illegal severity field is supplied

```java
public ParseConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)
```
ParseConversionEventImpl

public ParseConversionEventImpl(int _severity, 
String _message, 
ValidationEventLocator _locator, 
Throwable _linkedException)

Create a new ParseConversionEventImpl.

Parameters:

_severity - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR
_message - The text message for this event - may be null.
_locator - The locator object for this event - may be null.
_linkedException - An optional linked exception that may provide additional information about the event - may be null.

Throws:

IllegalArgumentException - if an illegal severity field is supplied
PS:
Class ParseException

javax.mail.internet.ParseException

java.lang.Object
  ▼ java.lang.Throwable
    ▼ java.lang.Exception
      ▼ javax.mail.MessagingException
        ▼ javax.mail.internet.ParseException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  AddressException

public class ParseException
  extends MessagingException

  Extends: Throwable > Exception > MessagingException
  Extended by: AddressException

RFC822 MIME

The exception thrown due to an error in parsing RFC822 or MIME headers

Author:
  John Mani

See Also:
  Serialized Form

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ParseException()</td>
<td>Constructs a ParseException with no detail message.</td>
</tr>
<tr>
<td>ParseException(String s)</td>
<td></td>
</tr>
</tbody>
</table>
Constructs a ParseException with the specified detail message.

### Method Summary

Methods inherited from class javax.mail.MessagingException
- `getCause`, `getNextException`, `setNextException`, `toString`

Methods inherited from class java.lang.Throwable
- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`

Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Constructor Detail

**public ParseException()**

**ParseException**

**ParseException**

**public ParseException()**

Constructs a ParseException with no detail message.

**public ParseException(String s)**

**ParseException**

s
ParseException

public ParseException(String s)

Constructs a ParseException with the specified detail message.

Parameters:
  s - the detail message
javax.mail Interface Part

All Known Subinterfaces: 
   MimePart

All Known Implementing Classes: 
   BodyPart, Message, MimeBodyPart, MimeMessage, PreencodedMimeBodyPart

public interface Part

Implemented by: BodyPart, Message, MimePart

Part Message BodyPart

Part "Content"

JavaMail API Part Mail Mail Part -
String

"content" getContentType() MIME

Part "content"

- DataHandler - getDataHandler() Part "content"
javax.activation.DataHandler DataHandler
- - getInputStream()
- Java - getContent() Java "content"
"multipart" Part Multipart "multipart" Part
getContentType() Multipart

Part writeTo() Part

Message BodyPart Part MIME Part
The Part interface is the common base interface for Messages and BodyParts.

Part consists of a set of attributes and a "Content".

**Attributes:**

The JavaMail API defines a set of standard Part attributes that are considered to be common to most existing Mail systems. These attributes have their own setter and getter methods. Mail systems may support other Part attributes as well, these are represented as name-value pairs where both the name and value are Strings.

**Content:**

The data type of the "content" is returned by the `getContentType()` method. The MIME typing system is used to name data types.

The "content" of a Part is available in various formats:

- As a DataHandler - using the `getDataHandler()` method. The "content" of a Part is also available through a `javax.activation.DataHandler` object. The DataHandler object allows clients to discover the operations available on the content, and to instantiate the appropriate component to perform those operations.
- As an input stream - using the `getInputStream()` method. Any mail-specific encodings are decoded before this stream is returned.
- As a Java object - using the `getContent()` method. This method returns the "content" as a Java object. The returned object is of course dependent on the content itself. In particular, a "multipart" Part's content is always a Multipart or subclass thereof. That is, `getContent()` on a "multipart" type Part will always return a Multipart (or subclass) object.

Part provides the `writeTo()` method that streams out its bytestream in
mail-safe form suitable for transmission. This bytestream is typically an aggregation of the Part attributes and its content's bytestream.

Message and BodyPart implement the Part interface. Note that in MIME parlance, Part models an Entity (RFC 2045, Section 2.4).

Author:
  John Mani

---

### Field Summary

<table>
<thead>
<tr>
<th>static String ATTACHMENT</th>
<th>This part should be presented as an attachment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String INLINE</td>
<td>This part should be presented inline.</td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>void addHeader(String header_name, String header_value)</th>
<th>Add this value to the existing values for this header_name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enumeration getAllHeaders()</td>
<td>Return all the headers from this part as an Enumeration of Header objects.</td>
</tr>
<tr>
<td>Object getContent()</td>
<td>Return the content as a Java object.</td>
</tr>
<tr>
<td>String getContentType()</td>
<td>Returns the Content-Type of the content of this part.</td>
</tr>
<tr>
<td>DataHandler getDataHandler()</td>
<td>Return a DataHandler for the content within this part.</td>
</tr>
<tr>
<td>String getDescription()</td>
<td>Return a description String for this part.</td>
</tr>
<tr>
<td>String getDisposition()</td>
<td>Return the disposition of this part.</td>
</tr>
<tr>
<td>String getFileName()</td>
<td></td>
</tr>
</tbody>
</table>
### Get the filename associated with this part, if possible.

**String[]**

**getHeader(String header_name)**
- Get all the headers for this header name.

**InputStream**

**getInputStream()**
- Return an input stream for this part's "content".

**int**

**getLineCount()**
- Return the number of lines in the content of this part.

**Enumeration**

**getMatchingHeaders(String[] header_names)**
- Return matching headers from this part as an Enumeration of Header objects.

**Enumeration**

**getNonMatchingHeaders(String[] header_names)**
- Return non-matching headers from this envelope as an Enumeration of Header objects.

**int**

**getSize()**
- Return the size of the content of this part in bytes.

**boolean**

**isMimeType(String mimeType)**
- Is this Part of the specified MIME type?

**void**

**removeHeader(String header_name)**
- Remove all headers with this name.

**void**

**setContent(Multipart mp)**
- This method sets the given Multipart object as this message's content.

**void**

**setContent(Object obj, String type)**
- A convenience method for setting this part's content.

**void**

**setDataHandler(DataHandler dh)**
- This method provides the mechanism to set this part's content.

**void**

**setDescription(String description)**
- Set a description String for this part.

**void**

**setDisposition(String disposition)**
- Set the disposition of this part.

**void**

**setFileName(String filename)**
- Set the filename associated with this part, if possible.

**void**

**setHeader(String header_name, String header_value)**
- Set the value for this header_name.
setText(String text)
A convenience method that sets the given String as this part's content with a MIME type of "text/plain".

writeTo(OutputStream os)
Output a bytestream for this Part.

Field Detail

ATTACHMENT
static final String ATTACHMENT
This part should be presented as an attachment.

See Also:
getDisposition(), setDisposition(java.lang.String), Constant Field Values

INLINE
static final String INLINE
This part should be presented inline.

See Also:
getDisposition(), setDisposition(java.lang.String), Constant Field Values

Method Detail
public int getSize() throws MessagingException
-1

    return
    Throws

getSize

int getSize()
throws MessagingException

    Return the size of the content of this part in bytes. Return -1 if the size cannot be determined.

    Note that the size may not be an exact measure of the content size and may or may not account for any transfer encoding of the content. The size is appropriate for display in a user interface to give the user a rough idea of the size of this part.

    Returns:
    size of content in bytes
    Throws:
    MessagingException

public int getLineCount() throws MessagingException
-1

    return
    Throws

getLineCount

int getLineCount()
throws MessagingException
Return the number of lines in the content of this part. Return -1 if the number cannot be determined. Note that this number may not be an exact measure of the content length and may or may not account for any transfer encoding of the content.

**Returns:**
- number of lines in the content.

**Throws:**
- MessagingException

---

```java
public String getContentType() throws MessagingException {
    return ContentType;
}
```

MIME Content-type

```
return
Throws
See also
```

**getContentType**

```java
String getContentType() throws MessagingException {
    return Content-Type;
}
```

Returns the Content-Type of the content of this part. Returns null if the Content-Type could not be determined.

The MIME typing system is used to name Content-types.

**Returns:**
- The ContentType of this part

**Throws:**
- MessagingException

**See Also:**
- javax.activation.DataHandler
public boolean isMimeType(String mimeType) throws MessagingException

Part MIME  primaryType  subType

"text/plain"  "text/plain; charset=foobar"

true

mimeType  subType  '*'  subtype

isMimeType

boolean isMimeType(String mimeType)

throws MessagingException

Is this Part of the specified MIME type? This method compares only the primaryType and subType. The parameters of the content types are ignored.

For example, this method will return true when comparing a Part of content type "text/plain" with "text/plain; charset=foobar".

If the subType of mimeType is the special character '*', then the subtype is ignored during the comparison.

Throws:
   MessagingException

public String getDisposition() throws MessagingException
disposition
disposition  RFC 2183

String disp = part.getDisposition();
if (disp == null || disp.equalsIgnoreCase(Part.ATTACHMENT))
    // treat as attachment if not first part

    return  disposition null
getDisposition

String getDisposition() throws MessagingException

Return the disposition of this part. The disposition describes how the part should be presented to the user. (See RFC 2183.) The return value should be considered without regard to case. For example:

```java
String disp = part.getDisposition();
if (disp == null || disp.equalsIgnoreCase(Part.ATTACHMENT))
   // treat as attachment if not first part
```

Returns:
- disposition of this part, or null if unknown

Throws:
- MessagingException

See Also:
- ATTACHMENT, INLINE, getFileName()

public void setDisposition(String disposition) throws MessagingException

disposition

Throws
- MessagingException
- IllegalWriteException:
- IllegalStateException: Part READONLY

See also
- ATTACHMENT, INLINE, setFileName

setDisposition

void setDisposition(String disposition)
Set the disposition of this part.

**Parameters:**
- disposition - disposition of this part

**Throws:**
- MessagingException
- IllegalWriteException - if the underlying implementation does not support modification of this header
- IllegalStateException - if this Part is obtained from a READ_ONLY folder

**See Also:**
- ATTACHMENT, INLINE, setFileName(java.lang.String)

---

```java
public String getDescription() throws MessagingException {
    return null;
}
```

**Description**

Return a description String for this part. This typically associates some descriptive information with this part. Returns null if none is available.

**Returns:**
- description of this part

**Throws:**
- MessagingException

---

```java
public void setDescription(String description) throws MessagingException {
}
```

**Description**

Set the disposition of this part.

**Parameters:**
- disposition - disposition of this part

**Throws:**
- MessagingException
- IllegalWriteException - if the underlying implementation does not support modification of this header
- IllegalStateException - if this Part is obtained from a READ_ONLY folder

**See Also:**
- ATTACHMENT, INLINE, setFileName(java.lang.String)
String

description

Throws MessagingException:

Throws IllegalArgumentException:

Throws IllegalStateException: Part READ_ONLY

setDescription

void setDescription(String description) throws MessagingException

Set a description String for this part. This typically associates some descriptive information with this part.

Parameters:

description - description of this part

Throws:

MessagingException

IllegalArgumentException - if the underlying implementation does not support modification of this header

IllegalStateException - if this Part is obtained from a READ_ONLY folder

public String getFileName() throws MessagingException

return

getFileName

String getFileName() throws MessagingException

Get the filename associated with this part, if possible. Useful if this part represents an "attachment" that was loaded from a file. The filename will usually be a simple name, not including directory
setFileName

void setFileName(String filename) throws MessagingException

""

filename

Throws IllegalWriteException:

Throws IllegalStateException: Part READ_ONLY

Parameters:
filename - Filename to associate with this part

Throws:

IllegalWriteException - if the underlying implementation does not support modification of this header
IllegalStateException - if this Part is obtained from a READ_ONLY folder
MessagingException

getInputStream

public java.io.InputStream getInputStream() throws java.io.IOException, MessagingException
public DataHandler getDataHandler() throws MessagingException

DataHandler

return InputStream

Throws java.io.IOException: DataHandler
javax.activation.DataHandler

Throws MessagingException:

See also getDataHandler, getInputStream

getInputStream

InputStream getInputStream()

throws IOException, MessagingException

Return an input stream for this part's "content". Any mail-specific transfer encodings will be decoded before the input stream is provided.

This is typically a convenience method that just invokes the DataHandler's getInputStream() method.

Returns:
an InputStream

Throws:
IOException - this is typically thrown by the DataHandler. Refer to the documentation for javax.activation.DataHandler for more details.
MessagingException

See Also:
getDataHandler(), DataHandler.getInputStream()
**getDataHandler**

```java
DataHandler getDataHandler() throws MessagingException
```

Return a DataHandler for the content within this part. The DataHandler allows clients to operate on as well as retrieve the content.

**Returns:**
- DataHandler for the content

**Throws:**
- MessagingException

---

**public Object getContent() throws java.io.IOException, MessagingException**

Java "text/plain" String "multipart"
Multipart DataHandler

```java
DataHandler getContent() throws MessagingException:
```

**Throws:**
- java.io.IOException: DataHandler
- javax.activation.DataHandler

See also
- getContent

---

**getContent**

```java
Object getContent()
```
Return the content as a Java object. The type of the returned object is of course dependent on the content itself. For example, the object returned for "text/plain" content is usually a String object. The object returned for a "multipart" content is always a Multipart subclass. For content-types that are unknown to the DataHandler system, an input stream is returned as the content.

This is a convenience method that just invokes the DataHandler's getContent() method

**Returns:**
Object

**Throws:**
- MessagingException
- IOException - this is typically thrown by the DataHandler. Refer to the documentation for javax.activation.DataHandler for more details.

**See Also:**
DataHandler.getContent()
DataHandler wraps around the actual content.

**Parameters:**

`dh` - The DataHandler for the content.

**Throws:**

- `MessagingException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this Part is obtained from a READ_ONLY folder

```java
public void setContent(Object obj, String type) throws MessagingException
    DataHandler
```

**DataContentHandler JavaMail**

```java
setContent("foobar", "application/x-foobar") "application/x-foobar" DataContentHandler Java Activation Framework
```

<table>
<thead>
<tr>
<th><code>obj</code></th>
<th>Java</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>type</code></td>
<td>MIME</td>
</tr>
</tbody>
</table>

**Throws**

- `IllegalWriteException`: 
- `IllegalStateException`: Part READ_ONLY

## setContent

```java
void setContent(Object obj,
                String type)
    throws MessagingException
```

A convenience method for setting this part's content. The part internally wraps the content in a DataHandler.

Note that a DataContentHandler class for the specified type should be available to the JavaMail implementation for this to work right.
i.e., to do `setContent(foobar, "application/x-foobar"),` a DataContentHandler for "application/x-foobar" should be installed. Refer to the Java Activation Framework for more information.

**Parameters:**
- `obj` - A java object.
- `type` - MIME type of this object.

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this Part is obtained from a `READ_ONLY` folder
- `MessagingException`

```java
public void setText(String text) throws MessagingException
MIME "text/plain" String
    text Message
Throws    IllegalWriteException:
Throws    IllegalStateException: Part `READ_ONLY`
```

A convenience method that sets the given String as this part's content with a MIME type of "text/plain".

**Parameters:**
- `text` - The text that is the Message's content.

**Throws:**
- `IllegalWriteException` - if the underlying implementation does not support modification of existing values
- `IllegalStateException` - if this Part is obtained from a `READ_ONLY` folder
- `MessagingException`
public void setContent(Multipart mp) throws MessagingException

Multipart

Throw: IllegalWriteException:

Thrown: IllegalStateException: Part READ_ONLY

**setContent**

void setContent(Multipart mp)

throws MessagingException

This method sets the given Multipart object as this message's content.

**Parameters:**

mp - The multipart object that is the Message's content

**Throws:**

IllegalWriteException - if the underlying implementation does not support modification of existing values

IllegalStateException - if this Part is obtained from a READ_ONLY folder

**Part**

Thrown: java.io.IOException: javax.activation

Thrown: MessagingException:

See also: writeTo
**writeTo**

```java
void writeTo(OutputStream os)
    throws IOException, MessagingException
```

Output a bytestream for this Part. This bytestream is typically an aggregation of the Part attributes and an appropriately encoded bytestream from its 'content'.

Classes that implement the Part interface decide on the appropriate encoding algorithm to be used.

The bytestream is typically used for sending.

**Throws:**

- `IOException` - if an error occurs writing to the stream or if an error is generated by the `javax.activation` layer.
- `MessagingException` - if an error occurs fetching the data to be written

**See Also:**

`DataHandler.writeTo(java.io.OutputStream)`

---

**public String[] getHeader(String header_name) throws MessagingException**

```java
null
```

```java
return
```  

**Throws**

`MessagingException`

---

**getHeader**

```java
String[] getHeader(String header_name)
    throws MessagingException
```
Get all the headers for this header name. Returns null if no headers for this header name are available.

Parameters:
- header_name - the name of this header

Returns:
- the value fields for all headers with this name

Throws:
- MessagingException

public void setHeader(String header_name, String header_value) throws MessagingException

header_name
- header_name
- header_value

Throws
- MessagingException:

Throws
- IllegalWriteException:

Throws
- IllegalStateException: Part READ_ONLY

setHeader

void setHeader(String header_name, String header_value) throws MessagingException

Set the value for this header_name. Replaces all existing header values with this new value.

Parameters:
- header_name - the name of this header
- header_value - the value for this header

Throws:
- MessagingException
- IllegalWriteException - if the underlying implementation does not support modification of existing values
- IllegalStateException - if this Part is obtained from a READ_ONLY folder
public void addHeader(String header_name, String header_value) throws MessagingException

header_name
  header_name
  header_value

Throws MessagingException:
Throws IllegalWriteException:
Throws IllegalStateException: Part READ_ONLY

addHeader

void addHeader(String header_name,
               String header_value)
throws MessagingException

Add this value to the existing values for this header_name.

Parameters:
  header_name - the name of this header
  header_value - the value for this header

Throws:
  MessagingException
  IllegalWriteException - if the underlying implementation does not support modification of existing values
  IllegalStateException - if this Part is obtained from a READ_ONLY folder

public void removeHeader(String header_name) throws MessagingException

header_name

Throws MessagingException:
Throws IllegalWriteException:
Throws IllegalStateException: Part READ_ONLY
removeHeader

void removeHeader(String header_name)
    throws MessagingException

Remove all headers with this name.

Parameters:
    header_name - the name of this header

Throws:
    MessagingException
    IllegalWriteException - if the underlying implementation does not support modification of existing values
    IllegalStateException - if this Part is obtained from a READ_ONLY folder

public java.utilEnumeration getHeaders() throws MessagingException

Header

return

Throws
    MessagingException:

getAllHeaders

Enumeration getHeaders()
    throws MessagingException

Return all the headers from this part as an Enumeration of Header objects.

Returns:
    enumeration of Header objects

Throws:
    MessagingException
public java.util.Enumeration<E> getMatchingHeaders(String[] header_names) throws MessagingException

Header

<table>
<thead>
<tr>
<th>return</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Throws: MessagingException

getMatchingHeaders

Enumeration getMatchingHeaders(String[] header_names) throws MessagingException

Returns:

- enumeration of Header objects

Throws:

- MessagingException

public java.util.Enumeration<E> getNonMatchingHeaders(String[] header_names) throws MessagingException

Header

<table>
<thead>
<tr>
<th>return</th>
<th>Header</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Throws: MessagingException

getNonMatchingHeaders

Enumeration getNonMatchingHeaders(String[] header_names) throws MessagingException

Returns:

Return non-matching headers from this envelope as an Enumeration of Header objects.
Returns:
   enumeration of Header objects

Throws:
   MessagingException
class PasswordAuthentication
extends Object

PasswordAuthentication Authenticator

version 1.6, 05/04/07
See java.net.PasswordAuthentication,
also javax.mail.Authenticator, getPasswordAuthentication()

The class PasswordAuthentication is a data holder that is used by Authenticator. It is simply a repository for a user name and a password.

Version:
1.6, 05/04/07
Author:
Bill Foote
See Also:
PasswordAuthentication, Authenticator, Authenticator.getPasswordAuthentication()

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PasswordAuthentication</td>
<td>(String userName, String password)</td>
<td>Initialize a new PasswordAuthentication</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>getPassword()</td>
<td>}</td>
</tr>
</tbody>
</table>
String **getUserName**()

<table>
<thead>
<tr>
<th>Constructor Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public PasswordAuthentication(String userName, String password)</td>
</tr>
<tr>
<td>PasswordAuthentication(userName, password)</td>
</tr>
</tbody>
</table>

Initialize a new PasswordAuthentication

**Parameters:**
- userName - the user name
- password - The user's password

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public String getUserName()</td>
</tr>
<tr>
<td>getUserName</td>
</tr>
</tbody>
</table>
public String getUserName()

    Returns:
    the user name

public String getPassword()
    return

getPassword

public String getPassword()

    Returns:
    the password

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi.security Class PasswordCredential

java.lang.Object
  └─javax.resource.spi.security.PasswordCredential

All Implemented Interfaces:
  Serializable

public final class PasswordCredential

extends Object
implements Serializable

Implements: java.io.Serializable

PasswordCredential
  version 0.6
  since 0.6
See also javax.resource.spi.ManagedConnectionFactory

The class PasswordCredential acts as a holder for username and password.

Since:
  0.6
Version:
  0.6
Author:
  Rahul Sharma
See Also:
  ManagedConnectionFactory, Serialized Form

Constructor Summary

PasswordCredential(String userName, char[] password)
Creates a new PasswordCredential object from the given user name and password.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>equals(Object other)</code></td>
<td>Compares this PasswordCredential with the specified object for equality.</td>
</tr>
<tr>
<td><code>getManagedConnectionFactory()</code></td>
<td>Gets the target ManagedConnectionFactory for which the user name and password has been set by the application server.</td>
</tr>
<tr>
<td><code>getPassword()</code></td>
<td>Returns the user password.</td>
</tr>
<tr>
<td><code>getUserName()</code></td>
<td>Returns the user name.</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td>Returns the hash code for this PasswordCredential.</td>
</tr>
<tr>
<td><code>setManagedConnectionFactory(ManagedConnectionFactory)</code></td>
<td>Sets the target ManagedConnectionFactory instance for which the user name and password has been set by the application server.</td>
</tr>
</tbody>
</table>

## Constructor Detail

```java
public PasswordCredential(String userName, char[] password)
```
PasswordCredential

public PasswordCredential(String userName, char[] password)

Creates a new PasswordCredential object from the given user name and password.

Note that the given user password is cloned before it is stored in the new PasswordCredential object.

Parameters:

- userName - the user name
- password - the user's password

Method Detail

public String getUserName()

return

getUserName

public String getUserName()

Returns the user name.

Returns:

the user name

public char[] getPassword()
getPassword

public char[] getPassword()

    Returns the user password.

    Note that this method returns a reference to the password. It is the
caller's responsibility to zero out the password information after it is
no longer needed.

    Returns:
    the password

public ManagedConnectionFactory getManagedConnectionFactory()
ManagedConnectionFactory EIS
return ManagedConnectionFactory

getManagedConnectionFactory

public ManagedConnectionFactory getManagedConnectionFactory()()

    Gets the target ManagedConnectionFactory for which the user name
and password has been set by the application server. A
ManagedConnection- Factory uses this field to find out whether
PasswordCredential should be used by it for sign-on to the target
EIS instance.

Returns:
ManagedConnectionFactory instance for which user name and password have been specified

```
public void setManagedConnectionFactory(ManagedConnectionFactory mcf)
```

`setManagedConnectionFactory` sets the target ManagedConnectionFactory instance for which the user name and password have been set by the application server.

**Parameters:**
- `mcf` - ManagedConnectionFactory instance for which user name and password have been specified

```
public boolean equals(Object other)
```

`equals` compares this PasswordCredential with the specified object for
equality. The two PasswordCredential instances are the same if they are equal in username and password.

Overrides:

equals in class Object

Parameters:

other - Object to which PasswordCredential is to be compared

Returns:

true if and if the specified object is a PasswordCredential whose username and password are equal to this instance.

public int hashCode()
PasswordCredential

return PasswordCredential

hashCode

public int hashCode()

Returns the hash code for this PasswordCredential

Overrides:

hashCode in class Object

Returns:

hash code for this PasswordCredential
PS:
javax.annotation.security Annotation Type PermitAll

@Documented
@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})
public @interface PermitAll

Implements: Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})

"" RolesAllowed PermitAll
RolesAllowed

\[since\] 1.0
\[See\] javax.annotation.security.RolesAllowed,
\[also\] javax.annotation.security.DenyAll

Specifies that all security roles are allowed to invoke the specified method(s) i.e that the specified method(s) are "unchecked". It can be specified on a class or on methods. Specifying it on the class means that it applies to all methods of the class. If specified at the method level, it only affects that method. If the RolesAllowed is specified at the class level and this annotation is applied at the method level, the PermitAll annotation overrides the RolesAllowed for the specified method.

\[Since\]:
1.0
\[See Also:\]
RolesAllowed, DenyAll

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public class Persistence
extends Object

Bootstrap class that is used to obtain an EntityManagerFactory.

Since: Java Persistence 1.0

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>PERSISTENCE_PROVIDER</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected Set&lt;PersistenceProvider&gt;</td>
<td>providers</td>
</tr>
</tbody>
</table>

Constructor Summary

Persistence()

Method Summary

static EntityManagerFactory createEntityManagerFactory(String persistenceUnitName) Create and return an EntityManagerFactory
<table>
<thead>
<tr>
<th>static EntityManagerFactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>createEntityManagerFactory(String persistenceUnitName, Map properties)</td>
</tr>
<tr>
<td>Create and return an EntityManagerFactory named persistence unit using the given properties.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Field Detail

**PERSISTENCE_PROVIDER**

public static final String PERSISTENCE_PROVIDER

See Also:  
Constant Field Values

providers

protected static final Set<PersistenceProvider> providers

### Constructor Detail

public Persistence()
```java
public Persistence()

---

**Method Detail**

```java
public static EntityManagerFactory createEntityManagerFactory(String persistenceUnitName)

 EntityManagerFactory persistenceUnitName
 return

createEntityManagerFactory
```

Create and return an EntityManagerFactory for the named persistence unit.

**Parameters:**
- `persistenceUnitName` - The name of the persistence unit

**Returns:**
The factory that creates EntityManagers configured according to the specified persistence unit

```java
public static EntityManagerFactory createEntityManagerFactory(String persistenceUnitName,
java.util.Map<K, V> properties)

 EntityManagerFactory persistenceUnitName
 properties
 return

createEntityManagerFactory
```

createEntityManagerFactory

public static EntityManagerFactory createEntityManagerFactory(String persistenceUnitName, Map<String, String> properties)

Create and return an EntityManagerFactory for the named persistence unit using the given properties.

**Parameters:**
- persistenceUnitName - The name of the persistence unit
- properties - Additional properties to use when creating the factory. The values of these properties override any values that may have been configured elsewhere.

**Returns:**
The factory that creates EntityManagers configured according to the specified persistence unit.
javax.persistence Annotation Type PersistenceContext

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface PersistenceContext

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

EntityManager

since Java Persistence 1.0

Expresses a dependency on an EntityManager persistence context.

Since:
Java Persistence 1.0

Optional Element Summary

<table>
<thead>
<tr>
<th>Optional Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>The name by which the entity manager is to be accessed in the environment referencing context, and is not needed when dependency injection is used.</td>
</tr>
<tr>
<td>properties</td>
<td>PersistenceProperty[]</td>
<td>Used to specify properties for the container or persistence provider.</td>
</tr>
<tr>
<td>type</td>
<td>PersistenceContextType</td>
<td>Specifies whether this is a transaction-scoped persistence context or an extended persistence context.</td>
</tr>
<tr>
<td>unitName</td>
<td>String</td>
<td>The name of the persistence unit.</td>
</tr>
</tbody>
</table>
abstract public String name()

name

public abstract String name

The name by which the entity manager is to be accessed in the environment referencing context, and is not needed when dependency injection is used.

Default:

    ""

abstract public String unitName()

unitName JNDI

unitName

public abstract String unitName

The name of the persistence unit. If the unitName element is specified, the persistence unit for the entity manager that is accessible in JNDI must have the same name.

Default:

    ""

abstract public PersistenceContextType type()
type

public abstract PersistenceContextType type

Specifies whether this is a transaction-scoped persistence context or an extended persistence context.

Default:
TRANSACTION

abstract public PersistenceProperty[] properties()

properties

public abstract PersistenceProperty[] properties

Used to specify properties for the container or persistence provider. Vendor specific properties may be included in this set of properties. Properties that are not recognized by a vendor are ignored.

Default:
{}
javax.persistence Annotation Type PersistenceContexts

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
public @interface PersistenceContexts

Implemented: Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

PersistenceContext[] value

Declares one or more PersistenceContext annotations. It is used to express a dependency on container-managed entity manager persistence contexts.

Since:  
Java Persistence 1.0

Required Element Summary

| PersistenceContext[] value | One or more persistence context |

Element Detail

abstract public PersistenceContext[] value()
public abstract PersistenceContext[] value

One or more persistence context
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>ENUM CONSTANTS</td>
<td>FIELD</td>
<td>METHOD</td>
</tr>
<tr>
<td>DETAIL: ENUM CONSTANTS</td>
<td>FIELD</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.persistence Enum PersistenceContextType

java.lang.Object
    java.lang.Enum<PersistenceContextType>
    javax.persistence.PersistenceContextType

All Implemented Interfaces:
    Serializable, Comparable<PersistenceContextType>

public enum PersistenceContextType
    extends Enum<PersistenceContextType>

Extends: Enum<E>

PersistenceContext type since Java Persistence 1.0

Specifies whether a transaction-scoped or extended persistence context is to be used in PersistenceContext. If the type element is not specified, a transaction-scoped persistence context is used.

Since:
    Java Persistence 1.0

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTENDED</td>
<td>Extended persistence context</td>
</tr>
<tr>
<td>TRANSACTION</td>
<td>Transaction-scoped persistence context</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueOf(String)</td>
<td>(String name)</td>
</tr>
<tr>
<td>static PersistenceContextType</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>static PersistenceContextType[]</td>
<td>values() Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Enum
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

### Methods inherited from class java.lang.Object
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Enum Constant Detail

### TRANSACTION

```java
public static final PersistenceContextType TRANSACTION
```

Transaction-scoped persistence context

### EXTENDED

```java
public static final PersistenceContextType EXTENDED
```

Extended persistence context

## Method Detail
final public static PersistenceContextType[] values()

values

public static final PersistenceContextType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(PersistenceContextType c : PersistenceContextType.values())
    System.out.println(c);
```

Returns:
- an array containing the constants of this enum type, in the order they're declared

public static PersistenceContextType valueOf(String name)

valueOf

public static PersistenceContextType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
- name - the name of the enum constant to be returned.

Returns:
- the enum constant with the specified name

Throws:
- IllegalArgumentException - if this enum type has no constant with the specified name
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.persistence  Class PersistenceException

java.lang.Object  
  └ java.lang.Throwable  
    └ java.lang.Exception  
      └ java.lang.RuntimeException  
        └ javax.persistence.PersistenceException

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  EntityExistsException, EntityNotFoundException, NonUniqueResultException, NoResultException, OptimisticLockException, RollbackException, TransactionRequiredException

public class PersistenceException
  extends RuntimeException

Extends: Throwable > Exception > RuntimeException
Extended by: EntityExistsException, EntityNotFoundException, NonUniqueResultException, NoResultException, OptimisticLockException, RollbackException, TransactionRequiredException

PersistenceException NoResultException
NonUniqueResultException

since  Java Persistence 1.0

Thrown by the persistence provider when a problem occurs. All instances of PersistenceException except for instances of NoResultException and NonUniqueResultException will cause the current transaction, if one is active, to be marked for rollback.

Since:
Java Persistence 1.0

See Also:
   Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PersistenceException()</td>
<td>Constructs a new PersistenceException exception with null as its detail message.</td>
</tr>
<tr>
<td>PersistenceException(String</td>
<td>Constructs a new PersistenceException exception with the specified detail message.</td>
</tr>
<tr>
<td>message)</td>
<td></td>
</tr>
<tr>
<td>PersistenceException(String,</td>
<td>Constructs a new PersistenceException exception with the specified detail message and cause.</td>
</tr>
<tr>
<td>Throwable)</td>
<td></td>
</tr>
<tr>
<td>PersistenceException(Throwable)</td>
<td>Constructs a new PersistenceException exception with the specified cause.</td>
</tr>
</tbody>
</table>
PersistenceException

public PersistenceException()

    Constructs a new PersistenceException exception with null as its detail message.

------------------------------------------------------------------------

public PersistenceException(String message)

    PersistenceException
    message

PersistenceException

public PersistenceException(String message)

    Constructs a new PersistenceException exception with the specified detail message.

    Parameters:
    message - the detail message.

------------------------------------------------------------------------

public PersistenceException(String message, Throwable cause)

    PersistenceException
    message
    cause

PersistenceException

public PersistenceException(String message,
Constructs a new PersistenciaException exception with the specified detail message and cause.

Parameters:
message - the detail message.
cause - the cause.

public PersistenceException(Throwable cause)

PersistenciaException

public PersistenceException(Throwable cause)

Constructs a new PersistenceException exception with the specified cause.

Parameters:
cause - the cause.
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: REQUIRED</td>
<td>OPTIONAL</td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
</tr>
<tr>
<td>DETAIL: ELEMENT</td>
<td></td>
</tr>
</tbody>
</table>
(javax.persistence) Annotation Type PersistenceProperty

@Target(value={})
@Retention(value=RUNTIME)
public @interface PersistenceProperty

Implements: Annotation
@Target(value={})
@Retention(value=RUNTIME)

since Java Persistence 1.0

Describes a single container or persistence provider property.

Vendor specific properties may be included in the set of properties, and are passed to the persistence provider by the container when the entity manager is created. Properties that are not recognized by a vendor will be ignored.

Since:
Java Persistence 1.0

Required Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>The name of the property</td>
</tr>
<tr>
<td>String</td>
<td>value</td>
<td>The value of the property</td>
</tr>
</tbody>
</table>
abstract public String name()

name

public abstract String name

The name of the property

abstract public String value()

value

public abstract String value

The value of the property
public interface PersistenceProvider

Interface implemented by a persistence provider. The implementation of this interface that is to be used for a given EntityManager is specified in persistence.xml file in the persistence archive. This interface is invoked by the Container when it needs to create an EntityManagerFactory, or by the Persistence class when running outside the Container.

Since:
Java Persistence 1.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createContainerEntityManagerFactory</code> &lt;i&gt;(PersistenceUnit info, Map)&lt;/i&gt;</td>
</tr>
<tr>
<td>Called by the container when an EntityManagerFactory be created.</td>
</tr>
<tr>
<td><code>createEntityManagerFactory</code> &lt;i&gt;(String emName, Map)&lt;/i&gt;</td>
</tr>
<tr>
<td>Called by Persistence class when an EntityManager is to be created.</td>
</tr>
</tbody>
</table>

Method Detail

public EntityManagerFactory
createEntityManagerFactory(String emName, java.util.Map<K, V> map)

javax.persistence.EntityManagerFactory Persistence

text

createEntityManagerFactory

EntityManagerFactory createEntityManagerFactory(String emName, Map map)

Called by Persistence class when an EntityManagerFactory is to be created.

Parameters:
emName - The name of the persistence unit
map - A Map of properties for use by the persistence provider. These properties may be used to override the values of the corresponding elements in the persistence.xml file or specify values for properties not specified in the persistence.xml.

Returns:
EntityManagerFactory for the persistence unit, or null if the provider is not the right provider

public EntityManagerFactory createContainerEntityManagerFactory(PersistenceUnitInfo info, java.util.Map<K, V> map)

javax.persistence.EntityManagerFactory

EntityManagerFactory

null

EntityManagerFactory
createContainerEntityManagerFactory

EntityManagerFactory createContainerEntityManagerFactory(PersistenceUnitInfo info, Map map)

Called by the container when an EntityManagerFactory is to be created.

Parameters:
  info - Metadata for use by the persistence provider
  map - A Map of integration-level properties for use by the persistence provider. Can be null if there is no integration-level property.

Returns:
  EntityManagerFactory for the persistence unit specified by the metadata

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.persistence  **Annotation Type PersistenceUnit**

```java
@Target({TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface PersistenceUnit
```

**Implements:** Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

Expresses a dependency on an [EntityManagerFactory](https://docs.oracle.com/javase/8/docs/api/javax/persistence/EntityManagerFactory.html).

**Since:**
Java Persistence 1.0

---

## Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>name</th>
<th>The name by which the entity manager factory is to be accessed in the environment referencing context, and is not needed when dependency injection is used.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>unitName</td>
<td>The name of the persistence unit as defined in the persistence.xml file.</td>
</tr>
</tbody>
</table>

```java
abstract public String name()
```

name
public abstract String name

The name by which the entity manager factory is to be accessed in the environment referencing context, and is not needed when dependency injection is used.

Default: 

abstract public String unitName()
persistence.xml  JNDI

unitName

public abstract String unitName

The name of the persistence unit as defined in the persistence.xml file. If specified, the persistence unit for the entity manager factory that is accessible in JNDI must have the same name.

Default: 

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**Interface** PersistenceUnitInfo

```java
public interface PersistenceUnitInfo
```

Since: Java Persistence 1.0

Interface implemented by the container and used by the persistence provider when creating an `EntityManagerFactory`.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addTransformer(ClassTransformer transformer)</code></td>
<td>Add a transformer supplied by the provider that will be called for every new class definition or class redefinition that gets loaded by the loader returned by the <code>getClassLoader()</code> method.</td>
</tr>
<tr>
<td><code>boolean excludeUnlistedClasses()</code></td>
<td>Returns whether classes in the root of the persistence unit that have not been explicitly listed are to be included in the set of managed classes.</td>
</tr>
<tr>
<td><code>ClassLoader getClassLoader()</code></td>
<td>Returns ClassLoader that the provider may use to load any classes, resources, or open URLs.</td>
</tr>
<tr>
<td><code>List&lt;URL&gt; getJarFileUrls()</code></td>
<td>Returns a list of URLs for the jar files or exploded jar file directories that the persistence</td>
</tr>
</tbody>
</table>
The provider must examine for managed classes of the persistence unit.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataSource.getJtaDataSource()</td>
<td>Returns the JTA-enabled data source to be used by the persistence provider.</td>
</tr>
<tr>
<td>List&lt;String&gt;.getManagedClassNames()</td>
<td>Returns the list of the names of the classes that the persistence provider must add it to its set of managed classes.</td>
</tr>
<tr>
<td>List&lt;String&gt;.getMappingFileNames()</td>
<td>Returns the list of mapping file names that the persistence provider must load to determine the mappings for the entity classes.</td>
</tr>
<tr>
<td>ClassLoader.getNewTempClassLoader()</td>
<td>Return a new instance of a ClassLoader that the provider may use to temporarily load any classes, resources, or open URLs.</td>
</tr>
<tr>
<td>DataSource.getNonJtaDataSource()</td>
<td>Returns the non-JTA-enabled data source to be used by the persistence provider for accessing data outside a JTA transaction.</td>
</tr>
<tr>
<td>String.getPersistenceProviderClassName()</td>
<td>Returns the fully qualified name of the persistence provider implementation class.</td>
</tr>
<tr>
<td>String.getPersistenceUnitName()</td>
<td>Returns the name of the persistence unit</td>
</tr>
<tr>
<td>URL.getPersistenceUnitRootUrl()</td>
<td>Returns the URL for the jar file or directory that is the root of the persistence unit.</td>
</tr>
<tr>
<td>Properties.getProperties()</td>
<td>Returns properties object.</td>
</tr>
<tr>
<td>PersistenceUnitTransactionType.getTransactionType()</td>
<td>Returns the transaction type of the entity managers created by the EntityManagerFactory.</td>
</tr>
</tbody>
</table>
**public String getPersistenceUnitName()**

`persistence.xml`
```
    return persistence.xml
```

**getPersistenceUnitName**

`String getPersistenceUnitName()`

Returns the name of the persistence unit. Corresponds to the name attribute in the persistence.xml file.

**Returns:**

The name of the persistence unit. Corresponds to the name attribute in the persistence.xml file.

---

**public String getPersistenceProviderClassName()**

`persistence.xml <provider>`
```
    return persistence.xml <provider>
```

**getPersistenceProviderClassName**

`String getPersistenceProviderClassName()`

Returns the fully qualified name of the persistence provider implementation class. Corresponds to the `<provider>` element in the persistence.xml file.

**Returns:**

The fully qualified name of the persistence provider implementation class. Corresponds to the `<provider>` element in the persistence.xml file.
public PersistenceUnitTransactionType getTransactionType()

EntityManagerFactory persistence.xml
    return EntityManagerFactory persistence.xml

getTransactionType

PersistenceUnitTransactionType getTransactionType()

Returns the transaction type of the entity managers created by the EntityManagerFactory. The transaction type corresponds to the transaction-type attribute in the persistence.xml file.

Returns:
    The transaction type of the entity managers created by the EntityManagerFactory. The transaction type corresponds to the transaction-type attribute in the persistence.xml file.

public javax.sql.DataSource getJtaDataSource()
JTA persistence.xml <jta-data-source>

    return JTA persistence.xml <jta-data-source>

getJtaDataSource

DataSource getJtaDataSource()

Returns the JTA-enabled data source to be used by the persistence provider. The data source corresponds to the <jta-data-source> element in the persistence.xml file or is provided at deployment or by the container.

Returns:
    the JTA-enabled data source to be used by the persistence provider. The data source corresponds to the <jta-data-source>
element in the persistence.xml file or is provided at deployment or by the container.

public javax.sql.DataSource getNonJtaDataSource()
JTA persistence.xml <non-jta-data-source>

return JTA persistence.xml <non-jta-data-source>

getNonJtaDataSource

Returns the non-JTA-enabled data source to be used by the persistence provider for accessing data outside a JTA transaction. The data source corresponds to the named <non-jta-data-source> element in the persistence.xml file or provided at deployment or by the container.

Returns:
The non-JTA-enabled data source to be used by the persistence provider for accessing data outside a JTA transaction. The data source corresponds to the named <non-jta-data-source> element in the persistence.xml file or provided at deployment or by the container.

public java.util.List<E> getMappingFileNames()
XML orm.xml persistence.xml <mapping-file>

return XML orm.xml persistence.xml <mapping-file>

getMappingFileNames
List<String> getMappingFileNames()

Returns the list of mapping file names that the persistence provider must load to determine the mappings for the entity classes. The mapping files must be in the standard XML mapping format, be uniquely named and be resource-loadable from the application classpath. This list will not include the orm.xml file if one was specified. Each mapping file name corresponds to a <mapping-file> element in the persistence.xml file.

Returns:
The list of mapping file names that the persistence provider must load to determine the mappings for the entity classes. The mapping files must be in the standard XML mapping format, be uniquely named and be resource-loadable from the application classpath. This list will not include the orm.xml file if one was specified. Each mapping file name corresponds to a <mapping-file> element in the persistence.xml file.

public java.util.List<E> getJarFileUrls()

Returns a list of URLs for the jar files or exploded jar file directories that the persistence provider must examine for managed classes of the persistence unit. Each URL corresponds to a named element in the persistence.xml file. A URL will either be a file: URL referring to a jar file or referring to a directory that contains an exploded jar file, or some other URL from which an InputStream in jar format can be obtained.
Returns:
a list of URL objects referring to jar files or directories.

```java
public java.net.URL getPersistenceUnitRootUrl()
    jar URL WEB-INF/classes URL
    jar jar file:URL jar InputStream URL
    return jar URL
```

**getPersistenceUnitRootUrl**

```java
URL getPersistenceUnitRootUrl()
```

Returns the URL for the jar file or directory that is the root of the persistence unit. (If the persistence unit is rooted in the WEB-INF/classes directory, this will be the URL of that directory.) The URL will either be a file: URL referring to a jar file or referring to a directory that contains an exploded jar file, or some other URL from which an InputStream in jar format can be obtained.

**Returns:**
a URL referring to a jar file or directory.

```java
public java.util.List<E> getManagedClassNames()
persistence.xml <class>
    return persistence.xml <class>
```

**getManagedClassNames**

```java
List<String> getManagedClassNames()
```

Returns the list of the names of the classes that the persistence provider must add it to its set of managed classes. Each name corresponds to a named <class> element in the persistence.xml file.
Returns:
The list of the names of the classes that the persistence provider must add it to its set of managed classes. Each name corresponds to a named <class> element in the persistence.xml file.

public boolean excludeUnlistedClasses()
persistence.xml  <exclude-unlisted-classes>
  return  persistence.xml  <exclude-unlisted-classes>

excludeUnlistedClasses

boolean excludeUnlistedClasses()

  Returns whether classes in the root of the persistence unit that have not been explicitly listed are to be included in the set of managed classes. This value corresponds to the <exclude-unlisted-classes> element in the persistence.xml file.

  Returns:
  Whether classes in the root of the persistence unit that have not been explicitly listed are to be included in the set of managed classes. This value corresponds to the <exclude-unlisted-classes> element in the persistence.xml file.

public java.util.Properties getProperties()
persistence.xml  <property>
  return  persistence.xml  <property>

getProperties

Properties getProperties()

  Returns properties object. Each property corresponds to a
<property> element in the persistence.xml file

**Returns:**
Properties object. Each property corresponds to a <property> element in the persistence.xml file

---

```java
public ClassLoader getClassLoader()
```

**ClassLoader getClassLoader()**

Returns ClassLoader that the provider may use to load any classes, resources, or open URLs.

**Returns:**
ClassLoader that the provider may use to load any classes, resources, or open URLs.

---

```java
public void addTransformer(ClassTransformer transformer)
```

**PersistenceUnitInfo#getClassLoader**

**PersistenceUnitInfo#getNewTempClassLoader**

**addTransformer**

```java
void addTransformer(ClassTransformer transformer)
```

Add a transformer supplied by the provider that will be called for every new class definition or class redefinition that gets loaded by the loader returned by the `getClassLoader()` method. The transformer has no effect on the result returned by the
getNewTempClassLoader() method. Classes are only transformed once within the same classloading scope, regardless of how many persistence units they may be a part of.

**Parameters:**

transformer - A provider-supplied transformer that the Container invokes at class-(re)definition time

```java
public ClassLoader getNewTempClassLoader()
```

URL ClassLoader

PersistenceUnitInfo#getClassLoader
PersistenceProvider#createContainerEntityManagerFactory(ClassLoader

```java
return ClassLoader
```

getNewTempClassLoader

ClassLoader getNewTempClassLoader()

```
ClassLoader
```

Return a new instance of a ClassLoader that the provider may use to temporarily load any classes, resources, or open URLs. The scope and classpath of this loader is exactly the same as that of the loader returned by getClassLoader(). None of the classes loaded by this class loader will be visible to application components. The provider may only use this ClassLoader within the scope of the PersistenceProvider.createContainerEntityManagerFactory(javax.persistence.spi.PersistenceUnitInfo, java.util.Map) call.

**Returns:**

Temporary ClassLoader with same visibility as current loader
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type PersistenceUnits

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface PersistenceUnits

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

PersistenceUnit

**since** Java Persistence 1.0

Declares one or more PersistenceUnit annotations.

**Since:**
Java Persistence 1.0

---

**Required Element Summary**

<table>
<thead>
<tr>
<th>PersistenceUnit[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>One or more PersistenceUnit annotations.</td>
</tr>
</tbody>
</table>

---

**Element Detail**

abstract public PersistenceUnit[] value()

PersistenceUnit

value

public abstract PersistenceUnit[] value
One or more PersistenceUnit annotations.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>ENUM CONSTANTS</td>
</tr>
</tbody>
</table>
javax.persistence.spi  Enum
PersistenceUnitTransactionType

java.lang.Object
   ↓  java.lang.Enum<PersistenceUnitTransactionType>
      ↓  javax.persistence.spi.PersistenceUnitTransactionType

All Implemented Interfaces:
Serializable, Comparable<PersistenceUnitTransactionType>

public enum PersistenceUnitTransactionType
   extends Enum<PersistenceUnitTransactionType>

Extends: Enum<E>

   javax.persistence.EntityManagerFactory  JTA
since  Java Persistence 1.0

This enum class defines whether the entity managers created by the
EntityManagerFactory will be JTA or resource-local entity managers.

Since:  
Java Persistence 1.0

<table>
<thead>
<tr>
<th>Enum Constant Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTA</td>
</tr>
<tr>
<td>JTA entity manager</td>
</tr>
</tbody>
</table>

| RESOURCE_LOCAL       |
| resource-local entity manager |

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
valueOf(String name) Returns the enum constant of this type with the specified name.

values() Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

JTA
public static final PersistenceUnitTransactionType JTA

   JTA entity manager

RESOURCE_LOCAL
public static final PersistenceUnitTransactionType RESOURCE_LOCAL

   resource-local entity manager

Method Detail
### values

```java
public static final PersistenceUnitTransactionType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(PersistenceUnitTransactionType c : PersistenceUnitTransactionType.values())
    System.out.println(c);
```

**Returns:**
- an array containing the constants of this enum type, in the order they're declared

### valueOf

```java
public static PersistenceUnitTransactionType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
- name - the name of the enum constant to be returned.

**Returns:**
- the enum constant with the specified name

**Throws:**
- `IllegalArgumentException` - if this enum type has no constant
with the specified name

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel  Interface PersonName

public interface PersonName

Represents a person's name.

Author:
   Farrukh S. Najmi

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
</tr>
<tr>
<td><strong>getFirstName</strong>()</td>
</tr>
<tr>
<td>Gets the first name for this Person.</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td><strong>getFullName</strong>()</td>
</tr>
<tr>
<td>Gets the fully formatted name for this person.</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td><strong>getLastName</strong>()</td>
</tr>
<tr>
<td>Gets the last name (surname) for this Person.</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td><strong>getMiddleName</strong>()</td>
</tr>
<tr>
<td>Gets the middle name for this Person.</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td><strong>setFirstName</strong>(String  firstName)</td>
</tr>
<tr>
<td>Sets the first name for this Person.</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td><strong>setFullName</strong>(String  fullName)</td>
</tr>
<tr>
<td>Sets the fully formatted name for this person.</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td><strong>setLastName</strong>(String  lastName)</td>
</tr>
<tr>
<td>Sets the last name (surname) for this Person.</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td><strong>setMiddleName</strong>(String  middleName)</td>
</tr>
<tr>
<td>Sets the middle name for this Person.</td>
</tr>
</tbody>
</table>
Method Detail

public String getLastName() throws JAXRException

Returns: the person's last name
Throws: JAXRException - If the JAXR provider encounters an internal error

public void setLastName(String lastName) throws JAXRException

Throws: JAXRException: JAXR
setLastName

void setLastName(String lastName) throws JAXRException

Sets the last name (surname) for this Person.

Capability Level: 1

Parameters:
    lastName - the person's last name

Throws:
    JAXRException - If the JAXR provider encounters an internal error

public String getFirstName() throws JAXRException

Person

1

return

Throws JAXRException: JAXR

gFirstName

String getFirstName() throws JAXRException

 Gets the first name for this Person. Default is an empty String.

Capability Level: 1

Returns:
    the person's first name

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setFirstName(String firstName) throws JAXRException
Person

1

firstName

Throws JAXRException: JAXR

setFirstName

void setFirstName(String firstName)
throws JAXRException

Sets the first name for this Person.

Capability Level: 1

Parameters:
firstName - the person's first name

Throws:
JAXRException - If the JAXR provider encounters an internal error

public String getMiddleName() throws JAXRException
Person

1

return

Throws JAXRException: JAXR
getMiddleName

String getMiddleName() throws JAXRException

Gets the middle name for this Person. Default is an empty String.

Capability Level: 1

Returns:
the person's middle name

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setMiddleName(String middleName) throws JAXRException
Person

1

middleName

Throws JAXRException: JAXR

setMiddleName

void setMiddleName(String middleName) throws JAXRException

Sets the middle name for this Person.

Capability Level: 1

Parameters:
middleName - the person's middle name
Throws: *JAXRException* - If the JAXR provider encounters an internal error

```java
public String getFullName() throws JAXRException
Person
```

```java
0
```

```java
return
Throws JAXRException: JAXR
```

**getFullName**

```java
String getFullName() throws JAXRException
```

Gets the fully formatted name for this person. Default is an empty `String`.

**Capability Level: 0**

**Returns:**
the person's full name

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

```java
public void setFullName(String fullName) throws JAXRException
Person
```

```java
0
```

```java
fullName
```
setFullName

```java
void setFullName(String fullName)
    throws JAXRException
```

Sets the fully formatted name for this person.

**Capability Level: 0**

**Parameters:**
- `fullName` - the person's full name

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
public class PhaseEvent

extends EventObject

Extends: java.util.EventObject

PhaseEvent represents the beginning or ending of processing for a particular phase of the request processing lifecycle, for the request encapsulated by the specified FacesContext.

See Also:
   Serialized Form

Field Summary

Fields inherited from class java.util.EventObject

source

Constructor Summary

PhaseEvent(FacesContext context, PhaseId phaseId, Lifecycle lifecycle)
Construct a new event object from the specified parameters.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getFacesContext()</code></td>
<td>Return the <code>FacesContext</code> for the request being processed.</td>
</tr>
<tr>
<td><code>getPhaseId()</code></td>
<td>Return the <code>PhaseId</code> representing the current request processing lifecycle phase.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.util.EventObject**

- `getSource`, `toString`

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`  

**Constructor Detail**

```java
public PhaseEvent(FacesContext context, PhaseId phasId, Lifecycle lifecycle)
```

- `Lifecycle`

  - `context`  
  - `phasId`  
  - `lifecycle`  

  **Throws**  
  - `NullPointerException`: `context` `phasId` `Lifecycle` `null`

**PhaseEvent**

```java
public PhaseEvent(FacesContext context,
```
Construct a new event object from the specified parameters. The specified Lifecycle will be the source of this event.

**Parameters:**
- context - FacesContext for the current request
- phaseId - Identifier of the current request processing lifecycle phase
- lifecycle - Lifecycle instance

**Throws:**
- NullPointerException - if context OR phaseId OR Lifecycle is null

### Method Detail

**public FacesContext getFacesContext()**

Return the FacesContext for the request being processed.

**public PhaseId getPhaseId()**

**public PhaseId getPhaseId()**
Return the `PhaseId` representing the current request processing lifecycle phase.
javax.faces.event Class PhaseId

java.lang.Object
   javax.faces.event.PhaseId

All Implemented Interfaces:
   Comparable

public class PhaseId
   extends Object
   implements Comparable

Implements: Comparable<T>

FacesEvent getPhaseId()

Typesafe enumeration of the legal values that may be returned by the getPhaseId() method of the FacesEvent interface.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static PhaseId ANY_PHASE</td>
</tr>
<tr>
<td>static PhaseId APPLY_REQUEST_VALUES</td>
</tr>
<tr>
<td>static PhaseId INVOKE_APPLICATION</td>
</tr>
<tr>
<td>static PhaseId</td>
</tr>
<tr>
<td>static PhaseId</td>
</tr>
<tr>
<td>static PhaseId</td>
</tr>
<tr>
<td>static PhaseId</td>
</tr>
<tr>
<td>static List</td>
</tr>
</tbody>
</table>

**Method Summary**

| int compareTo(Object other) | Compare this PhaseId instance to the specified one. |
| int getOrdinal() | Return the ordinal value of this PhaseId instance. |
| String toString() | Return a String representation of this PhaseId instance. |

**Methods inherited from class java.lang.Object**

close, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Field Detail**
ANY_PHASE

`public static final PhaseId ANY_PHASE`

Identifier that indicates an interest in events, no matter which request processing phase is being performed.

---

RESTORE_VIEW

`public static final PhaseId RESTORE_VIEW`

Identifier that indicates an interest in events queued for the Restore View phase of the request processing lifecycle.

---

APPLY_REQUEST_VALUES

`public static final PhaseId APPLY_REQUEST_VALUES`

Identifier that indicates an interest in events queued for the Apply Request Values phase of the request processing lifecycle.

---

PROCESS_VALIDATIONS

`public static final PhaseId PROCESS_VALIDATIONS`

Identifier that indicates an interest in events queued for the Process Validations phase of the request processing lifecycle.
**UPDATE_MODEL_VALUES**

```java
public static final PhaseId UPDATE_MODEL_VALUES
```

Identifier that indicates an interest in events queued for the *Update Model Values* phase of the request processing lifecycle.

---

**INVOKE_APPLICATION**

```java
public static final PhaseId INVOKE_APPLICATION
```

Identifier that indicates an interest in events queued for the *Invoke Application* phase of the request processing lifecycle.

---

**RENDER_RESPONSE**

```java
public static final PhaseId RENDER_RESPONSE
```

Identifier for the *Render Response* phase of the request processing lifecycle.

---

**VALUES**

```java
public static final List<PhaseId> VALUES
```

List of valid *PhaseId* instances, in ascending order of their ordinal value.
public int compareTo(Object other)

    PhaseId 0

    other

compareTo

public int compareTo(Object other)

    Compare this PhaseId instance to the specified one. Returns a negative integer, zero, or a positive integer if this object is less than, equal to, or greater than the specified object.

    Specified by: compareTo in interface Comparable

    Parameters:
    other - The other object to be compared to

public int getOrdinal()

    PhaseId

getOrdinal

public int getOrdinal()

    Return the ordinal value of this PhaseId instance.

gpublic String toString()
toString

public String toString()

Return a String representation of this PhaseId instance.

Overrides:
toString in class Object
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.faces.event Interface PhaseListener

All Superinterfaces: 
EventListerner, Serializable

public interface PhaseListener
extends EventListener, Serializable

Implements: java.util.EventListerner, java.io.Serializable

An interface implemented by objects that wish to be notified at the beginning and ending of processing for each standard phase of the request processing lifecycle.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void afterPhase(PhaseEvent event)</td>
</tr>
<tr>
<td>void beforePhase(PhaseEvent event)</td>
</tr>
<tr>
<td>PhaseId getPhaseId()</td>
</tr>
</tbody>
</table>

| Method Detail |
public void afterPhase(PhaseEvent event)

""

afterPhase

void afterPhase(PhaseEvent event)

Handle a notification that the processing for a particular phase has just been completed.

public void beforePhase(PhaseEvent event)

""

beforePhase

void beforePhase(PhaseEvent event)

Handle a notification that the processing for a particular phase of the request processing lifecycle is about to begin.

public PhaseId getPhaseId()

PhaseEvent PhaseId
PhaseId.ANY_PHASE

getPhaseId
PhaseId getPhaseId()

Return the identifier of the request processing phase during which this listener is interested in processing PhaseEvent events. Legal values are the singleton instances defined by the PhaseId class, including PhaseId.ANY_PHASE to indicate an interest in being notified for all standard phases.
javax.security.jacc  Interface PolicyConfiguration

public interface PolicyConfiguration

Policy PolicyConfiguration Policy
"open" "inService" "deleted"

"open" PolicyConfiguration "open" Policy "inService" commit "open" "inService"

"inService" Policy refresh refresh "inService" refresh "inService"
PolicyConfigurationFactory getPolicyConfiguration "open"

"deleted" Provider PolicyConfigurationFactory
getPolicyConfiguration "deleted"
getPolicyConfiguration "deleted" "open"

PolicyConfiguration 3 PolicyConfiguration

<table>
<thead>
<tr>
<th>Method</th>
<th>deleted</th>
<th>open</th>
<th>inService</th>
</tr>
</thead>
<tbody>
<tr>
<td>addToExcludedPolicy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>addRole</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>addUncheckedPolicy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>commit</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>delete</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>getContextID</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>inService</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>linkConfiguration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>removeExcludedPolicy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>removeRole</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>removeUncheckedPolicy</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

refresh PolicyConfiguration PolicyConfigurationFactory
getPolicyConfiguration inService
The methods of this interface are used by containers to create policy statements in a Policy provider. An object that implements the PolicyConfiguration interface provides the policy statement configuration interface for a corresponding policy context within the corresponding Policy provider.

The life cycle of a policy context is defined by three states; "open", "inService", and "deleted". A policy context is in one of these three states.

A policy context in the "open" state is in the process of being configured, and may be operated on by any of the methods of the PolicyConfiguration interface. A policy context in the "open" state must not be assimilated at Policy.refresh into the policy statements used by the Policy provider in performing its access decisions. In order for the policy statements of a policy context to be assimilated by the associated provider, the policy context must be in the "inService" state. A policy context in the "open" state is transitioned to the "inService" state by calling the commit method.

A policy context in the "inService" state is available for assimilation into the policy statements being used to perform access decisions by the associated Policy provider. Providers assimilate policy contexts containing policy statements when the refresh method of the provider is called. When a provider's refresh method is called, it must assimilate only those policy contexts whose state is "inService" and it must ensure that the policy statements put into service for each policy context are only those defined in the context at the time of the call to refresh. A policy context in the "inService" state is not available for additional configuration and may be returned to the "open" state by calling the getPolicyConfiguration method of the PolicyConfigurationFactory.

A policy context in the "deleted" state is neither available for configuration, nor is it available for assimilation into the Provider. A policy
context whose state is "deleted" may be reclaimed for subsequent processing by calling the getPolicyConfiguration method of the associated PolicyConfigurationFactory. A "deleted" policy context is transitioned to the "open" state when it it returned as a result of a call to getPolicyConfiguration.

The following table captures the correspondence between the policy context life cycle and the methods of the PolicyConfiguration interface. The rightmost 3 columns of the table correspond to the PolicyConfiguration state identified at the head of the column. The values in the cells of these columns indicate the next state resulting from a call to the method identified in the leftmost column of the corresponding row, or that calling the method is unsupported in the state represented by the column (in which case the state will remain unchanged).

<table>
<thead>
<tr>
<th>Method</th>
<th>Current State to Next State</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>deleted</td>
</tr>
<tr>
<td>addToExcludedPolicy</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>addToRole</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>addToUncheckedPolicy</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>commit</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>delete</td>
<td>deleted</td>
</tr>
<tr>
<td>getExcludedPolicy</td>
<td>deleted</td>
</tr>
<tr>
<td>inService</td>
<td>deleted</td>
</tr>
<tr>
<td>linkConfiguration</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>removeExcludedPolicy</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>removeRole</td>
<td>Unsupported Operation</td>
</tr>
<tr>
<td>removeUncheckedPolicy</td>
<td>Unsupported Operation</td>
</tr>
</tbody>
</table>

For a provider implementation to be compatible with multi-threaded environments, it may be necessary to synchronize the refresh method of the provider with the methods of its PolicyConfiguration interface and with the getPolicyConfiguration and inService methods of its PolicyConfigurationFactory.

**Author:**
Ron Monzillo, Gary Ellison

**See Also:**
[Permission], [PermissionCollection], [PolicyContextException], [PolicyConfigurationFactory]
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addToExcludedPolicy(Permission permission)</code></td>
<td>Used to add a single excluded policy statement to this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void addToExcludedPolicy(PermissionCollection permissions)</code></td>
<td>Used to add excluded policy statements to this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void addToRole(String roleName, Permission permission)</code></td>
<td>Used to add a single permission to a named role in this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void addToRole(String roleName, PermissionCollection permissions)</code></td>
<td>Used to add permissions to a named role in this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void addToUncheckedPolicy(Permission permission)</code></td>
<td>Used to add a single unchecked policy statement to this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void addToUncheckedPolicy(PermissionCollection permissions)</code></td>
<td>Used to add unchecked policy statements to this PolicyConfiguration.</td>
</tr>
<tr>
<td><code>void commit()</code></td>
<td>This method is used to set to &quot;inService&quot; the state of the policy context whose interface is this PolicyConfiguration Object.</td>
</tr>
<tr>
<td><code>void delete()</code></td>
<td>Causes all policy statements to be deleted from this PolicyConfiguration and sets its internal state such that calling any method, other than delete, getContextID, or inService on the PolicyConfiguration will be rejected and cause an UnsupportedOperationException to be thrown.</td>
</tr>
<tr>
<td><code>String getContextID()</code></td>
<td>This method returns this object's policy context identifier.</td>
</tr>
<tr>
<td><code>boolean inService()</code></td>
<td>This method is used to determine if the policy context whose interface is this PolicyConfiguration Object is in the &quot;inService&quot; state.</td>
</tr>
<tr>
<td><code>void linkConfiguration(PolicyConfiguration link)</code></td>
<td></td>
</tr>
</tbody>
</table>
void Creates a relationship between this configuration and another such that they share the same principal-to-role mappings.

void removeExcludedPolicy()
Used to remove any excluded policy statements from this PolicyConfiguration.

void removeRole(String roleName)
Used to remove a role and all its permissions from this PolicyConfiguration.

void removeUncheckedPolicy()
Used to remove any unchecked policy statements from this PolicyConfiguration.

Method Detail

public String getContextID() throws PolicyContextException

    return

    Throws
    SecurityException: "setPolicy" SecurityPermission
    AccessControlContext

    Throws
    PolicyContextException: getContextID

getContextID

String getContextID()
    throws PolicyContextException

This method returns this object's policy context identifier.

Returns:
    this object's policy context identifier.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.

PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the getBuildContextID method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

```java
public void addToRole(String roleName, java.security.PermissionCollection permissions) throws PolicyContextException
PolicyConfiguration Role PolicyConfiguration

Policy ""

roleName
permissions CollectionCollection Collection Collection

Throws SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws UnsupportedOperationException: PolicyConfiguration
"deleted" "inService"

Throws PolicyContextException: addToRole
PolicyContextException
```

addToRole

```java
void addToRole(String roleName,
PermissionCollection permissions)
throws PolicyContextException

Used to add permissions to a named role in this PolicyConfiguration. If the named Role does not exist in the PolicyConfiguration, it is created as a result of the call to this function.
```
It is the job of the Policy provider to ensure that all the permissions added to a role are granted to principals "mapped to the role".

**Parameters:**
- **roleName** - the name of the Role to which the permissions are to be added.
- **permissions** - the collection of permissions to be added to the role. The collection may be either a homogenous or heterogenous collection.

**Throws:**
- **SecurityException** - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- **UnsupportedOperationException** - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
- **PolicyConfigurationException** - if the implementation throws a checked exception that has not been accounted for by the addToRole method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyConfigurationException.

```java
public void addToRole(String roleName, java.security.Permission permission) throws PolicyConfigurationException
    PolicyConfiguration Role PolicyConfiguration
```

Policy ""

- **roleName**
- **permission**

**Throws**
- SecurityException: "setPolicy" SecurityPermission
- AccessControlContext
- UnsupportedOperationException: PolicyConfigurationException
  - "deleted" "inService"
Throws  

PolicyContextException: addToRole

```java
public void addToUncheckedPolicy(java.security.PermissionCollection
```

**addToRole**

```java
void addToRole(String roleName, Permission permission)
```

Throws  

SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.

UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.

PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the addToRole method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

Parameters:

- **roleName** - the name of the Role to which the permission is to be added.

- **permission** - the permission to be added to the role.
permissions) throws PolicyConfigurationException
PolicyConfiguration

permissions Collection Collection Collection Collection
Throws SecurityException: "setPolicy" SecurityPermission
AccessControlContext
Throws UnsupportedOperationException: PolicyConfigurationException
"deleted" "inService"
Throws PolicyConfigurationException: addToUncheckedPolicy
PolicyConfigurationException

addToUncheckedPolicy

void addToUncheckedPolicy(PermissionCollection permissions)
throws PolicyConfigurationException

Used to add unchecked policy statements to this PolicyConfiguration.

Parameters:
permissions - the collection of permissions to be added as unchecked policy statements. The collection may be either a homogenous or heterogenous collection.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
PolicyConfigurationException - if the implementation throws a checked exception that has not been accounted for by the addToUncheckedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyConfigurationException.
public void addToUncheckedPolicy(java.security.Permission permission) throws PolicyContextException
PolicyConfiguration

permission

Throws SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws UnsupportedOperationException: PolicyConfiguration
"deleted" "inService"

Throws PolicyContextException: addToUncheckedPolicy
PolicyContextException

addToUncheckedPolicy

void addToUncheckedPolicy(Permission permission)
throws PolicyContextException

Used to add a single unchecked policy statement to this PolicyConfiguration.

Parameters:
        permission - the permission to be added to the unchecked policy statements.

Throws:
        SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
        UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
        PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the addToUncheckedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.
public void addToExcludedPolicy(java.security.PermissionCollection permissions) throws PolicyContextException
PolicyConfiguration

_permissions CollectionCollection Collection Collection

Throws
SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws
UnsupportedOperationException: PolicyConfiguration
"deleted" "inService"

Throws
PolicyContextException: addToExcludedPolicy
PolicyContextException

addToExcludedPolicy

void addToExcludedPolicy(PermissionCollection permissions)
throws PolicyContextException

Used to add excluded policy statements to this PolicyConfiguration.

Parameters:
permissions - the collection of permissions to be added to the excluded policy statements. The collection may be either a homogenous or heterogenous collection.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the addToExcludedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.
public void addToExcludedPolicy(java.security.Permission permission) throws PolicyContextException
PolicyConfiguration

permission

Throws
SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws
UnsupportedOperationException: PolicyConfiguration
"deleted" "inService"

Throws
PolicyContextException: addToExcludedPolicy
PolicyContextException

addToExcludedPolicy

void addToExcludedPolicy(Permission permission)
throws PolicyContextException

Used to add a single excluded policy statement to this PolicyConfiguration.

Parameters:
permission - the permission to be added to the excluded policy statements.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the addToExcludedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.
public void removeRole(String roleName) throws PolicyConfigurationException
PolicyConfiguration

removeRole
void removeRole(String roleName)
throws PolicyConfigurationException

Used to remove a role and all its permissions from this PolicyConfiguration.

Parameters:
roleName - the name of the Role to remove from this PolicyConfiguration.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
PolicyConfigurationException - if the implementation throws a checked exception that has not been accounted for by the removeRole method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyConfigurationException.
public void removeUncheckedPolicy() throws PolicyConfigurationException

PolicyConfiguration

Throws SecurityException: "setPolicy" SecurityPermission AccessControlContext

Throws UnsupportedOperationException: PolicyConfiguration "deleted" "inService"

Throws PolicyConfigurationException: removeUncheckedPolicy

PolicyConfigurationException

removeUncheckedPolicy

void removeUncheckedPolicy() throws PolicyConfigurationException

Used to remove any unchecked policy statements from this PolicyConfiguration.

Throws:

SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.

UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.

PolicyConfigurationException - if the implementation throws a checked exception that has not been accounted for by the removeUncheckedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyConfigurationException.

public void removeExcludedPolicy() throws PolicyConfigurationException

PolicyConfiguration

Throws SecurityException: "setPolicy" SecurityPermission AccessControlContext

Throws UnsupportedOperation exception: PolicyConfiguration
removeExcludedPolicy

void removeExcludedPolicy() throws PolicyContextException

Used to remove any excluded policy statements from this PolicyConfiguration.

Throws:
- SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.
- PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the removeExcludedPolicy method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

public void linkConfiguration(PolicyConfiguration link) throws PolicyContextException

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration Collection

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration

PolicyConfiguration PolicyConfiguration
linkConfiguration

void linkConfiguration(PolicyConfiguration link)
throws PolicyContextException

Creates a relationship between this configuration and another such that they share the same principal-to-role mappings. PolicyConfigurations are linked to apply a common principal-to-role mapping to multiple separately manageable PolicyConfigurations, as is required when an application is composed of multiple modules.

Note that the policy statements which comprise a role, or comprise the excluded or unchecked policy collections in a PolicyConfiguration are unaffected by the configuration being linked to another.

Parameters:
link - a reference to a different PolicyConfiguration than this PolicyConfiguration.

The relationship formed by this method is symetric, transitive and idempotent. If the argument PolicyConfiguration does not have a different Policy context identifier than this PolicyConfiguration no relationship is formed, and an exception, as described below, is thrown.

Throws:
SecurityException - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
UnsupportedOperationException - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" or "inService" when this method is called.

IllegalArgumentException - if called with an argument PolicyConfiguration whose Policy context is equivalent to that of this PolicyConfiguration.

PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the linkConfiguration method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

```java
public void delete() throws PolicyContextException
PolicyConfiguration PolicyConfiguration
deletegetContextID inService
UnsupportedOperationException
```

Throws

SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws PolicyContextException: delete PolicyContextException

delete

void delete()
throws PolicyContextException

Causes all policy statements to be deleted from this PolicyConfiguration and sets its internal state such that calling any method, other than delete, getContextID, or inService on the PolicyConfiguration will be rejected and cause an UnsupportedOperationException to be thrown.

This operation has no affect on any linked PolicyConfigurations other
than removing any links involving the deleted PolicyConfiguration.

**Throws:**
- **SecurityException** - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- **PolicyContextException** - if the implementation throws a checked exception that has not been accounted for by the delete method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

```java
public void commit() throws PolicyContextException
PolicyConfiguration "inService" "inService"
Policy.refresh PolicyConfiguration
getPolicyConfiguration "inService" "open"

"inService" PolicyConfiguration commit
deletegetContextID inService
UnsupportedOperationException

Throws
- SecurityException: "setPolicy" SecurityPermission
  AccessControlContext
- UnsupportedOperationException: PolicyConfiguration
  "deleted"
- PolicyContextException: commit
  PolicyContextException
```

`commit`

```java
void commit()
    throws PolicyContextException
```

This method is used to set to "inService" the state of the policy context whose interface is this PolicyConfiguration Object. Only those policy contexts whose state is "inService" will be included in
the policy contexts processed by the Policy.refresh method. A policy context whose state is "inService" may be returned to the "open" state by calling the getPolicyConfiguration method of the PolicyConfiguration factory with the policy context identifier of the policy context.

When the state of a policy context is "inService", calling any method other than commit, delete, getApplicationContext, or inService on its PolicyConfiguration Object will cause an UnsupportedOperationException to be thrown.

Throws:

- **SecurityException** - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- **UnsupportedOperationException** - if the state of the policy context whose interface is this PolicyConfiguration Object is "deleted" when this method is called.
- **PolicyContextException** - if the implementation throws a checked exception that has not been accounted for by the commit method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

```java
public boolean inService() throws PolicyContextException

PolicyConfiguration "inService"

return "inService" true false

Throws SecurityException: "setPolicy" SecurityPermission
AccessControlContext

Throws PolicyContextException: inService
PolicyContextException
```

This method is used to determine if the policy context whose
interface is this PolicyConfiguration Object is in the "inService" state.

**Returns:**
true if the state of the associated policy context is "inService";
false otherwise.

**Throws:**
- `SecurityException` - if called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- `PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the inService method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.
public abstract class PolicyConfigurationFactory extends Object

finder PolicyConfigurationFactory PolicyConfiguration Policy Provider

See also java.security.Permission, javax.security.jacc.PolicyConfiguration, javax.security.jacc.PolicyConfigurationException

Abstract factory and finder class for obtaining the instance of the class that implements the PolicyConfigurationFactory of a provider. The factory will be used to instantiate PolicyConfiguration objects that will be used by the deployment tools of the container to create and manage policy contexts within the Policy Provider.

Implementation classes must have a public no argument constructor that may be used to create an operational instance of the factory implementation class.

Author:
Ron Monzillo, Gary Ellison, Harpreet Singh

See Also:
Permission, PolicyConfiguration, PolicyConfigurationException
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getPolicyConfiguration</strong></td>
<td>This method is used to obtain an instance of the provider specific class that implements the PolicyConfiguration interface that corresponds to the identified policy context within the provider.</td>
</tr>
<tr>
<td><strong>getPolicyConfigurationFactory</strong></td>
<td>This static method uses a system property to find and instantiate (via a public constructor) a provider specific factory implementation class.</td>
</tr>
<tr>
<td><strong>inService</strong></td>
<td>This method determines if the identified policy context exists with state &quot;inService&quot; in the Policy provider associated with the factory.</td>
</tr>
</tbody>
</table>

#### Methods inherited from class java.lang.**Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

---

### Constructor Detail

**public PolicyConfigurationFactory()**

**PolicyConfigurationFactory**

**public PolicyConfigurationFactory()**
public static PolicyConfigurationFactory getPolicyConfigurationFactory() throws ClassNotFoundException, PolicyContextException

javax.security.jacc.PolicyConfigurationFactory.provider

return PolicyConfigurationFactory

Throws: SecurityException: "setPolicy" SecurityPermission

AccessControlContext

Throws: ClassNotFoundException:

PolicyContextException: getPolicyConfigurationFactory
PolicyContextException

getPolicyConfigurationFactory

public static PolicyConfigurationFactory getPolicyConfigurationFactory() throws ClassNotFoundException, PolicyContextException

This static method uses a system property to find and instantiate (via a public constructor) a provider specific factory implementation class. The name of the provider specific factory implementation class is obtained from the value of the system property,

javax.security.jacc.PolicyConfigurationFactory.provider.

Returns: the singleton instance of the provider specific PolicyConfigurationFactory implementation class.

Throws: SecurityException - when called by an AccessControlContext
that has not been granted the "setPolicy" SecurityPermission.  

`ClassNotFoundException` - when the class named by the system property could not be found including because the value of the system property has not been set. 

`PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the getPolicyConfigurationFactory method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException

```
abstract public PolicyConfiguration getPolicyConfiguration(String contextID, boolean remove) 
throws PolicyContextException
PolicyConfiguration PolicyConfiguration

PolicyConfiguration "deleted"
"inService" "open"  PolicyConfiguration

PolicyConfiguration PolicyConfiguration

PolicyConfiguration

contextID PolicyConfiguration null
remove boolean PolicyConfiguration true false
return Policy PolicyConfiguration
Throws SecurityException: "setPolicy" SecurityPermission
AccessControlContext
Throws PolicyContextException: getPolicyConfiguration
PolicyContextException

g etPolicyConfiguration
```
This method is used to obtain an instance of the provider specific class that implements the PolicyConfiguration interface that corresponds to the identified policy context within the provider. The methods of the PolicyConfiguration interface are used to define the policy statements of the identified policy context.

If at the time of the call, the identified policy context does not exist in the provider, then the policy context will be created in the provider and the Object that implements the context's PolicyConfiguration Interface will be returned. If the state of the identified context is "deleted" or "inService" it will be transitioned to the "open" state as a result of the call. The states in the lifecycle of a policy context are defined by the PolicyConfiguration interface.

For a given value of policy context identifier, this method must always return the same instance of PolicyConfiguration and there must be at most one actual instance of a PolicyConfiguration with a given policy context identifier (during a process context).

To preserve the invariant that there be at most one PolicyConfiguration object for a given policy context, it may be necessary for this method to be thread safe.

**Parameters:**

- contextID - A String identifying the policy context whose PolicyConfiguration interface is to be returned. The value passed to this parameter must not be null.

- remove - A boolean value that establishes whether or not the policy statements of an existing policy context are to be removed before its PolicyConfiguration object is returned. If the value passed to this parameter is true, the policy statements of an existing policy context will be removed. If the value is false, they will not be removed.

**Returns:**

an Object that implements the PolicyConfiguration Interface
matched to the Policy provider and corresponding to the identified policy context.

**Throws:**

- [*SecurityException*](#) - when called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.
- [*PolicyContextException*](#) - if the implementation throws a checked exception that has not been accounted for by the getPolicyConfiguration method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.

```java
abstract public boolean inService(String contextID) throws PolicyContextException
Policy "inService"

  contextID
  return "inService" true false
  Throws SecurityException: "setPolicy" SecurityPermission
          AccessControlContext
  Throws PolicyContextException: inService
          PolicyContextException

inService

public abstract boolean inService(String contextID)
  throws PolicyContextException

This method determines if the identified policy context exists with state "inService" in the Policy provider associated with the factory.

**Parameters:**

- contextID - A string identifying a policy context

**Returns:**

true if the identified policy context exists within the provider and its state is "inService", false otherwise.
Throws:

`SecurityException` - when called by an AccessControlContext that has not been granted the "setPolicy" SecurityPermission.

`PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the inService method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException.
This utility class is used by containers to communicate policy context identifiers and other policy relevant context to Policy providers. Policy providers use the policy context identifier to select the subset of policy to apply in access decisions.

The value of a policy context identifier is a String and each thread has an independently established policy context identifier. A container will establish the thread-scoped value of a policy context identifier by calling the static setContextID method. The value of a thread-scoped policy
context identifier is available (to Policy) by calling the static `getContextID` method.

This class is also used by Policy providers to request additional thread-scoped policy relevant context objects from the calling container. Containers register container-specific `PolicyContext` handlers using the static `registerHandler` method. Handler registration is scoped to the class, such that the same handler registrations are active in all thread contexts. Containers may use the static method `setHandlerData` to establish a thread-scoped parameter that will be passed to handlers when they are activated by Policy providers. The static `getContext` method is used to activate a handler and obtain the corresponding context object.

The static accessor functions provided by this class allow per-thread policy context values to be established and communicated independent of a common reference to a particular `PolicyContext` instance.

The `PolicyContext` class may encapsulate static `ThreadLocal` instance variables to represent the policy context identifier and handler data values.

The Application server must bundle or install the `PolicyContext` class, and the containers of the application server must prevent the methods of the `PolicyContext` class from being called from calling contexts that are not authorized to call these methods. With the exception of the `getContextID` and `GetHandlerKeys` methods, containers must restrict and afford access to the methods of the `PolicyContext` class to calling contexts trusted by the container to perform container access decisions. The `PolicyContext` class may satisfy this requirement (on behalf of its container) by rejecting calls made from an `AccessControlContext` that has not been granted the "setPolicy" SecurityPermission, and by ensuring that Policy providers used to perform container access decisions are granted the "setPolicy" permission.

**Author:**
Ron Monzillo, Gary Ellison

**See Also:**
`PolicyContextHandler`
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static</strong> <code>Object getContext(String key)</code></td>
<td>This method may be used by a Policy provider to activate the PolicyContextHandler registered to the context object key and cause it to return the corresponding policy context object from the container.</td>
</tr>
<tr>
<td><strong>static</strong> <code>String getContextID()</code></td>
<td>This static method returns the value of the policy context identifier associated with the thread on which the accessor is called.</td>
</tr>
<tr>
<td><strong>static</strong> <code>Set getHandlerKeys()</code></td>
<td>This method may be used to obtain the keys that identify the container specific context handlers registered by the container.</td>
</tr>
<tr>
<td><strong>static</strong> <code>void registerHandler(String key, PolicyContextHandler handler, boolean replace)</code></td>
<td>Authorization protected method used to register a container specific PolicyContext handler.</td>
</tr>
<tr>
<td><strong>static</strong> <code>void setContextID(String contextID)</code></td>
<td>Authorization protected method used to modify the value of the policy context identifier associated with the thread on which this method is called.</td>
</tr>
<tr>
<td><strong>static</strong> <code>void setHandlerData(Object data)</code></td>
<td>Authorization protected method that may be used to associate a thread-scoped handler data object with the PolicyContext.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public static void setContextID(String contextID)

contextID String PolicyContext null

Throws SecurityException: AccessControlContext

setContextID

Authorization protected method used to modify the value of the policy context identifier associated with the thread on which this method is called.

Parameters:
contextID - a String that represents the value of the policy context identifier to be assigned to the PolicyContext for the calling thread. The value null is a legitimate value for this parameter.

Throws:
SecurityException - if the calling AccessControlContext is not authorized by the container to call this method.

public static String getContextID()

return String null setContext null

Throws SecurityException: AccessControlContext

getContextID

This static method returns the value of the policy context identifier associated with the thread on which the accessor is called.
Returns:
The String (or null) policy context identifier established for the thread. This method must return the default policy context identifier, null, if the policy context identifier of the thread has not been set via setContext to another value.

Throws:
SecurityException - if the calling AccessControlContext is not authorized by the container to call this method. Containers may choose to authorize calls to this method by any AccessControlContext.

```java
public static void setHandlerData(Object data)
PolicyContext
data Policy null
```

Throws  SecurityException:  AccessControlContext

setHandlerData

Authorization protected method that may be used to associate a thread-scoped handler data object with the PolicyContext. The handler data object will be made available to handlers, where it can serve to supply or bind the handler to invocation scoped state within the container.

Parameters:

data - a container-specific object that will be associated with the calling thread and passed to any handler activated by a Policy provider (on the thread). The value null is a legitimate value for this parameter, and is the value that will be used in the activation of handlers if the setHandlerData has not been called on the thread.

Throws:
SecurityException - if the calling AccessControlContext is not authorized by the container to call this method.
public static void registerHandler(String key, PolicyContextHandler handler, boolean replace) throws PolicyContextException

Authorization protected method used to register a container specific PolicyContext handler. A handler may be registered to handle multiple keys, but at any time, at most one handler may be registered for a key.

Parameters:
- key - a (case-sensitive) String that identifies the context object handled by the handler. The value of this parameter must not be null.
- handler - an object that implements the PolicyContextHandler interface. The value of this parameter must not be null.
- replace - this boolean value defines the behavior of this method if, when it is called, a PolicyContextHandler has already been registered to handle the same key. In that case, and if the value of this argument is true, the existing handler is replaced with the argument handler. If the value of this parameter is false the
existing registration is preserved and an exception is thrown.

**Throws:**
- `IllegalArgumentException` - if the value of either of the handler or key arguments is null, or the value of the replace argument is false and a handler with the same key as the argument handler is already registered.
- `SecurityException` - if the calling AccessControlContext is not authorized by the container to call this method.
- `PolicyContextException` - if an operation by this method on the argument PolicyContextHandler causes it to throw a checked exception that is not accounted for in the signature of this method.

---

### public static java.util.Set<E> getHandlerKeys()

**return**

```
Set E String PolicyContext
```

**Throws**

```
SecurityException: AccessControlContext
AccessControlContext
```

### getHandlerKeys

This method may be used to obtain the keys that identify the container specific context handlers registered by the container.

**Returns:**

A Set, the elements of which, are the String key values that identify the handlers that have been registered and therefore may be activated on the PolicyContext.

**Throws:**

```
SecurityException - if the calling AccessControlContext is not authorized by the container to call this method. Containers may choose to authorize calls to this method by any AccessControlContext.
```
public static Object getContext(String key) throws PolicyContextException

Policy PolicyContextHandler

key String PolicyContextHandler null

return null

Throws IllegalArgumentException: PolicyContextHandler

Throws SecurityException: AccessControlContext

Throws PolicyContextException: PolicyContextHandler

getContext

public static Object getContext(String key)

throws PolicyContextException

This method may be used by a Policy provider to activate the PolicyContextHandler registered to the context object key and cause it to return the corresponding policy context object from the container. When this method activates a handler, it passes to the handler the context object key and the handler data associated with the calling thread.

Parameters:

key - a String that identifies the PolicyContextHandler to activate and the context object to be acquired from the handler. The value of this parameter must not be null.

Returns:

the container and handler specific object containing the desired context. A null value is returned if the corresponding handler has been registered, and the value of the corresponding context is null.

Throws:

IllegalArgumentException - if a PolicyContextHandler has not been registered for the key or the registered handler no longer supports the key.

SecurityException - if the calling AccessControlContext is not authorized by the container to call this method.

PolicyContextException - if an operation by this method on the
identified PolicyContextHandler causes it to throw a checked exception that is not accounted for in the signature of this method.

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| FRAMES | NO FRAMES |
| DETAIL: FIELD | CONSTR | METHOD |

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.security.jacc  **Class PolicyContextException**

**java.lang.Object**
- **java.lang.Throwable**
- **java.lang.Exception**
- **javax.security.jacc.PolicyContextException**

**All Implemented Interfaces:**

**Serializable**

---

```java
public class PolicyContextException
extends Exception

Extends: Throwable > Exception
```

**javax.security.jacc.PolicyConfiguration**

**javax.security.jacc.PolicyConfigurationFactory**

**javax.security.jacc.PolicyContext**

**javax.security.jacc.PolicyContextException**

**javax.security.jacc**

---


This checked exception is thrown by implementations of the
**javax.security.jacc.PolicyConfiguration Interface**, the
**javax.security.jacc.PolicyConfigurationFactory abstract class**, the
**javax.security.jacc.PolicyContext utility class**, and implementations of
the **javax.security.jacc.PolicyContextException Interface**.

This exception is used by **javax.security.jacc** implementation classes to
rethrow checked exceptions occurring within an implementation that are
not declared by the interface or class being implemented.
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PolicyContextException()</code></td>
<td>Constructs a new PolicyContextException with <code>null</code> as its detail message.</td>
</tr>
<tr>
<td><code>PolicyContextException(String msg)</code></td>
<td>Constructs a new PolicyContextException with the specified detail message.</td>
</tr>
<tr>
<td><code>PolicyContextException(String msg, Throwable cause)</code></td>
<td>Constructs a new PolicyContextException with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>PolicyContextException(Throwable cause)</code></td>
<td>Constructs a new PolicyContextException with the specified cause.</td>
</tr>
</tbody>
</table>

## Method Summary

**Methods inherited from class java.lang.Throwable**
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

**Methods inherited from class java.lang.Object**
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public PolicyContextException()
PolicyContextException
null

PolicyContextException

public PolicyContextException()

Constructs a new PolicyContextException with null as its detail message, describing the cause of the exception.

public PolicyContextException(String msg)
PolicyContextException

    msg
String

PolicyContextException

public PolicyContextException(String msg)

Constructs a new PolicyContextException with the specified detail message.

Parameters:
    msg -- a String containing a detail message describing the cause of the exception.

public PolicyContextException(String msg, Throwable cause)
PolicyContextException

    msg
String
    cause  "" Throwable null

PolicyContextException
public PolicyContextException(String msg, Throwable cause)

Constructs a new PolicyContextException with the specified detail message and cause. The cause will be encapsulated in the constructed exception.

Parameters:
- msg -- a String containing a detail message describing the cause of the exception.
- cause -- the Throwable that is "causing" this exception to be constructed. A null value is permitted, and the value passed through this parameter may subsequently be retrieved by calling getCause() on the constructed exception.

public PolicyContextException(Throwable cause)

cause PolicyContextException

cause "" Throwable null
getcause

PolicyContextException

public PolicyContextException(Throwable cause)

Constructs a new PolicyContextException with the specified cause. The cause will be encapsulated in the constructed exception.

Parameters:
- cause -- the Throwable that is "causing" this exception to be constructed. A null value is permitted, and the value passed through this parameter may subsequently be retrieved by calling getCause() on the constructed exception.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public interface PolicyContextHandler

This interface defines the methods that must be implemented by handlers that are to be registered and activated by the PolicyContext class. The PolicyContext class provides methods for containers to register and activate container-specific PolicyContext handlers. Policy providers use the PolicyContext class to activate handlers to obtain (from the container) additional policy relevant context to apply in their access decisions. All handlers registered and activated via the PolicyContext class must implement the PolicyContextHandler interface.

Author:
Ron Monzillo, Gary Ellison

See Also:
PolicyContext, PolicyContextException

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td><strong>getContext</strong>(String key, Object data)</td>
</tr>
<tr>
<td>This public method is used by the PolicyContext class to activate the handler and obtain from it the context object identified by the (case-sensitive) key.</td>
</tr>
<tr>
<td><strong>String[]</strong></td>
</tr>
<tr>
<td><strong>getKeys</strong>()</td>
</tr>
<tr>
<td>This public method returns the keys identifying the context objects supported by the handler.</td>
</tr>
<tr>
<td><strong>String</strong></td>
</tr>
<tr>
<td><strong>supports</strong>(String key)</td>
</tr>
</tbody>
</table>
This public method returns a boolean result indicating whether or not the handler supports the context object identified by the (case-sensitive) key value.

**Method Detail**

```java
public boolean supports(String key) throws PolicyContextException

**Parameters:**
key - a String value identifying a context object that could be supported by the handler. The value of this parameter must not be null.

**Returns:**
a boolean indicating whether or not the context object corresponding to the argument key is handled by the handler.

**Throws:**
PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown exception.
```
public String[] getKeys() throws PolicyContextException
null
    return String 0 null
Throws PolicyContextException: PolicyContextException

getKeys

String[] getKeys()
    throws PolicyContextException

This public method returns the keys identifying the context objects supported by the handler. The value of each key supported by a handler must be a non-null String value.

Returns:
    an array containing String values identifying the context objects supported by the handler. The array must not contain duplicate key values. In the unlikely case that the Handler supports no keys, the handler must return a zero length array. The value null must never be returned by this method.

Throws:
    PolicyContextException - if the implementation throws a checked exception that has not been accounted for by the method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown PolicyContextException

public Object getContext(String key, Object data) throws PolicyContextException

PolicyContext getRuleContext()

PolicyContext
PolicyContext
getContext

```java
Object getContext(String key,
                   Object data)
throws PolicyContextException
```

This public method is used by the PolicyContext class to activate the handler and obtain from it the context object identified by the (case-sensitive) key. In addition to the key, the handler will be activated with the handler data value associated within the PolicyContext class with the thread on which the call to this method is made.

Note that the policy context identifier associated with a thread is available to the handler by calling PolicyContext.getContextID().

**Parameters:**
- `key` - a String that identifies the context object to be returned by the handler. The value of this parameter must not be null.
- `data` - the handler data object associated with the thread on which the call to this method has been made. Note that the value passed through this parameter may be null.

**Returns:**
- The container and handler specific object containing the desired context. A null value may be returned if the value of the corresponding context is null.

**Throws:**
- `PolicyContextException` - if the implementation throws a checked exception that has not been accounted for by the method signature. The exception thrown by the implementation class will be encapsulated (during construction) in the thrown `PolicyContextException`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
<table>
<thead>
<tr>
<th>Summary</th>
<th>Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail</td>
<td>Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTRUCTOR | Method
FRAMES NO FRAMES
DETAIL: FIELD | CONSTRUCTOR | Method
The `PortInfo` interface is used by a `HandlerResolver` to query information about the port it is being asked to create a handler chain for. This interface is never implemented by an application, only by a JAX-WS implementation.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getBindingID()</code></td>
<td>Gets the URI identifying the binding used by the port being accessed.</td>
</tr>
<tr>
<td><code>QName getPortName()</code></td>
<td>Gets the qualified name of the WSDL port being accessed.</td>
</tr>
<tr>
<td><code>QName getServiceName()</code></td>
<td>Gets the qualified name of the WSDL service name containing the port being accessed.</td>
</tr>
</tbody>
</table>
public javax.xml.namespace.QName getServiceName()
WSDL
    return javax.xml.namespace.QName WSDL

getServiceName

QName getServiceName()

Gets the qualified name of the WSDL service name containing the port being accessed.

Returns:
    javax.xml.namespace.QName The qualified name of the WSDL service.

public javax.xml.namespace.QName getPortName()
WSDL
    return javax.xml.namespace.QName WSDL

getPortName

QName getPortName()

Gets the qualified name of the WSDL port being accessed.

Returns:
    javax.xml.namespace.QName The qualified name of the WSDL port.

public String getBindingID()
URI
    return String

See also
    javax.xml.ws.Binding
getBindingID

String getBindingID()

Gets the URI identifying the binding used by the port being accessed.

Returns:
   String The binding identifier for the port.

See Also:
   Binding
javax.ejb Annotation Type PostActivate

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PostActivate

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Bean

Designates a method to receive a callback after a stateful session bean has been activated.

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

PREV CLASS  NEXT CLASS  FRAMES  NO FRAMES
SUMMARY: REQUIRED | OPTIONAL  DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.registry.infomodel Interface PostalAddress

All Superinterfaces:
   ExtensibleObject

public interface PostalAddress
extends ExtensibleObject

Implements: ExtensibleObject

PostalAddress  Address

PostalAddress is a simple re-usable entity class that defines attributes of a postal Address.

Author:
   Farrukh S. Najmi

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getCity()</td>
<td>Returns the city.</td>
</tr>
<tr>
<td>String getCountry()</td>
<td>Returns the country.</td>
</tr>
<tr>
<td>String getPostalCode()</td>
<td>Returns the postal or zip code.</td>
</tr>
<tr>
<td>ClassificationScheme getPostalScheme()</td>
<td>Returns a user-defined postal scheme for codifying the attributes of PostalAddress.</td>
</tr>
<tr>
<td>String getStateOrProvince()</td>
<td>Returns the state or province.</td>
</tr>
<tr>
<td>String getStreet()</td>
<td>Returns the street name.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td><code>getStreetNumber()</code></td>
<td>Returns the street number.</td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Returns the type of address (for example, &quot;headquarters&quot;) as a String.</td>
</tr>
<tr>
<td><code>setCity(String city)</code></td>
<td>Sets the city.</td>
</tr>
<tr>
<td><code>setCountry(String country)</code></td>
<td>Sets the country.</td>
</tr>
<tr>
<td><code>setPostalCode(String postalCode)</code></td>
<td>Sets the postal or zip code.</td>
</tr>
<tr>
<td><code>setPostalScheme(ClassificationScheme scheme)</code></td>
<td>Sets a user-defined postal scheme for codifying the attributes of PostalAddress.</td>
</tr>
<tr>
<td><code>setStateOrProvince(String stateOrProvince)</code></td>
<td>Sets the state or province.</td>
</tr>
<tr>
<td><code>setStreet(String street)</code></td>
<td>Sets the street name.</td>
</tr>
<tr>
<td><code>setStreetNumber(String streetNumber)</code></td>
<td>Sets the street number.</td>
</tr>
<tr>
<td><code>setType(String type)</code></td>
<td>Sets the type of address (for example, &quot;headquarters&quot;) as a String.</td>
</tr>
<tr>
<td></td>
<td>Methods inherited from interface</td>
</tr>
<tr>
<td></td>
<td><code>javax.xml.registry.infomodel.ExtensibleObject</code></td>
</tr>
<tr>
<td></td>
<td><code>addSlot</code>, <code>addSlots</code>, <code>getSlot</code>, <code>getSlots</code>, <code>removeSlot</code>, <code>removeSlots</code></td>
</tr>
</tbody>
</table>

**Method Detail**

public String getStreet() throws `JAXRException`
getStreet

```java
String getStreet() throws JAXRException
```

Returns the street name. Default is an empty String.

**Capability Level:** 0

**Returns:**
- the street name

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

```java
public void setStreet(String street) throws JAXRException
```

Sets the street name.
public String getStreetNumber() throws JAXRException

    Returns the street number. Default is an empty String.

    Capability Level: 0

    Parameters:
            street - the street name
    Throws:
        JAXRException - If the JAXR provider encounters an internal error

public void setStreetNumber(String streetNumber) throws JAXRException


**setStreetNumber**

```java
void setStreetNumber(String streetNumber)
throws JAXRException
```

Sets the street number.

**Capability Level: 0**

**Parameters:**
- `streetNumber` - the street number

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public String getCity() throws JAXRException**

```java
String getCity()
throws JAXRException
```

Returns the city. Default is an empty String.
public void setCity(String city) throws JAXRException

Sets the city.

Parameters:  
city - the city

Throws:  
JAXRException - If the JAXR provider encounters an internal error

public String getStateOrProvince() throws JAXRException

0
return

Throws

JAXRException: JAXR

getStateOrProvince

String getStateOrProvince() throws JAXRException

Returns the state or province. Default is an empty String.

Capability Level: 0

Returns:
the state or province

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setStateOrProvince(String stateOrProvince) throws JAXRException

0

stateOrProvince

Throws

JAXRException: JAXR

setStateOrProvince

void setStateOrProvince(String stateOrProvince) throws JAXRException

Sets the state or province.

Capability Level: 0
Parameters:
  stateOrProvince - the state or province

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public String getPostalCode() throws JAXRException

0

    return US

Throws:
  JAXRException: JAXR

getPostalCode

String getPostalCode() throws JAXRException

    Returns the postal or zip code. Default is an empty String.

    Capability Level: 0

    Returns:
    the postal code (e.g. US zip code)

    Throws:
    JAXRException - If the JAXR provider encounters an internal error

public void setPostalCode(String postalCode) throws JAXRException

0
**postalCode**

**US**

**Throws**

**JAXRException**

JAXR

### setPostalCode

**void** setPostalCode(String postalCode)

*Throws* **JAXRException**

Sets the postal or zip code.

**Capability Level: 0**

**Parameters:**

*postalCode* - the postal code (e.g. US zip code)

**Throws:**

**JAXRException** - If the JAXR provider encounters an internal error

---

**public String getCountry() throws JAXRException**

**0**

**return**

**Throws** **JAXRException**

JAXR

### getCountry

**String** getCountry()

*throws* **JAXRException**

Returns the country. Default is an empty String.

**Capability Level: 0**

**Returns:**
public void setCountry(String country) throws JAXRException

Sets the country.

Capability Level: 0

Parameters:
  country - the country

Throws:  
  JAXRException - If the JAXR provider encounters an internal error

public String getType() throws JAXRException

String "headquarters"

return PostalAddress "Home" "Office"
Throws

JAXRException: JAXR

get Type

String getType()

throws JAXRException

Returns the type of address (for example, "headquarters") as a String.

Capability Level: 0

Returns:
the type for this PostalAddress. This is an arbitrary String (e.g. "Home", "Office")

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setType(String type) throws JAXRException

String "headquarters"

0

type PostalAddress "Home""Office"

Throws JAXRException: JAXR

set Type

void setType(String type)

throws JAXRException

Sets the type of address (for example, "headquarters") as a String.

Capability Level: 0
Parameters:
  type - the type for this PostalAddress. This is an arbitrary String (e.g. "Home", "Office")

Throws:
  JAXRException - If the JAXR provider encounters an internal error

---

public void setPostalScheme(ClassificationScheme scheme) throws JAXRException
PostalAddress

setPostalScheme

void setPostalScheme(ClassificationScheme scheme)
throws JAXRException

Sets a user-defined postal scheme for codifying the attributes of PostalAddress.

Capability Level: 0

Parameters:
  scheme - the user defined postal scheme.

Throws:
  JAXRException - If the JAXR provider encounters an internal error

---

public ClassificationScheme getPostalScheme() throws JAXRException
PostalAddress
RegistryService#getDefaultPostalScheme()

0

return

Throws

JAXRException: JAXR

See also
getDefaultPostalScheme()

getPostalScheme

ClassificationScheme getPostalScheme() throws JAXRException

Returns a user-defined postal scheme for codifying the attributes of PostalAddress. If none is defined for this object, then must return the default value returned by RegistryService# getDefaultPostalScheme()

Capability Level: 0

Returns:

the user defined postal scheme.

Throws:

JAXRException - If the JAXR provider encounters an internal error

See Also:

RegistryService.getDefaultPostalScheme()
The `PostConstruct` annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization. This method MUST be invoked before the class is put into service. This annotation MUST be supported on all classes that support dependency injection. The method annotated with `PostConstruct` MUST be invoked even if the class does not request any resources to be injected. Only one method can be annotated with this annotation. The method on which the `PostConstruct` annotation is applied MUST fulfill all of the following criteria:

- The method MUST NOT have any parameters except in the case of EJB interceptors in which case it takes an `InvocationContext` object as defined by the EJB specification.
- The return type of the method MUST be `void`.
- The method MUST NOT throw a checked exception.
- The method on which `PostConstruct` is applied MAY be public, protected, package private or private.
- The method MUST NOT be static except for the application client.
- The method MAY be final.
- If the method throws an unchecked exception the class MUST NOT be put
into service except in the case of EJBs where the EJB can handle exceptions and even recover from them.

Since:
1.0

See Also:
PreDestroy, Resource

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type PostLoad

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PostLoad

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since:** Java Persistence 1.0
javax.persistence Annotation Type PostPersist

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PostPersist

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since:**
Java Persistence 1.0

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

---

PS:
javax.persistence  **Annotation** Type **PostRemove**

```java
@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PostRemove
```

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

---

**since**  
Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since:**  
Java Persistence 1.0

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.persistence Annotation Type PostUpdate

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PostUpdate

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since:**
Java Persistence 1.0

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.annotation  Annotation Type PreDestroy

@Documented
@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface PreDestroy

Implements: Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=METHOD)

PreDestroy  PreDestroy  Java EE 5
PostConstruct  PreDestroy  EJB (interceptor)
EJB  InvocationContext  void PreDestroy  public
protectedpackage private  private static final EJB

since  1.0

See  javax.annotation.PostConstruct,
also  javax.annotation.Resource

The PreDestroy annotation is used on methods as a callback notification
to signal that the instance is in the process of being removed by the
container. The method annotated with PreDestroy is typically used to
release resources that it has been holding. This annotation MUST be
supported by all container managed objects that support PostConstruct
except the application client container in Java EE 5. The method on
which the PreDestroy annotation is applied MUST fulfill all of the
following criteria - - The method MUST NOT have any parameters except
in the case of EJB interceptors in which case it takes an
InvocationContext object as defined by the EJB specification. - The return
type of the method MUST be void. - The method MUST NOT throw a
checked exception. - The method on which PreDestroy is applied MAY
be public, protected, package private or private. - The method MUST
NOT be static. - The method MAY be final. - If the method throws an
unchecked exception it is ignored except in the case of EJBs where the
EJB can handle exceptions.
Since:
1.0
See Also:
PostConstruct, Resource
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTR | METHOD |
Class PreencodedMimeBodyPart

extends MimeBodyPart

Extends: BodyPart > MimeBodyPart

A MimeBodyPart that handles data that has already been encoded. This class is useful when constructing a message and attaching data that has already been encoded (for example, using base64 encoding). The data may have been encoded by the application, or may have been stored in a file or database in encoded form. The encoding is supplied when this object is created. The data is attached to this object in the usual fashion, by using the setText, setContent, or setDataHandler methods.

Since: JavaMail 1.4

Field Summary

Fields inherited from class javax.mail.internet.MimeBodyPart

**Constructor Summary**

<table>
<thead>
<tr>
<th>PreencodedMimeBodyPart(String encoding)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a PreencodedMimeBodyPart that assumes the data is encoded using the specified encoding.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>String getEncoding()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the content transfer encoding specified when this object was created.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>protected void updateHeaders()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Force the Content-Transfer-Encoding header to use the encoding that was specified when this object was created.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>void writeTo(OutputStream os)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output the body part as an RFC 822 format stream.</td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.mail.internet.MimeBodyPart**

| addHeader, addHeaderLine, attachFile, attachFile, addAllHeaderLines, getAllHeaders, getContent, getContentID, getContentType, getDataHandler, getDescription, getDisposition, getFileName, getHeader, getHeader, getInputStream, getLineCount, getMatchingHeaderLines, getMatchingHeaders, getRawInputStream, getSize, isMimeType, removeHeader, saveFile, saveFile, setContent, setContent, setContent, setContentLanguage, setContentLanguage, setContentMD5, setDataHandler, setDescription, setDescription, setText, setText |
Methods inherited from class javax.mail.BodyPart
getParent

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public PreencodedMimeBodyPart(String encoding)
PreencodedMimeBodyPart MIME Content-Transfer-Encoding

PreencodedMimeBodyPart

public PreencodedMimeBodyPart(String encoding)

Create a PreencodedMimeBodyPart that assumes the data is encoded using the specified encoding. The encoding must be a MIME supported Content-Transfer-Encoding.

Method Detail

public String getEncoding() throws MessagingException

getEncoding

public String getEncoding()
throws MessagingException
Returns the content transfer encoding specified when this object was created.

**Specified by:**
`getEncoding` in interface `MimePart`

**Overrides:**
`getEncoding` in class `MimeBodyPart`

**Returns:**
content-transfer-encoding

**Throws:**
`MessagingException`

**See Also:**
`MimeBodyPart.headers`

---

```java
public void writeTo(java.io.OutputStream os) throws java.io.IOException, MessagingException
RFC 822
```

**Throws**
`MessagingException`

**Throws**
`java.io.IOException`: `javax.activation`

**See also**
`writeTo`

---

**writeTo**

```java
public void writeTo(OutputStream os)
```

**Throws**
`IOException`, `MessagingException`

Output the body part as an RFC 822 format stream.

**Specified by:**
`writeTo` in interface `Part`

**Overrides:**
`writeTo` in class `MimeBodyPart`

**Throws:**
`MessagingException`  
`IOException`: if an error occurs writing to the stream or if an error is generated by the `javax.activation` layer.
protected void updateHeaders() throws MessagingException

Content-Transfer-Encoding

updateHeaders

protected void updateHeaders()
    throws MessagingException

Force the Content-Transfer-Encoding header to use the encoding that was specified when this object was created.

Overrides:
    updateHeaders in class MimeBodyPart

Throws:
    MessagingException

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Annotation Type PrePassivate

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PrePassivate

**Implements**: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Bean

Designates a method to receive a callback before a stateful session bean is passivated.

---

Overview Package Tree Deprecated Index Help

PREV CLASS NEXT CLASS FRAMES NO FRAMES
SUMMARY: REQUIRED | OPTIONAL DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:

javax.persistence Annotation Type PrePersist

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PrePersist

**Implements**: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since**: Java Persistence 1.0

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.persistence Annotation Type PreRemove

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PreRemove

Implements: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

since Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

Since:
Java Persistence 1.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type PreUpdate

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface PreUpdate

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

**since** Java Persistence 1.0

Is used to specify callback methods for the corresponding lifecycle event. This annotation may be applied to methods of an entity class, a mapped superclass, or a callback listener class.

**Since:**
Java Persistence 1.0
javax.persistence  Annotation Type
PrimaryKeyJoinColumn

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface PrimaryKeyJoinColumn

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

JOINED   SecondaryTable   OneToOne

JOINED   PrimaryKeyJoinColumn

Customer  ValuedCustomer

@Entity
@Table(name="CUST")
@Inheritance(strategy=JOINED)
@DiscriminatorValue("CUST")
public class Customer { ... }

@Entity
@Table(name="VCUST")
@DiscriminatorValue("VCUST")
@PrimaryKeyJoinColumn(name="CUST_ID")
public class ValuedCustomer extends Customer { ... }

since  Java Persistence 1.0

This annotation specifies a primary key column that is used as a foreign key to join to another table.

It is used to join the primary table of an entity subclass in the JOINED mapping strategy to the primary table of its superclass; it is used within a
SecondaryTable annotation to join a secondary table to a primary table; and it may be used in a OneToOne mapping in which the primary key of the referencing entity is used as a foreign key to the referenced entity.

If no PrimaryKeyJoinColumn annotation is specified for a subclass in the JOINED mapping strategy, the foreign key columns are assumed to have the same names as the primary key columns of the primary table of the superclass.

Example: Customer and ValuedCustomer subclass

```java
@Entity
@Table(name="CUST")
@Inheritance(strategy=JOINED)
@DiscriminatorValue("CUST")
public class Customer {
    ...
}

@Entity
@Table(name="VCUST")
@DiscriminatorValue("VCUST")
@PrimaryKeyJoinColumn(name="CUST_ID")
public class ValuedCustomer extends Customer {
    ...
}
```

Since:
Java Persistence 1.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>columnDefinition</th>
<th>(Optional) The SQL fragment that is used when generating the DDL for the column.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>The name of the primary key column of the current table.</td>
</tr>
<tr>
<td>String</td>
<td>referencedColumnName</td>
<td>(Optional) The name of the primary key column of the table being joined to.</td>
</tr>
</tbody>
</table>

abstract public String name()
name

public abstract String name

The name of the primary key column of the current table.

Defaults to the same name as the primary key column of the primary table of the superclass (JOINED mapping strategy); the same name as the primary key column of the primary table (SecondaryTable mapping); or the same name as the primary key column for the table for the referencing entity (OneToOne mapping)

Default: ""

abstract public String referencedColumnName()

referencedColumnName

public abstract String referencedColumnName

(Optional) The name of the primary key column of the table being joined to.

Defaults to the same name as the primary key column of the primary table of the superclass (JOINED mapping strategy); the same name
as the primary key column of the primary table (SecondaryTable mapping); or the same name as the primary key column for the table for the referencing entity (OneToOne mapping)

Default:

```java
abstract public String columnDefinition()
DDL  SQL

OneToOne
```

```
sql

columnDefinition

public abstract String columnDefinition

(Optional) The SQL fragment that is used when generating the DDL for the column. This should not be specified for a OneToOne primary key association.

Defaults to the generated SQL to create a column of the inferred type.

Default:

```java
""
```
PS:
**javax.persistence**

**Annotation Type**

**PrimaryKeyJoinColumn**

```java
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface PrimaryKeyJoinColumn
```

**Implements:** Annotation

```java
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
```

**PrimaryCustomerJoinColumn**

```java
@Entity
@Table(name="VCUST")
@DiscriminatorValue("VCUST")
@PrimaryKeyJoinColumn(
    @PrimaryKeyJoinColumn(name="CUST_ID",
        referencedColumnName="ID"),
    @PrimaryKeyJoinColumn(name="CUST_TYPE",
        referencedColumnName="TYPE")
)
public class ValuedCustomer extends Customer {
    ...
}
```

**since** Java Persistence 1.0

This annotation groups `PrimaryKeyJoinColumn` annotations. It is used to map composite foreign keys.

**Example 1: ValuedCustomer subclass**

```java
@Entity
@Table(name="VCUST")
@DiscriminatorValue("VCUST")
@PrimaryKeyJoinColumn(
    @PrimaryKeyJoinColumn(name="CUST_ID",
        referencedColumnName="ID"),
    @PrimaryKeyJoinColumn(name="CUST_TYPE",
        referencedColumnName="TYPE")
)
public class ValuedCustomer extends Customer { ... }

Since:
Java Persistence 1.0

Required Element Summary

| PrimaryKeyJoinColumn[] value | One or more PrimaryKeyJoinColumn annotations. |

Element Detail

abstract public PrimaryKeyJoinColumn[] value()

public abstract PrimaryKeyJoinColumn[] value

One or more PrimaryKeyJoinColumn annotations.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.bind  Interface PrintConversionEvent

All Superinterfaces:
   ValidationEvent

All Known Implementing Classes:
   PrintConversionEventImpl

```java
public interface PrintConversionEvent
extends ValidationEvent

Implements: ValidationEvent
Implemented by: PrintConversionEventImpl
```

Java

version $Revision: 1.1 $
since JAXB1.0

See also
   javax.xml.bind.ValidationEvent,
   javax.xml.bind.ValidationEventHandler,
   javax.xml.bind.Marshaller

This event indicates that a problem was encountered while converting data from the Java content tree into its lexical representation.

Since:
   JAXB1.0

Version:
   $Revision: 1.1 $

Author:
   • Ryan Shoemaker, Sun Microsystems, Inc.
   • Kohsuke Kawaguchi, Sun Microsystems, Inc.
   • Joe Fialli, Sun Microsystems, Inc.

See Also:
   ValidationEvent, ValidationEventHandler, Marshaller
Field Summary

Fields inherited from interface javax.xml.bind.ValidationEvent
ERROR, FATAL_ERROR, WARNING

Method Summary

Methods inherited from interface javax.xml.bind.ValidationEvent
getLinkedException, getLocator, getMessage, getSeverity

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.helpers Class PrintConversionEventImpl

java.lang.Object
    - javax.xml.bind.helpers.ValidationEventImpl
        - javax.xml.bind.helpers.PrintConversionEventImpl

All Implemented Interfaces:
    PrintConversionEvent, ValidationEvent

public class PrintConversionEventImpl

extends ValidationEventImpl
implements PrintConversionEvent

Extends: ValidationEventImpl
Implements: PrintConversionEvent

PrintConversionEvent

JAXB ValidationEvent

$Revision: 1.1 $
since JAXB1.0

javax.xml.bind.PrintConversionEvent,
javax.xml.bind.Validator,
javax.xml.bind.ValidationEventHandler,
javax.xml.bind.ValidationEvent,
javax.xml.bind.ValidationEventLocator

See also

Default implementation of the PrintConversionEvent interface.

JAXB providers are allowed to use whatever class that implements the ValidationEvent interface. This class is just provided for a convenience.

Since:
    JAXB1.0
Field Summary

Fields inherited from interface javax.xml.bind.ValidationEvent
ERROR, FATAL_ERROR, WARNING

Constructor Summary

PrintConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator)
Create a new PrintConversionEventImpl.

PrintConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)
Create a new PrintConversionEventImpl.

Method Summary

Methods inherited from class javax.xml.bind.helpers.ValidationEventImpl
getLinkedException, getLocator, getMessage, getSeverity, setLinkedException, setLocator, setMessage, setSeverity, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait wait

Methods inherited from interface javax.xml.bind.ValidationEvent
Constructor Detail

public PrintConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator)

PrintConversionEventImpl

_create severity_ ValidationEvent.WARNING  ValidationEvent.ERROR  ValidationEvent.FATAL_ERROR
_message Locator null
_locator null

Throws IllegalArgumentException: severity

PrintConversionEventImpl

public PrintConversionEventImpl(int _severity, String _message, ValidationEventLocator _locator)

Create a new PrintConversionEventImpl.

Parameters:

_severity - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR
_message - The text message for this event - may be null.
_locator - The locator object for this event - may be null.

Throws: IllegalArgumentException - if an illegal severity field is supplied
severity  ValidationEvent.WARNING
_severity  ValidationEvent.ERROR
ValidationEvent.FATAL_ERROR
_locator null
_locator null
_linkedException null
Throws  IllegalArgumentException: severity

PrintConversionEventImpl

public PrintConversionEventImpl(int _severity,
    String _message,
    ValidationEventLocator _locator,
    Throwable _linkedException)

Create a new PrintConversionEventImpl.

Parameters:
_severity - The severity value for this event. Must be one of
    ValidationEvent.WARNING, ValidationEvent.ERROR, or
    ValidationEvent.FATAL_ERROR
_message - The text message for this event - may be null.
_locator - The locator object for this event - may be null.
_linkedException - An optional linked exception that may
    provide additional information about the event - may be null.

Throws:
IllegalArgumentException - if an illegal severity field is supplied

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES  NO FRAMES  DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.stream.events  Interface ProcessingInstruction

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface ProcessingInstruction
extends XMLEvent

Implements: XMLEvent

version 1.0

An interface that describes the data found in processing instructions

Version:
   1.0
Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

| Fields inherited from interface javax.xml.stream.XMLStreamConstants |
| ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT |

Method Summary

| String | getData () |
| The data section of the processing instruction |
### Method Detail

#### public String getTarget()

    return PI null

**getTarget**

String getTarget()

The target section of the processing instruction

**Returns:**

the String value of the PI or null

#### public String getData()

    return PI null

**getData**

String getData()
The data section of the processing instruction

**Returns:**
the String value of the PI's data or null
javax.enterprise.deploy.spi.status  Class ProgressEvent

java.lang.Object
   \ java.util.EventObject
      \ javax.enterprise.deploy.spi.status.ProgressEvent

All Implemented Interfaces:
   Serializable

public class ProgressEvent
   extends EventObject

   Extends: java.util.EventObject

   See also  javax.enterprise.deploy.spi.status.ProgressListener,
   javax.enterprise.deploy.spi.status.ProgressObject

An event which indicates that a deployment status change has occurred.

See Also:
   ProgressListener, ProgressObject, Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class java.util.EventObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

Constructor Summary

ProgressEvent(Object source, TargetModuleID targetModuleID, DeploymentStatus sCode)

Creates a new object representing a deployment progress event.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeploymentStatus</td>
<td>getDeploymentStatus()</td>
</tr>
<tr>
<td>TargetModuleID</td>
<td>getTargetModuleID()</td>
</tr>
</tbody>
</table>

- **DeploymentStatus**
  - getDeploymentStatus() — Retrieve the status information.

- **TargetModuleID**
  - getTargetModuleID() — Retrieve the TargetModuleID for this event

Methods inherited from class java.util.EventObject:
- getSource, toString

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

```java
public ProgressEvent(Object source, TargetModuleID targetModuleID, DeploymentStatus sCode)
```

- `source` - Event
- `sCode` - ProgressEvent

**ProgressEvent**

```java
public ProgressEvent(Object source, TargetModuleID targetModuleID, DeploymentStatus sCode)
```

Creates a new object representing a deployment progress event.

**Parameters:**
- `source` - the object on which the Event initially occurred.
- `sCode` - the object containing the status information.
public TargetModuleID getTargetModuleID()
TargetModuleID

    return TargetModuleID

getTargetModuleID

public TargetModuleID getTargetModuleID()

    Retrieve the TargetModuleID for this event

    Returns:
        the object containing the TargetModuleID

gDeploymentStatus

public DeploymentStatus getDeploymentStatus()

    return

gDeploymentStatus

public DeploymentStatus getDeploymentStatus()

    Retrieve the status information.

    Returns:
        the object containing the status information.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.enterprise.deploy.spi.status  

**Interface ProgressListener**

*All Superinterfaces:*  
*EventListener*

```java
public interface ProgressListener extends EventListener

Implements: java.util.EventListener
```

The listener interface for receiving deployment progress events.

### Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>handleProgressEvent</code></td>
<td>Invoked when a deployment progress event occurs.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void handleProgressEvent(ProgressEvent event)
```

`event`

**handleProgressEvent**

```java
void handleProgressEvent(ProgressEvent event)
```

Invoked when a deployment progress event occurs.
Parameters:

event - the progress status event.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.enterprise.deploy.spi.status  Interface ProgressObject

public interface ProgressObject

ProgressObject

cancel  isCancelSupported  cancel

The ProgressObject interface tracks and reports the progress of the deployment activities, distribute, start, stop, undeploy.

This class has an optional cancel method. The support of the cancel function can be tested by the isCancelSupported method.

The ProgressObject structure allows the user the option of polling for status or to provide a callback.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addProgressListener(ProgressListener pol)</code></td>
<td>Add a listener to receive Progress events on deployment actions.</td>
</tr>
<tr>
<td><code>cancel()</code></td>
<td>(optional) A cancel request on an in-process operation stops all further processing of the operation and returns the environment to its original state before the operation was executed.</td>
</tr>
<tr>
<td><code>getClientConfiguration(TargetModuleID id)</code></td>
<td>Return the ClientConfiguration object associated with the TargetModuleID.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>getDeploymentStatus()</td>
<td>Retrieve the status of this activity.</td>
</tr>
<tr>
<td>getResultTargetModuleIDs()</td>
<td>Retrieve the list of TargetModuleIDs successfully processed or created by the associated DeploymentManager operation.</td>
</tr>
<tr>
<td>isCancelSupported()</td>
<td>Tests whether the vendor supports a cancel operation for deployment activities.</td>
</tr>
<tr>
<td>isStopSupported()</td>
<td>Tests whether the vendor supports a stop operation for deployment activities.</td>
</tr>
<tr>
<td>removeProgressListener(ProgressListener pol)</td>
<td>Remove a ProgressObject listener.</td>
</tr>
<tr>
<td>stop()</td>
<td>(optional) A stop request on an in-process operation allows the operation on the current TargetModuleID to run to completion but does not process any of the remaining unprocessed TargetModuleID objects.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public DeploymentStatus getDeploymentStatus()
```

Returns:

getDeploymentStatus

```java
DeploymentStatus getDeploymentStatus()
```

Retrieve the status of this activity.

**Returns:**
An object containing the status information.

```java
public TargetModuleID[] getResultTargetModuleIDs()
DeploymentManager TargetModuleID
return TargetModuleID
```

### getResultTargetModuleIDs

```java
TargetModuleID[] getResultTargetModuleIDs()
```

Retrieve the list of TargetModuleIDs successfully processed or created by the associated DeploymentManager operation.

**Returns:**
- a list of TargetModuleIDs.

```java
public ClientConfiguration getClientConfiguration(TargetModuleID id)
TargetModuleID ClientConfiguration
return TargetModuleID ClientConfiguration null
```

### getClientConfiguration

```java
ClientConfiguration getClientConfiguration(TargetModuleID id)
```

Return the ClientConfiguration object associated with the TargetModuleID.

**Returns:**
- ClientConfiguration for a given TargetModuleID or null if none exists.

```java
public boolean isCancelSupported()
```
isCancelSupported

boolean isCancelSupported()

Tests whether the vendor supports a cancel operation for deployment activities.

Returns:
true if canceling an activity is supported by this platform.

public void cancel() throws OperationUnsupportedException

void cancel() throws OperationUnsupportedException

(optional) A cancel request on an in-process operation stops all further processing of the operation and returns the environment to its original state before the operation was executed. An operation that has run to completion cannot be cancelled.

Throws:
OperationUnsupportedException - this optional command is not supported by this implementation.

public boolean isStopSupported()
**isStopSupported**

boolean isStopSupported()

Tests whether the vendor supports a stop operation for deployment activities.

**Returns:**

true if canceling an activity is supported by this platform.

---

**public void stop() throws OperationUnsupportedException**

**TargetModuleID**  **TargetModuleID**

**getResultTargetModuleIDs**  **TargetModuleID**

**Throws**  **OperationUnsupportedException**:

(stop) A stop request on an in-process operation allows the operation on the current TargetModuleID to run to completion but does not process any of the remaining unprocessed TargetModuleID objects. The processed TargetModuleIDs must be returned by the method getResultTargetModuleIDs.

**Throws:**

**OperationUnsupportedException** - this optional command is not supported by this implementation.

---

**public void addProgressListener(ProgressListener pol)**

Progress
addProgressListener

void addProgressListener(ProgressListener pol)

Add a listener to receive Progress events on deployment actions.

Parameters:
   pol - the listener to receive events

See Also:
   ProgressEvent

public void removeProgressListener(ProgressListener pol)

Remove a ProgressObject listener.

Parameters:
   pol - the listener being removed

See Also:
   ProgressEvent
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.xml.bind  **Class PropertyException**

```java
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.xml.bind.JAXBException
              └ javax.xml.bind.PropertyException
```

**All Implemented Interfaces:**

  * [Serializable](https://docs.oracle.com/en/java/javase/11/docs/api/java/lang/Serializable.html)

---

```java
public class PropertyException

    extends JAXBException

Extends: Throwable > Exception > JAXBException
```

**version**  
$Revision: 1.1 $  $Date: 2004/12/14 21:50:40 $

**since**  
JAXB1.0

**See also**  

This exception indicates that an error was encountered while getting or setting a property.

**Since:**

JAXB1.0

**Version:**

$Revision: 1.1 $  $Date: 2004/12/14 21:50:40 $

**Author:**

- Ryan Shoemaker, Sun Microsystems, Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems, Inc.

**See Also:**

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PropertyException(String message)</code></td>
<td>Construct a <code>PropertyException</code> with the specified detail message.</td>
</tr>
<tr>
<td><code>PropertyException(String name, Object value)</code></td>
<td>Construct a <code>PropertyException</code> whose message field is set based on the name of the property and value.toString().</td>
</tr>
<tr>
<td><code>PropertyException(String message, String errorCode)</code></td>
<td>Construct a <code>PropertyException</code> with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td><code>PropertyException(String message, String errorCode, Throwable exception)</code></td>
<td>Construct a <code>PropertyException</code> with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td><code>PropertyException(String message, Throwable exception)</code></td>
<td>Construct a <code>PropertyException</code> with the specified detail message and linkedException.</td>
</tr>
<tr>
<td><code>PropertyException(Throwable exception)</code></td>
<td>Construct a <code>PropertyException</code> with a linkedException.</td>
</tr>
</tbody>
</table>

## Method Summary

Methods inherited from class `javax.xml.bind.JAXBException`
- `getCause`, `getErrorCode`, `getLinkedException`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setLinkedException`, `toString`

Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `setStackTrace`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
## Constructor Detail

```java
public PropertyException(String message)
PropertyExceptionerrorCode  linkedException
null
    message
```

### PropertyException

```java
public PropertyException(String message)
    Construct a PropertyException with the specified detail message. The errorCode and linkedException will default to null.

Parameters:
    message - a description of the exception
```

```java
public PropertyException(String message, String errorCode)
PropertyExceptionerrorCode  linkedException
null
    message
    errorCode
```

### PropertyException

```java
public PropertyException(String message, String errorCode)
    Construct a PropertyException with the specified detail message and vendor specific errorCode. The linkedException will default to null.

Parameters:
```
message - a description of the exception
errorCode - a string specifying the vendor specific error code

public PropertyException(Throwable exception)
linkedException  PropertyException errorCode
null

exception

PropertyException

public PropertyException(Throwable exception)

Construct a PropertyException with a linkedException. The detail message and vendor specific errorCode will default to null.

Parameters:
exception - the linked exception

PropertyException

public PropertyException(String message, Throwable exception)
linkedException  PropertyException errorCode
null

message
exception

PropertyException

public PropertyException(String message, Throwable exception)

Construct a PropertyException with the specified detail message and linkedException. The errorCode will default to null.

Parameters:
message - a description of the exception
exception - the linked exception

public PropertyException(String message, String errorCode, Throwable exception)
errorCode linkedException PropertyException
message
errorCode
exception

PropertyException

public PropertyException(String name, Object value)
value.toString() PropertyException
name
value

PropertyException

public PropertyException(String name, Object value)
Construct a PropertyException whose message field is set based on the name of the property and value.toString().

**Parameters:**
- name - the name of the property related to this exception
- value - the value of the property related to this exception
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONST | METHOD |
FRAMES  NO FRAMES
DETAIL: FIELD | CONST | METHOD |
javax.el Class PropertyNotFoundException

java.lang.Object
  ▼ java.lang.Throwable
     ▼ java.lang.Exception
          ▼ java.lang.RuntimeException
               ▼ javax.el.ELException
                    ▼ javax.el.PropertyNotFoundException

All Implemented Interfaces:
    Serializable

public class PropertyNotFoundException

extends ELException

Extends: Throwable > Exception > RuntimeException > ELException

    ValueExpression    MethodExpression

JavaBean

    since                JSP 2.1

Thrown when a property could not be found while evaluating a
ValueExpression or MethodExpression.

For example, this could be triggered by an index out of bounds while
setting an array value, or by an unreadable property while getting the
value of a JavaBeans property.

Since:
    JSP 2.1
See Also:
    Serialized Form
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PropertyNotFoundException()</code></td>
<td>Creates a <code>PropertyNotFoundException</code> with no detail message.</td>
</tr>
<tr>
<td><code>PropertyNotFoundException(String message)</code></td>
<td>Creates a <code>PropertyNotFoundException</code> with the provided detail message.</td>
</tr>
<tr>
<td><code>PropertyNotFoundException(String pMessage, Throwable pRootCause)</code></td>
<td>Creates a <code>PropertyNotFoundException</code> with the given detail message and root cause.</td>
</tr>
<tr>
<td><code>PropertyNotFoundException(Throwable exception)</code></td>
<td>Creates a <code>PropertyNotFoundException</code> with the given root cause.</td>
</tr>
</tbody>
</table>

### Method Summary

### Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`
- `getCause`
- `getLocalizedMessage`
- `getMessage`
- `getStackTrace`
- `initCause`
- `printStackTrace`
- `printStackTrace`
- `printStackTrace`
- `setStackTrace`
- `toString`

### Methods inherited from class `java.lang.Object`

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `wait`
- `wait`
- `wait`

### Constructor Detail

```java
public PropertyNotFoundException()

public PropertyNotFoundException()
```

```java
public PropertyNotFoundException()
```

```java
public PropertyNotFoundException()
```

```java
public PropertyNotFoundException()
```

```java
public PropertyNotFoundException()
```

```java
public PropertyNotFoundException()
```
Creates a PropertyNotFoundException with no detail message.

```java
public PropertyNotFoundException(String message)

    PropertyNotFoundException
    message

PropertyNotFoundException

public PropertyNotFoundException(String message)

    Creates a PropertyNotFoundException with the provided detail message.

    Parameters:
    message - the detail message

PropertyNotFoundException

public PropertyNotFoundException(Throwable exception)

    PropertyNotFoundException
    exception

PropertyNotFoundException

public PropertyNotFoundException(Throwable exception)

    Creates a PropertyNotFoundException with the given root cause.

    Parameters:
    exception - the originating cause of this exception

PropertyNotFoundException

public PropertyNotFoundException(String pMessage, Throwable pRootCause)

    PropertyNotFoundException
public PropertyNotFoundException(String pMessage, Throwable pRootCause)

Creates a PropertyNotFoundException with the given detail message and root cause.

Parameters:
  pMessage - the detail message
  pRootCause - the originating cause of this exception

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.el  Class PropertyNotFoundException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.faces.FacesException
                  └ javax.faces.el.EvaluationException
                      └ javax.faces.el.PropertyNotFoundException

All Implemented Interfaces:
  Serializable

Deprecated. This has been replaced by PropertyNotFoundException.

Extends: Throwable > Exception > RuntimeException > FacesException > EvaluationException

public class PropertyNotFoundException
extends EvaluationException

An exception caused by a property name that cannot be resolved against a base object.

See Also:
  Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>PropertyNotFoundException()</td>
</tr>
<tr>
<td>Deprecated. Construct a new exception with no detail message</td>
</tr>
</tbody>
</table>
or root cause.

**PropertyNotFoundException***(String message)***

*Deprecated.* Construct a new exception with the specified detail message and no root cause.

**PropertyNotFoundException***(String message, Throwable cause)***

*Deprecated.* Construct a new exception with the specified detail message and root cause.

**PropertyNotFoundException***(Throwable cause)***

*Deprecated.* Construct a new exception with the specified root cause.

---

**Method Summary**

Methods inherited from class javax.faces.*FacesException*

getCause

Methods inherited from class java.lang.*Throwable*

fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.*Object*

close, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

**Constructor Detail**

public PropertyNotFoundException()
**PropertyNotFoundException**

```java
public PropertyNotFoundException()

    Deprecated.

    Construct a new exception with no detail message or root cause.
```

```java
public PropertyNotFoundException(String message)

message
```

**PropertyNotFoundException**

```java
public PropertyNotFoundException(String message)

    Deprecated.

    Construct a new exception with the specified detail message and no root cause.

Parameters:
message - The detail message for this exception
```

```java
public PropertyNotFoundException(Throwables cause)

    (cause == null ? null : cause.toString())

cause
```

**PropertyNotFoundException**
public PropertyNotFoundException( Throwable cause)

    Deprecated.

    Construct a new exception with the specified root cause. The detail
    message will be set to (cause == null ? null : cause.toString())

    Parameters:
    cause - The root cause for this exception

public PropertyNotFoundException(String message, Throwable cause)

    message
    cause

PropertyNotFoundException

public PropertyNotFoundException(String message, Throwable cause)

    Deprecated.

    Construct a new exception with the specified detail message and
    root cause.

    Parameters:
    message - The detail message for this exception
    cause - The root cause for this exception
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.el Class PropertyNotWritableException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.el.ELException
                  └ javax.el.PropertyNotWritableException

All Implemented Interfaces:
  Serializable

public class PropertyNotWritableException
  extends ELException

Extends: Throwable > Exception > RuntimeException > ELException

  ValueExpression

since JSP 2.1

Thrown when a property could not be written to while setting the value on a ValueExpression.

For example, this could be triggered by trying to set a map value on an unmodifiable map.

Since: JSP 2.1

See Also:
  Serialized Form

Constructor Summary
### PropertyNotWritableException

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>()</code></td>
<td>Creates a <code>PropertyNotWritableException</code> with no detail message.</td>
</tr>
<tr>
<td><code>(String pMessage)</code></td>
<td>Creates a <code>PropertyNotWritableException</code> with the provided detail message.</td>
</tr>
<tr>
<td><code>(String pMessage, Throwable pRootCause)</code></td>
<td>Creates a <code>PropertyNotWritableException</code> with the given detail message and root cause.</td>
</tr>
<tr>
<td><code>(Throwable exception)</code></td>
<td>Creates a <code>PropertyNotWritableException</code> with the given root cause.</td>
</tr>
</tbody>
</table>

### Method Summary

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getCause`, `getLocaleMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Constructor Detail

```java
public PropertyNotWritableException()
```

```java
PropertyNotWritableException
```

```java
public PropertyNotWritableException()
```
Creates a PropertyNotWritableException with no detail message.

```java
public PropertyNotWritableException(String pMessage)
```

**PropertyNotWritableException**

```java
public PropertyNotWritableException(String pMessage)
```

Creates a PropertyNotWritableException with the provided detail message.

**Parameters:**

- `pMessage` - the detail message

```
public PropertyNotWritableException(Throwable exception)
```

**PropertyNotWritableException**

```java
public PropertyNotWritableException(Throwable exception)
```

Creates a PropertyNotWritableException with the given root cause.

**Parameters:**

- `exception` - the originating cause of this exception

```
public PropertyNotWritableException(String pMessage, Throwable pRootCause)
```

**PropertyNotWritableException**
PropertyNotWritableException

public PropertyNotWritableException(String pMessage, Throwable pRootCause)

Creates a PropertyNotWritableException with the given detail message and root cause.

Parameters:
  pMessage - the detail message
  pRootCause - the originating cause of this exception

PS:
**Class PropertyResolver**

```java
import java.lang.Object;
import javax.faces.el.PropertyResolver;
```

**Deprecated. This has been replaced by ELResolver.**

**PropertyResolver** represents a pluggable mechanism for accessing a "property" of an underlying Java object instance. Different **PropertyResolver** implementations can support property resolution on instances of different Java classes (such as introspection-based access to properties of a JavaBeans component, or `get()` and `put()` calls on a `java.util.Map` instance).

All implementations must respect the rules for JavaBeans component, `java.util.Map`, `java.util.List` and array instances defined for each method but are allowed to add custom semantics for other types.

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PropertyResolver()</code></td>
<td>Deprecated.</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract Class getType()</td>
<td>(Object base, int index)</td>
<td>Deprecated. Return the java.lang.Class representing the type of the specified index.</td>
</tr>
<tr>
<td>abstract Class getType()</td>
<td>(Object base, Object property)</td>
<td>Deprecated. Return the java.lang.Class representing the type of the specified property.</td>
</tr>
<tr>
<td>abstract Object getValue()</td>
<td>(Object base, int index)</td>
<td>Deprecated. Return the value at the specified index of the specified base object.</td>
</tr>
<tr>
<td>abstract Object getValue()</td>
<td>(Object base, Object property)</td>
<td>Deprecated. Return the value of the specified property from the specified base object.</td>
</tr>
<tr>
<td>abstract boolean isReadOnly()</td>
<td>(Object base, int index)</td>
<td>Deprecated. Checks if the specified index is read-only.</td>
</tr>
<tr>
<td>abstract boolean isReadOnly()</td>
<td>(Object base, Object property)</td>
<td>Deprecated. Checks if the specified property is read-only.</td>
</tr>
<tr>
<td>abstract void setValue()</td>
<td>(Object base, int index, Object value)</td>
<td>Deprecated. Set the value at the specified index of the specified base object.</td>
</tr>
<tr>
<td>abstract void setValue()</td>
<td>(Object base, Object property, Object value)</td>
<td>Deprecated. Set the specified value of the specified property on the specified base object.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
- wait

## Constructor Detail

public PropertyResolver()
PropertyResolver

public PropertyResolver()

    Deprecated.

Method Detail

abstract public Object getValue(Object base, Object property) throws EvaluationException, PropertyNotFoundException

base

Bean base String base Map

    base
    property
    return

    throws EvaluationException:

    throws PropertyNotFoundException: Bean base

getValue

public abstract Object getValue(Object base, Object property) throws EvaluationException, PropertyNotFoundException

    Deprecated.

    Return the value of the specified property from the specified base object.

    For a bean base object, the property is coerced to a String and used as the property name. For all other base object types (e.g., a Map),
the property is used without any coercing.

**Parameters:**
- base - The base object whose property value is to be returned
- property - The property to be returned

**Returns:**
- The property value, or null if base or property is null, or if the property doesn't exist and the base object is a Map instance

**Throws:**
- `EvaluationException` - if an exception is thrown while getting the property value (the thrown exception must be included as the cause property of this exception)
- `PropertyNotFoundException` - if the specified property for a bean base object does not exist or is not readable

```java
abstract public Object getValue(Object base, int index)
throws EvaluationException, PropertyNotFoundException
```

```java
base

    base
index
return

    base null base null
```

**Throws**

- `EvaluationException`:  
- `PropertyNotFoundException`:  

```java
getValue
```

```java
public abstract Object getValue(Object base,
int index)
throws EvaluationException,
PropertyNotFoundException
```

**Deprecated.**

Return the value at the specified index of the specified base object.
Parameters:
- base - The base object whose property value is to be returned
- index - The index of the value to return

Returns:
- The property value, or null if base is null, or if the index is out of bounds for the base object

Throws:
- EvaluationException - if an exception is thrown while getting the property value (the thrown exception must be included as the cause property of this exception)
- PropertyNotFoundException - if the index is out of bounds or if base is null

abstract public void setValue(Object base, Object property, Object value) throws EvaluationException, PropertyNotFoundException

base

Bean base String base Map

Throws EvaluationException:

Throws PropertyNotFoundException: Bean base
null

setValue

public abstract void setValue(Object base, Object property, Object value)
throws EvaluationException, PropertyNotFoundException
Deprecated.

Set the specified value of the specified property on the specified base object.

For a bean base object, the property is coerced to a `String` and used as the property name. For all other base object types (e.g., a Map), the property is used without any coercing.

**Parameters:**
- `base` - The base object whose property value is to be set
- `property` - The property to be set
- `value` - The value of the property to be set

**Throws:**
- `EvaluationException` - if an exception is thrown while setting the property value (the thrown exception must be included as the cause property of this exception)
- `PropertyNotFoundException` - if the specified bean base object property does not exist or is not writeable, or if `base` or `name` is `null`

```java
abstract public void setValue(Object base, int index, Object value) throws EvaluationException, PropertyNotFoundException

base

base

index

value

Throws EvaluationException:

Throws PropertyNotFoundException: base null

setValue

public abstract void setValue(Object base,

```
Set the value at the specified index of the specified base object.

**Parameters:**
- `base` - The base object whose property value is to be set
- `index` - The index of the value to set
- `value` - The value to be set

**Throws:**
- `EvaluationException` - if an exception is thrown while setting the property value (the thrown exception must be included as the cause property of this exception)
- `PropertyNotFoundException` - if the index is out of bounds or if `base` is null

```java
abstract public boolean isReadOnly(Object base, Object property) throws EvaluationException, PropertyNotFoundException
```

```java
Bean base String base Map

  base base
  property
  return base true false

  Throws EvaluationException:
  property
  base

  Throws PropertyNotFoundException: Bean base
  property null
```

`isReadOnly`
public abstract boolean isReadOnly(Object base, Object property)
throws EvaluationException, PropertyNotFoundException

Deprecated.

Checks if the specified property is read-only.

For a bean base object, the property is coerced to a String and used as the property name. For all other base object types (e.g., a Map), the property is used without any coercing.

Parameters:
- base - The base object whose property is to be analyzed
- property - The property to be analyzed

Returns:
- true if the specified property of the specified base object is known to be immutable; otherwise false

Throws:
- EvaluationException - if an exception is thrown while testing the property (the thrown exception must be included as the cause property of this exception)
- PropertyNotFoundException - if the specified bean base object property does not exist or if base or property is null

abstract public boolean isReadOnly(Object base, int index)
throws EvaluationException, PropertyNotFoundException

base
index
return base

base

true
false

base
null

base
null
isReadOnly

public abstract boolean isReadOnly(Object base, int index) throws EvaluationException, PropertyNotFoundException

Deprecated.

Checks if the specified index is read-only.

Parameters:
  base - The base object whose property is to be analyzed
  index - The index of the value whose type is to be returned

Returns:
  true if the value at the specified index of the specified base object is known to be immutable; otherwise, false

Throws:
  EvaluationException - if an exception is thrown while testing the property (the thrown exception must be included as the cause property of this exception)
  PropertyNotFoundException - if the index is out of bounds or if base is null

abstract public Class<T> getType(Object base, Object property) throws EvaluationException, PropertyNotFoundException

java.lang.Class null

Bean base       String base Map
base
property
return java.lang.Class base null
Throws EvaluationException:
Throws PropertyNotFoundException: Bean base property null
getType

public abstract Class getType(Object base, Object property) throws EvaluationException, PropertyNotFoundException

Deprecated.

Return the java.lang.Class representing the type of the specified property. An instance of any Object of that type may be set as the value of that property, as long as the property is not read-only. The actual value of the property, if non-null, is guaranteed to be an instance of this type, or an instance of a subclass of this type. This method is also useful for discovering the type of Objects that may be set as the value of the property.

For a bean base object, the property is coerced to a String and used as the property name. For all other base object types (e.g., a Map), the property is used without any coercing.

Parameters:
  base - The base object whose property is to be analyzed
  property - The property to be analyzed

Returns:
  the java.lang.Class representing the type of the specified property of the specified base object, if it can be determined; otherwise null

Throws:
  EvaluationException - if an exception is thrown while testing the property (the thrown exception must be included as the cause property of this exception)
  PropertyNotFoundException - if the specified bean base object property does not exist or if base or property is null
Throws `EvaluationException`, `PropertyNotFoundException`

`java.lang.Class`

```
public abstract Class getType(Object base, int index)
throws EvaluationException,
PropertyNotFoundException
```

**Deprecated.**

Return the `java.lang.Class` representing the type of the specified index.

**Parameters:**
- `base` - The base object whose property is to be analyzed
- `index` - The index of the value whose type is to be returned

**Returns:**
- The `java.lang.Class` representing the type of value at the specified index of the specified base object, if it can be determined; otherwise null

**Throws:**
- `EvaluationException` - if an exception is thrown while testing the property (the thrown exception must be included as the cause property of this exception)
- `PropertyNotFoundException` - if the index is out of bounds or if `base` is null

---

**Overview  Package  Tree  Deprecated  Index  Help**
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws  **Class ProtocolException**

`java.lang.Object`  
  `java.lang.Throwable`  
  `java.lang.Exception`  
  `java.lang.RuntimeException`  
  `javax.xml.ws.WebServiceException`  
  `javax.xml.ws.ProtocolException`

**All Implemented Interfaces:**  
[Serializable](#)

**Direct Known Subclasses:**  
[HttpException](#),  
[SOAPFaultException](#)

---

```java
public class ProtocolException
extends WebServiceException

Extends: Throwable > Exception > RuntimeException > WebServiceException  
Extended by: HTTPException, SOAPFaultException
```

**ProtocolException**  

**since**  
JAX-WS 2.0

The `ProtocolException` class is a base class for exceptions related to a specific protocol binding. Subclasses are used to communicate protocol level fault information to clients and may be used on the server to control the protocol specific fault representation.

**Since:**  
JAX-WS 2.0

**See Also:**  
[Serialized Form](#)
Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtocolException()</td>
<td>Constructs a new protocol exception with null as its detail message.</td>
</tr>
<tr>
<td>ProtocolException(String message)</td>
<td>Constructs a new protocol exception with the specified detail message.</td>
</tr>
<tr>
<td>ProtocolException(String message, Throwable cause)</td>
<td>Constructs a new runtime exception with the specified detail message and cause.</td>
</tr>
<tr>
<td>ProtocolException(Throwable cause)</td>
<td>Constructs a new runtime exception with the specified cause and a detail message of (cause==null ?</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

class ProtocolException()
null cause
Throwable.initCause(java.lang Throwable)
**ProtocolException**

```java
public ProtocolException()
```

Constructs a new protocol exception with null as its detail message. The cause is not initialized, and may subsequently be initialized by a call to Throwable.initCause(java.lang.Throwable).

```java
public ProtocolException(String message)
cause Throwable.initCause(java.lang.Throwable)
```

```java
message Throwable.getMessage()
```

**ProtocolException**

```java
public ProtocolException(String message)
```

Constructs a new protocol exception with the specified detail message. The cause is not initialized, and may subsequently be initialized by a call to Throwable.initCause(java.lang.Throwable).

**Parameters:**

- `message` - the detail message. The detail message is saved for later retrieval by the Throwable.getMessage() method.

```java
public ProtocolException(String message, Throwable cause)
```

```java
message Throwable.getMessage()
cause cause Throwable.getCause() null
```

**ProtocolException**
public ProtocolException(String message, Throwable cause)

Constructs a new runtime exception with the specified detail message and cause. Note that the detail message associated with cause is not automatically incorporated in this runtime exception's detail message.

Parameters:
message - the detail message (which is saved for later retrieval by the Throwable.getMessage() method).
cause - the cause (which is saved for later retrieval by the Throwable.getCause() method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

public ProtocolException(Throwable cause)

cause (cause==null ? null :cause.toString())

cause

ProtocolException

public ProtocolException(Throwable cause)

Constructs a new runtime exception with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause). This constructor is useful for runtime exceptions that are little more than wrappers for other throwables.

Parameters:
cause - the cause (which is saved for later retrieval by the Throwable.getCause() method). (A null value is permitted, and indicates that the cause is nonexistent or unknown.)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail Class Provider

java.lang.Object
  ↓ javax.mail.Provider

public class Provider
  extends Object

Inner classes: Provider.Type

Provider  javamail.providers  javamail.default.providers
Provider

  version  1.11, 07/05/04

The Provider is a class that describes a protocol implementation. The values typically come from the javamail.providers and javamail.default.providers resource files. An application may also create and register a Provider object to dynamically add support for a new provider.

Version: 1.11, 07/05/04
Author: Max Spivak, Bill Shannon

---

## Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>Provider.Type</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>This inner class defines the Provider type.</td>
</tr>
</tbody>
</table>

## Constructor Summary

Provider(Provider.Type type, String protocol, String classname, String vendor, String version)
Create a new provider of the specified type for the specified protocol.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getClassName()</code></td>
<td>Returns name of the class that implements the protocol</td>
</tr>
<tr>
<td><code>getProtocol()</code></td>
<td>Returns the protocol supported by this Provider</td>
</tr>
<tr>
<td><code>getType()</code></td>
<td>Returns the type of this Provider</td>
</tr>
<tr>
<td><code>getVendor()</code></td>
<td>Returns name of vendor associated with this implementation or null</td>
</tr>
<tr>
<td><code>getVersion()</code></td>
<td>Returns version of this implementation or null if no version</td>
</tr>
<tr>
<td><code>toString()</code></td>
<td>Overrides Object.toString()</td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public Provider(Provider.Type type, String protocol, String classname, String vendor, String version)
```

- `type` Type.STORE  Type.TRANSPORT
- `protocol`            
- `classname`            
- `vendor` null
public Provider(Provider.Type type,
        String protocol,
        String classname,
        String vendor,
        String version)

Create a new provider of the specified type for the specified protocol. The specified class implements the provider.

Parameters:
  type - Type.STORE or Type.TRANSPORT
  protocol - valid protocol for the type
  classname - class name that implements this protocol
  vendor - optional string identifying the vendor (may be null)
  version - optional implementation version string (may be null)

Since:
   JavaMail 1.4

Method Detail

public Provider.Type getType()

getType

public Provider.Type getType()

  Returns the type of this Provider

public String getProtocol()
Provider

definition

public String getProtocol()

    Returns the protocol supported by this Provider

public String getClassName()

    Returns name of the class that implements the protocol

public String getVendor()
    null

    Returns name of vendor associated with this implementation or null

public String getVersion()
    null

    Returns version information about the implementation.
public String getVersion()

Returns version of this implementation or null if no version

public String toString()
Object.toString()

toString

public String toString()

Overrides Object.toString()

Overrides:
toString in class Object

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws  Interface Provider<T>

public interface Provider<T>

    Provider SEI
    ServiceMode Provider

since JAX-WS 2.0
See javax.xml.transform.Source,
also javax.xml.soap.SOAPMessage, javax.xml.ws.ServiceMode

Service endpoints may implement the Provider interface as a dynamic alternative to an SEI. Implementations are required to support Provider<Source> and Provider<SOAPMessage>. The ServiceMode annotation can be used to control whether the Provider instance will receive entire protocol messages or just message payloads.

Since: JAX-WS 2.0
See Also: Source, SOAPMessage, ServiceMode

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>invoke(T request)</td>
<td>Invokes an operation according to the contents of the request message.</td>
</tr>
</tbody>
</table>

### Method Detail
invoke

I invoke(I request)

Invokes an operation according to the contents of the request message.

Parameters:
request - The request message or message payload.

Returns:
The response message or message payload. May be null if there is no response.

Throws:
WebServiceException - if there is an error processing request. The cause of the WebServiceException may be set to a subclass of ProtocolException to control the protocol level representation of the exception.

See Also:
MessageContext, ProtocolException

---

Overview  Package  Tree  Deprecated  Index  Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws.spi  **Class Provider**

`java.lang.Object`  
  `javax.xml.ws.spi.Provider`

---

public abstract class **Provider**

extends `Object`

---

<table>
<thead>
<tr>
<th>Service Mode</th>
<th>SEI</th>
<th>Provider&lt;Source&gt;</th>
<th>Provider&lt;SOAPMessage&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>since</td>
<td></td>
<td>Provider</td>
<td></td>
</tr>
</tbody>
</table>

**Service provider for** `ServiceDelegate` and `Endpoint` objects.

**Since:**

JAX-WS 2.0

---

### Field Summary

| static String JAXWSPROVIDER_PROPERTY | A constant representing the property used to lookup the name of a `Provider` implementation class. |

### Constructor Summary

| protected `Provider()` | Creates a new instance of `Provider` |

### Method Summary

| `createAndPublishEndpoint(String address, |
Abstract Endpoint

Endpoint createEndpoint(String bindingId, Object implementor)

Creates and publishes an endpoint object with the specified address and implementation object.

Abstract Endpoint

createEndpoint(String bindingId, Object implementor)

Creates an endpoint object with the provided binding and implementation object.

Abstract ServiceDelegate

ServiceDelegate createServiceDelegate(URL wsdlDocumentLocation, QName serviceName, Class serviceClass)

Creates a service delegate object.

Static Provider

provider()

Creates a new provider object.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

JAXWSPROVIDER_PROPERTY

public static final String JAXWSPROVIDER_PROPERTY

A constant representing the property used to lookup the name of a Provider implementation class.

See Also:

Constant Field Values

Constructor Detail

Provider
protected Provider()

    Creates a new instance of Provider

## Method Detail

provider

public static Provider provider()

    Creates a new provider object.

    The algorithm used to locate the provider subclass to use consists of
    the following steps:

    - If a resource with the name of META-INF/services/javax.xml.ws.spi.Provider exists, then its first
      line, if present, is used as the UTF-8 encoded name of the
      implementation class.
    - If the $java.home/lib/jaxws.properties file exists and it is
      readable by the java.util.Properties.load(InputStream)
      method and it contains an entry whose key is
      javax.xml.ws.spi.Provider, then the value of that entry is used
      as the name of the implementation class.
    - If a system property with the name javax.xml.ws.spi.Provider
      is defined, then its value is used as the name of the
      implementation class.
    - Finally, a default implementation class name is used.

createServiceDelegate

public abstract ServiceDelegate createServiceDelegate(URL wsdlDocume

QName serviceN

class serviceC
Creates a service delegate object.

**Parameters:**

wsdlDocumentLocation - A URL pointing to the WSDL document for the service, or null if there isn't one.

serviceName - The qualified name of the service.

serviceClass - The service class, which must be either javax.xml.ws.Service or a subclass thereof.

**Returns:**

The newly created service delegate.

---

createEndpoint

```java
public abstract Endpoint createEndpoint(String bindingId, Object implementor)
```

Creates an endpoint object with the provided binding and implementation object.

**Parameters:**

bindingId - A URI specifying the desired binding (e.g. SOAP/HTTP)

implementor - A service implementation object to which incoming requests will be dispatched. The corresponding class must be annotated with all the necessary Web service annotations.

**Returns:**

The newly created endpoint.

---

createAndPublishEndpoint

```java
public abstract Endpoint createAndPublishEndpoint(String address, Object implementor)
```

Creates and publishes an endpoint object with the specified address
and implementation object.

**Parameters:**

address - A URI specifying the address and transport/protocol to use. A http: URI must result in the SOAP 1.1/HTTP binding being used. Implementations may support other URI schemes.

implementor - A service implementation object to which incoming requests will be dispatched. The corresponding class must be annotated with all the necessary Web service annotations.

**Returns:**

The newly created endpoint.
**javax.mail**  **Class Provider.Type**

import java.lang.Object
- javax.mail.Provider.Type

**Enclosing class:**
- Provider

`public static class Provider.Type
extends Object`

**Contained within:** Provider

Provider STORE  TRANSPORT

This inner class defines the Provider type. Currently, STORE and TRANSPORT are the only two provider types supported.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static Provider.Type STORE</td>
</tr>
<tr>
<td>static Provider.Type TRANSPORT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String toString()</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
clonause, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
Field Detail

STORE

public static final Provider.Type STORE

TRANSPORT

public static final Provider.Type TRANSPORT

Method Detail

public String toString()

toString

public String toString()

Overrides:
 toString in class Object
PS:
javax.xml.rpc.holders  Class QNameHolder

java.lang.Object
   └─javax.xml.rpc.holders.QNameHolder

All Implemented Interfaces:
   Holder

public final class QNameHolder
  extends Object
  implements Holder

Implements: Holder

Field Summary

| QName value |

Constructor Summary

QNameHolder()

QNameHolder(QName myQName)

Method Summary

Methods inherited from class java.lang.Object: clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Field Detail

value

public QName value

Constructor Detail

public QNameHolder()

QNameHolder

public QNameHolder()

public QNameHolder(javax.xml.namespace.QName myQName)

QNameHolder

public QNameHolder(QName myQName)

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY: NESTED  |  FIELD  |  CONSTRUCTOR  |  METHOD
DETAIL: FIELD  |  CONSTRUCTOR  |  METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
public interface Query

since Java Persistence 1.0

Interface used to control query execution.

**Since:**
Java Persistence 1.0

---

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int executeUpdate()</code></td>
<td>Execute an update or delete statement.</td>
</tr>
<tr>
<td><code>List getResultList()</code></td>
<td>Execute a SELECT query and return the query results as a List.</td>
</tr>
<tr>
<td><code>Object getSingleResult()</code></td>
<td>Execute a SELECT query that returns a single result.</td>
</tr>
<tr>
<td><code>Query setFirstResult(int startPosition)</code></td>
<td>Set the position of the first result to retrieve.</td>
</tr>
<tr>
<td><code>Query setFlushMode(FlushModeType flushMode)</code></td>
<td>Set the flush mode type to be used for the query execution.</td>
</tr>
<tr>
<td><code>Query setHint(String hintName, Object value)</code></td>
<td>Set an implementation-specific hint.</td>
</tr>
<tr>
<td><code>Query setMaxResults(int maxResult)</code></td>
<td>Set the maximum number of results to retrieve.</td>
</tr>
<tr>
<td><code>setParameter(int position, Calendar value, TemporalType temporalType)</code></td>
<td>Set a parameter for a query.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>setParameter</code></td>
<td>Bind an instance of <code>java.util.Calendar</code> to a positional parameter.</td>
</tr>
<tr>
<td><code>setParameter(int position, Date value, TemporalType temporalType)</code></td>
<td>Bind an instance of <code>java.util.Date</code> to a positional parameter.</td>
</tr>
<tr>
<td><code>setParameter(int position, Object value)</code></td>
<td>Bind an argument to a positional parameter.</td>
</tr>
<tr>
<td><code>setParameter(String name, Calendar value, TemporalType temporalType)</code></td>
<td>Bind an instance of <code>java.util.Calendar</code> to a named parameter.</td>
</tr>
<tr>
<td><code>setParameter(String name, Date value, TemporalType temporalType)</code></td>
<td>Bind an instance of <code>java.util.Date</code> to a named parameter.</td>
</tr>
<tr>
<td><code>setParameter(String name, Object value)</code></td>
<td>Bind an argument to a named parameter.</td>
</tr>
</tbody>
</table>

### Method Detail

**public java.util.List&lt;E&gt; getResultList()**

SELECT List

```
return
```

Throws

- `IllegalStateException`: Java Persistence
- `IllegalStateException`: UPDATE
- `IllegalStateException`: DELETE

**getResultList**

```
List getResultList()
```

Execute a SELECT query and return the query results as a List.

**Returns:**

- a list of the results

**Throws:**
IllegalStateException - if called for a Java Persistence query language UPDATE or DELETE statement

public Object getSingleResult()

SELECT
return

Throws

NoResultException:
NonUniqueResultException:
IllegalStateException: Java Persistence UPDATE DELETE

getSingleResult

Object getSingleResult()

Execute a SELECT query that returns a single result.

Returns:
the result

Throws:

NoResultException - if there is no result
NonUniqueResultException - if more than one result
IllegalStateException - if called for a Java Persistence query language UPDATE or DELETE statement

public int executeUpdate()

return

Throws
IllegalStateException: Java Persistence SELECT
TransactionRequiredException:

executeUpdate
int executeUpdate()

Execute an update or delete statement.

**Returns:**
the number of entities updated or deleted

**Throws:**
- `IllegalStateException` - if called for a Java Persistence query language SELECT statement
- `TransactionRequiredException` - if there is no transaction

---

public `Query` setMaxResults(int maxResult)

```
maxResult
return
```

**Throws**
`IllegalArgumentException:`

---

`setMaxResults`

`Query` `setMaxResults`(int `maxResult`)

Set the maximum number of results to retrieve.

**Parameters:**
- `maxResult`

**Returns:**
the same query instance

**Throws:**
- `IllegalArgumentException` - if argument is negative

---

public `Query` setFirstResult(int startPosition)

```
startPosition 0
return
```

**Throws**
`IllegalArgumentException:`
setFirstResult

*Query setFirstResult*(int startPosition)

Set the position of the first result to retrieve.

**Parameters:**
- startPosition - the start position of the first result, numbered from 0

**Returns:**
- the same query instance

**Throws:**
- `IllegalArgumentException` - if argument is negative

---

public *Query setHint*(String hintName, Object value)

**hintName**
- value

**return**
- Throws: `IllegalArgumentException`

---

setHint

*Query setHint*(String hintName, Object value)

Set an implementation-specific hint. If the hint name is not recognized, it is silently ignored.

**Parameters:**
- hintName -
- value -

**Returns:**
- the same query instance
Throws:

*IllegalArgumentException* - if the second argument is not valid for the implementation

```java
public Query setParameter(String name, Object value)
```

- **name**
- **value**
- return
- **Throws**  IllegalArgument Exception:

**setParameter**

```java
Query setParameter(String name, Object value)
```

Bind an argument to a named parameter.

**Parameters:**
- name - the parameter name
- value -

**Returns:**
- the same query instance

**Throws:**

*IllegalArgumentException* - if parameter name does not correspond to parameter in query string or argument is of incorrect type

```java
public Query setParameter(String name, java.util.Date value, TemporalType temporalType)
```

- **name**
- **value**
- **temporalType**
setParameter

Query setParameter(String name, Date value, TemporalType temporalType)

Bind an instance of java.util.Date to a named parameter.

Parameters:
- name -
- value -
- temporalType -

Returns:
- the same query instance

Throws:
- IllegalArgumentException - if parameter name does not correspond to parameter in query string

public Query setParameter(String name, java.util.Calendar value, TemporalType temporalType)

java.util.Calendar
- name
- value
- temporalType

return

Throws IllegalArgumentException:
Bind an instance of java.util.Calendar to a named parameter.

**Parameters:**
- name - 
- value - 
- temporalType -

**Returns:**
the same query instance

**Throws:**
- **IllegalArgumentException** - if parameter name does not correspond to parameter in query string

---

```java
public Query setParameter(int position, Object value)
```

- **position**
- **value**
- **return**

**Throws**
- IllegalArgumentException:

---

**setParameter**

```java
Query setParameter(int position, Object value)
```

 Bind an argument to a positional parameter.

**Parameters:**
- position -
- value -

**Returns:**
the same query instance

**Throws:**
- **IllegalArgumentException** - if position does not correspond to positional parameter of query or argument is of incorrect type
public Query setParameter(int position, java.util.Date value, TemporalType temporalType)
java.util.Date
  position
  value
  temporalType
  return
  Throws IllegalArgumentException:

setParameter

Query setParameter(int position, Date value, TemporalType temporalType)
Bind an instance of java.util.Date to a positional parameter.

Parameters:
  position -
  value -
  temporalType -

Returns:
  the same query instance

Throws:
  IllegalArgumentException - if position does not correspond to positional parameter of query

public Query setParameter(int position, java.util.Calendar value, TemporalType temporalType)
java.util.Calendar
  position
  value
  temporalType
  return
  Throws IllegalArgumentException:
setParameter

```java
Query setParameter(int position,
    Calendar value,
    TemporalType temporalType)
```

Bind an instance of java.util.Calendar to a positional parameter.

**Parameters:**
- position -
- value -
- temporalType -

**Returns:**
the same query instance

**Throws:**
IllegalArgumentException - if position does not correspond to positional parameter of query

---

public Query setFlushMode(FlushModeType flushMode)

```java
flushMode
```

setFlushMode

```java
Query setFlushMode(FlushModeType flushMode)
```

Set the flush mode type to be used for the query execution. The flush mode type applies to the query regardless of the flush mode type in use for the entity manager.

**Parameters:**
- flushMode -

---

Overview  Package  Tree  Deprecated  Index  Help
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The Query interface encapsulates a query in a declarative query language. Currently a Query can be an SQL query or an ebXML Filter Query only. In future support for other query languages such as XQL query may be added. The query must conform to a fixed schema as defined by the JAXR specification.

Author:
Farrukh S. Najmi

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static int</strong> QUERY_TYPE_EBXML_FILTER_QUERY</td>
</tr>
<tr>
<td><strong>static int</strong> QUERY_TYPE_SQL</td>
</tr>
<tr>
<td><strong>static int</strong> QUERY_TYPE_XQUERY</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>int getType()</strong></td>
</tr>
<tr>
<td><strong>String toString()</strong></td>
</tr>
</tbody>
</table>
### Field Detail

**QUERY_TYPE_SQL**

```java
static final int QUERY_TYPE_SQL
```

An SQL query type.

**See Also:**
- [Constant Field Values](#)

---

**QUERY_TYPE_XQUERY**

```java
static final int QUERY_TYPE_XQUERY
```

A W3C XQuery type.

**See Also:**
- [Constant Field Values](#)

---

**QUERY_TYPE_EBXML_FILTER_QUERY**

```java
static final int QUERY_TYPE_EBXML_FILTER_QUERY
```

An OASIS ebXML Registry XML Filter Query type.

**See Also:**
- [Constant Field Values](#)
public int getType() throws JAXRException
Query QUERY_TYPE_SQL

1

return

Throws: JAXRException: JAXR
See also QUERY_TYPE_SQL, QUERY_TYPE_XQUERY,
QUERY_TYPE_EBXML_FILTER_QUERY

getType

int getType() throws JAXRException

Gets the type of Query (for example, QUERY_TYPE_SQL).

Capability Level: 1

Returns:
the type of query

Throws: JAXRException - If the JAXR provider encounters an internal error

See Also:
QUERY_TYPE_SQL, QUERY_TYPE_XQUERY,
QUERY_TYPE_EBXML_FILTER_QUERY

public String toString()
String SQL

1

return String
**toString**

`String toString()`

Returns the String representing the query. For example, in the case of an SQL query, returns the SQL query as a string.

**Capability Level: 1**

**Overrides:**
`toString` in class `Object`

**Returns:**
the String representation for this query
javax.persistence  Annotation Type QueryHint

@Target(value={})
@Retention(value=RUNTIME)
public @interface QueryHint

**Implements:** Annotation
@Target(value={})
@Retention(value=RUNTIME)

**Query** since Java Persistence 1.0

An implementation-specific *query* hint.

**Since:**
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>Name of the hint</td>
</tr>
<tr>
<td>value</td>
<td>String</td>
<td>Value of the hint</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String name()
public abstract String name

Name of the hint

abstract public String value()

value

public abstract String value

Value of the hint
javax.xml.registry  **Interface QueryManager**

**All Known Subinterfaces:**  
BusinessQueryManager, DeclarativeQueryManager

**public interface QueryManager**

**Implemented by:** BusinessQueryManager, DeclarativeQueryManager

**API**  
QueryManager

This is the common base interface for all QueryManagers in the API.

**Author:**  
Farrukh S. Najmi

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getRegistryObject</strong> <em>(String id)</em></td>
<td>Gets the RegistryObject specified by the Id.</td>
</tr>
<tr>
<td><strong>getRegistryObject</strong> <em>(String id, String objectType)</em></td>
<td>Gets the RegistryObject specified by the Id and type of object.</td>
</tr>
<tr>
<td><strong>getRegistryObjects</strong> ()</td>
<td>Gets the RegistryObjects owned by the caller.</td>
</tr>
<tr>
<td><strong>getRegistryObjects</strong> <em>(Collection objectKeys)</em></td>
<td>Gets the specified RegistryObjects.</td>
</tr>
<tr>
<td><strong>getRegistryObjects</strong> <em>(Collection objectKeys, String objectTypes)</em></td>
<td>Gets the specified RegistryObjects.</td>
</tr>
<tr>
<td><strong>getRegistryObjects</strong> <em>(String objectType)</em></td>
<td>Gets the RegistryObjects owned by the caller, that are of the specified type.</td>
</tr>
</tbody>
</table>
Returns the parent RegistryService that created this object.

Method Detail

public RegistryObject getRegistryObject(String id, String objectType) throws JAXRException

ID RegistryObject

0

id RegistryObject Key ID

objectType LifeCycleManager

return OrganizationUser RegistryObject

Throws JAXRException: JAXR

getRegistryObject

RegistryObject getRegistryObject(String id, String objectType)

Throws JAXRException

Gets the RegistryObject specified by the Id and type of object.

Capability Level: 0

Parameters:

id - is the id of the Key for a RegistryObject.

objectType - is a constant definition from LifeCycleManager that specifies the type of object desired.

Returns:

the RegistryObject, returned as its concrete type (e.g. Organization, User etc.).

Throws:
public RegistryObject getRegistryObject(String id) throws JAXRException
ID RegistryObject

1

<table>
<thead>
<tr>
<th>id</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>OrganizationUser RegistryObject</td>
</tr>
</tbody>
</table>

Throws: JAXRException - If the JAXR provider encounters an internal error

getRegistryObject

RegistryObject getRegistryObject(String id) throws JAXRException

Gets the RegistryObject specified by the Id.

Capability Level: 1

Parameters:
- id - the id for the desired object

Returns:
- the RegistryObject, returned as its concrete type (e.g. Organization, User etc.).

Throws:
- JAXRException - If the JAXR provider encounters an internal error

public BulkResponse getRegistryObjects(java.util.Collection<E> objectKeys) throws JAXRException
RegistryObjectOrganizationUser

1

objectKeys | Key Collection
--- | ---
return | BulkResponse RegistryObjectOrganizationUser Collection
Throws | JAXRException: JAXR

getRegistryObjects

BulkResponse getRegistryObjects(Collection objectKeys) throws JAXRException

Gets the specified RegistryObjects. The objects are returned as their concrete type (e.g. Organization, User etc.).

Capability Level: 1

Parameters:
objectKeys - a Collection of Key objects for the desired objects

Returns:
BulkResponse containing a heterogeneous Collection of RegistryObjects (e.g. Organization, User etc.).

Throws:
JAXRException - If the JAXR provider encounters an internal error

public BulkResponse getRegistryObjects(java.util.Collection<E> objectKeys, String objectTypes) throws JAXRException
RegistryObjectOrganizationUser

0

objectKeys | Key Collection
getRegistryObjects

BulkResponse getRegistryObjects(Collection objectKeys, String objectTypes)
throws JAXRException

Gets the specified RegistryObjects. The objects are returned as their concrete type (e.g. Organization, User etc.).

Capability Level: 0

Parameters:
  objectKeys - a Collection of Key objects for the desired objects
  objectTypes - a Collection of String objects that allow filtering desired objects by their type

Returns:
  BulkResponse containing a heterogeneous Collection of RegistryObjects (e.g. Organization, User etc.).

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public BulkResponse getRegistryObjects() throws JAXRException

RegistryObjectOrganizationUser

0

return BulkResponse RegistryObjectOrganizationUser Collection
Throws JAXRException: JAXR
getRegistryObjects

**BulkResponse getRegistryObjects() throws JAXRException**

Gets the RegistryObjects owned by the caller. The objects are returned as their concrete type (e.g. Organization, User etc.).

**Capability Level: 0**

**Returns:**
- BulkResponse containing a heterogeneous Collection of RegistryObjects (e.g. Organization, User etc.).

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

```java
public BulkResponse getRegistryObjects(String objectType) throws JAXRException
RegistryObject RegistryObject
OrganizationUser

0

objectType  LifeCycleManager
return BulkResponse RegistryObjectOrganizationUser
Collection
Throws JAXRException: JAXR
See also ORGANIZATION
```

getRegistryObjects

**BulkResponse getRegistryObjects(String objectType) throws JAXRException**
Gets the RegistryObjects owned by the caller, that are of the specified type. The objects are returned as their concrete type (e.g. Organization, User etc.).

**Capability Level: 0**

**Parameters:**
- **objectType** - Is a constant that defines the type of object sought. See LifeCycleManager for constants for object types.

**Returns:**
BulkResponse containing a heterogeneous Collection of RegistryObjects (e.g. Organization, User etc.).

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

**See Also:**
LifeCycleManager.ORGANIZATION

```java
public RegistryService getRegistryService() throws JAXRException

RegistryService
0

return RegistryService

Throws JAXRException: JAXR

associates
<javax.xml.registry.RegistryService>

getRegistryService

RegistryService getRegistryService()

throws JAXRException

Returns the parent RegistryService that created this object.

**Capability Level: 0**
```
Returns:
the RegistryService created this object

Throws:
JAXRException - If the JAXR provider encounters an internal error
javax.jms Interface Queue

All Superinterfaces:  
Destination

All Known Subinterfaces:  
TemporaryQueue

public interface Queue
extends Destination

Implements: Destination
Implemented by: TemporaryQueue

<table>
<thead>
<tr>
<th>Queue</th>
<th>JMS API</th>
<th>Destination</th>
<th>Queue</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageConsumer</td>
<td>MessageProducer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Session.CreateConsumer(Destination destination)
- Session.CreateProducer(Destination destination)

JMS API

version 1.1 February 2 - 2000

See also createConsumer(Destination), createProducer(Destination), createQueue(String), createQueue(String)

A Queue object encapsulates a provider-specific queue name. It is the way a client specifies the identity of a queue to JMS API methods. For those methods that use a Destination as a parameter, a Queue object used as an argument. For example, a queue can be used to create a MessageConsumer and a MessageProducer by calling:

- Session.CreateConsumer(Destination destination)
- Session.CreateProducer(Destination destination)
The actual length of time messages are held by a queue and the consequences of resource overflow are not defined by the JMS API.

Version:
1.1 February 2 - 2000

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
Session.createConsumer(Destination), Session.createProducer(Destination), Session.createQueue(String), QueueSession.createQueue(String)

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String getQueueName()</strong></td>
<td>Gets the name of this queue.</td>
</tr>
<tr>
<td><strong>String toString()</strong></td>
<td>Returns a string representation of this object.</td>
</tr>
</tbody>
</table>

---

### Method Detail

**public String getQueueName() throws JMSException**

```java
return
```

**Throws**

- JMSException
- Queue
- JMS

**getQueueName**

```java
String getQueueName() throws JMSException
```
Gets the name of this queue.

Clients that depend upon the name are not portable.

**Returns:**
the queue name

**Throws:**
JMSException - if the JMS provider implementation of `Queue` fails to return the queue name due to some internal error.

---

```java
public String toString()

    return toString()
```

toString

`String` `toString()`

Returns a string representation of this object.

**Overrides:**
toString in class `Object`

**Returns:**
the provider-specific identity values for this queue
javax.jms  Interface QueueBrowser

public interface QueueBrowser

QueueBrowser

getEnumeration  java.util.Enumeration

JMS API  JMS

Session  QueueSession  QueueBrowser

version  1.1 April 9, 2002

See  createBrowser, createBrowser,
also  javax.jms.QueueReceiver

A client uses a QueueBrowser object to look at messages on a queue without removing them.

The getEnumeration method returns a java.util.Enumeration that is used to scan the queue's messages. It may be an enumeration of the entire content of a queue, or it may contain only the messages matching a message selector.

Messages may be arriving and expiring while the scan is done. The JMS API does not require the content of an enumeration to be a static snapshot of queue content. Whether these changes are visible or not depends on the JMS provider.

A QueueBrowser can be created from either a Session or a QueueSession.

Version:
   1.1 April 9, 2002

Author:
   Mark Hapner, Rich Burridge, Kate Stout
See Also:

- `Session.createBrowser(javax.jms.Queue)`
- `QueueSession.createBrowser(javax.jms.Queue)`
- `QueueReceiver`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void close()</code></td>
<td>Closes the QueueBrowser.</td>
</tr>
<tr>
<td><code>Enumeration getEnumeration()</code></td>
<td>Gets an enumeration for browsing the current queue messages in the order they would be received.</td>
</tr>
<tr>
<td><code>String getMessageSelector()</code></td>
<td>Gets this queue browser's message selector expression.</td>
</tr>
<tr>
<td><code>Queue getQueue()</code></td>
<td>Gets the queue associated with this queue browser.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public Queue getQueue() throws JMSException {
    return ...
}
```

**Throws:**

- `JMSException`: JMS

**getQueue**

```java
Queue getQueue() throws JMSException {
    Gets the queue associated with this queue browser.
    
    **Returns:**
    
    the queue
```
Throws:

`JMSException` - if the JMS provider fails to get the queue associated with this browser due to some internal error.

```java
public String getMessageSelector() throws JMSException
{
    return null;
}
```

Throws

`JMSException`: JMS

`getMessageSelector`

```java
String getMessageSelector() throws JMSException;
```

Gets this queue browser's message selector expression.

Returns:

this queue browser's message selector, or null if no message selector exists for the message consumer (that is, if the message selector was not set or was set to null or the empty string)

Throws:

`JMSException` - if the JMS provider fails to get the message selector for this browser due to some internal error.

```java
public java.util Enumeration<E> getEnumeration() throws JMSException
{
    return null;
}
```

Throws

`JMSException`: JMS

`getEnumeration`

```java
Enumeration getEnumeration();
```
throws JMSException

Gets an enumeration for browsing the current queue messages in the order they would be received.

Returns:
an enumeration for browsing the messages

Throws:
   JMSException - if the JMS provider fails to get the enumeration for this browser due to some internal error.

public void close() throws JMSException

QueueBrowser

QueueBrowser Java

Throws JMSException: JMS

close

void close()
   throws JMSException

Closes the QueueBrowser.

Since a provider may allocate some resources on behalf of a QueueBrowser outside the Java virtual machine, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Throws:
   JMSException - if the JMS provider fails to close this browser due to some internal error.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface QueueConnection

All Superinterfaces:
  Connection

All Known Subinterfaces:
  XAQueueConnection

public interface QueueConnection
extends Connection

Implements: Connection
Implemented by: XAQueueConnection

QueueConnection	JMS	QueueConnection
QueueConnection	QueueSession	QueueSession
QueueConnection

QueueConnection / createDurableConnection
Connection	QueueConnection	IllegalStateException

version 1.1 - April 9, 2002
See javax.jms.Connection, javax.jms.ConnectionFactory,
also javax.jms.QueueConnectionFactory

A QueueConnection object is an active connection to a point-to-point JMS provider. A client uses a QueueConnection object to create one or more QueueSession objects for producing and consuming messages.

A QueueConnection can be used to create a QueueSession, from which specialized queue-related objects can be created. A more general, and recommended, approach is to use the Connection object.
The QueueConnection object should be used to support existing code that has already used it.

A QueueConnection cannot be used to create objects specific to the publish/subscribe domain. The createDurableConnectionConsumer method inherits from Connection, but must throw an IllegalStateException if used from QueueConnection.

Version:
   1.1 - April 9, 2002

Author:
   Mark Hapner, Rich Burridge, Kate Stout

See Also:
   Connection, ConnectionFactory, QueueConnectionFactory

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createConnectionConsumer</td>
<td>Creates a connection consumer for this connection (optional operation).</td>
</tr>
<tr>
<td>createQueueSession</td>
<td>Creates a QueueSession object.</td>
</tr>
</tbody>
</table>

#### Methods inherited from interface javax.jms.Connection

- close, createConnectionConsumer, createDurableConnectionConsumer, createSession, getClientID, getExceptionListener, getMetaData, setClientID, setExceptionListener, start, stop

---

### Method Detail

```java
public QueueSession createQueueSession(boolean transacted, int acknowledgeMode) throws JMSException
```
createQueueSession

QueueSession createQueueSession(boolean transacted, int acknowledgeMode) throws JMSException

Creates a QueueSession object.

Parameters:
- transacted - indicates whether the session is transacted
- acknowledgeMode - indicates whether the consumer or the client will acknowledge any messages it receives; ignored if the session is transacted. Legal values are Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, and Session.DUPS_OK_ACKNOWLEDGE.

Returns:
a newly created queue session

Throws:
- JMSException - if the QueueConnection object fails to create a session due to some internal error or lack of support for the specific transaction and acknowledgement mode.

See Also:
- Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, Session.DUPS_OK_ACKNOWLEDGE
createConnectionConsumer(Queue queue, String messageSelector, ServerSessionPool sessionPool, int maxMessages) throws JMSException

queue
messageSelector null
sessionPool
maxMessages
return

Throws JMSException: QueueConnection
messageSelector

Throws InvalidDestinationException:

Throws InvalidSelectorException:

See also javax.jms.ConnectionConsumer

createConnectionConsumer

ConnectionConsumer createConnectionConsumer(Queue queue,
String messageSelector,
ServerSessionPool sessionPool,
int maxMessages) throws JMSException

Creates a connection consumer for this connection (optional operation). This is an expert facility not used by regular JMS clients.

Parameters:
queue - the queue to access
messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
sessionPool - the server session pool to associate with this connection consumer
maxMessages - the maximum number of messages that can be assigned to a server session at one time

Returns:
the connection consumer

Throws:

JMSException - if the QueueConnection object fails to create a connection consumer due to some internal error or invalid arguments for sessionPool and messageSelector.

InvalidDestinationException - if an invalid queue is specified.

InvalidSelectorException - if the message selector is invalid.

See Also:

ConnectionConsumer
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>INDEX</th>
<th>Help</th>
</tr>
</thead>
</table>
javax.jms Interface QueueConnectionFactory

All Superinterfaces:  
ConnectionFactory

All Known Subinterfaces:  
XAQueueConnectionFactory

public interface QueueConnectionFactory
extends ConnectionFactory

Implements: ConnectionFactory
Implemented by: XAQueueConnectionFactory

QueueConnectionFactory JMS QueueConnection
QueueConnectionFactory QueueConnection QueueConnectionFactory
QueueConnectionFactory

version 1.1 - February 2, 2002
See also javax.jms.ConnectionFactory

A client uses a QueueConnectionFactory object to create QueueConnection objects with a point-to-point JMS provider.

QueueConnectionFactory can be used to create a QueueConnection, from which specialized queue-related objects can be created. A more general, and recommended, approach is to use the ConnectionFactory object.

The QueueConnectionFactory object can be used to support existing code that already uses it.

Version:
Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueConnection</td>
<td><code>createQueueConnection()</code></td>
<td>Creates a queue connection with the default user identity.</td>
</tr>
<tr>
<td>QueueConnection</td>
<td><code>createQueueConnection(String userName, String password)</code></td>
<td>Creates a queue connection with the specified user identity.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.jms.ConnectionFactory`:
- `createConnection`
- `createConnection`

Method Detail

```java
public QueueConnection createQueueConnection() throws JMSException
```

```java
return Connection
```

Throws:
- `JMSException`: JMS
- `JMSSecurityException`: JMSSecurityException
Creates a queue connection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

Returns:
a newly created queue connection

Throws:

- JMSException - if the JMS provider fails to create the queue connection due to some internal error.
- JMSSecurityException - if client authentication fails due to an invalid user name or password.

```java
public QueueConnection createQueueConnection(String userName, String password) throws JMSException
```

**createQueueConnection**

**Parameters:**
- userName - the caller's user name
- password - the caller's password

**Returns:**
a newly created queue connection
Throws:

- **JMSException** - if the JMS provider fails to create the queue connection due to some internal error.
- **JMSSecurityException** - if client authentication fails due to an invalid user name or password.
public interface QueueReceiver

extends MessageConsumer

Implements: MessageConsumer

A client uses a QueueReceiver object to receive messages that have been delivered to a queue.

Although it is possible to have multiple QueueReceivers for the same queue, the JMS API does not define how messages are distributed between the QueueReceivers.

If a QueueReceiver specifies a message selector, the messages that are not selected remain on the queue. By definition, a message selector
allows a QueueReceiver to skip messages. This means that when the skipped messages are eventually read, the total ordering of the reads does not retain the partial order defined by each message producer. Only QueueReceiver instances without a message selector will read messages in message producer order.

Creating a MessageConsumer provides the same features as creating a QueueReceiver. A MessageConsumer object is recommended for creating new code. The QueueReceiver is provided to support existing code.

**Version:**
1.1 February 2, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
Session.createConsumer(Destination, String), Session.createConsumer(Destination), QueueSession.createReceiver(Queue, String), QueueSession.createReceiver(Queue), MessageConsumer

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getQueue()</code></td>
<td>Gets the Queue associated with this queue receiver.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface javax.jms.MessageConsumer

- close, getMessageListener, getMessageSelector, receive, receive, receiveNoWait, setMessageListener

### Method Detail

```java
public Queue getQueue() throws JMSException
```

Return Type: Queue

Throws: JMSException: JMS
getQueue

Queue getQueue() throws JMSException

Gets the Queue associated with this queue receiver.

Returns:
this receiver's Queue

Throws:
JMSException - if the JMS provider fails to get the queue for this queue receiver due to some internal error.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jms Class QueueRequestor

javax.lang.Object
  ↓ javax.jms.QueueRequestor

public class QueueRequestor
extends Object

QueueRequestor
QueueRequestor QueueSession Queue TemporaryQueue request

/JMS

version 1.0 - 8 July 1998

See also javax.jms.TopicRequestor

The QueueRequestor helper class simplifies making service requests.

The QueueRequestor constructor is given a non-transacted QueueSession and a destination Queue. It creates a TemporaryQueue for the responses and provides a request method that sends the request message and waits for its reply.

This is a basic request/reply abstraction that should be sufficient for most uses. JMS providers and clients are free to create more sophisticated versions.

Version: 1.0 - 8 July 1998

Author: Mark Hapner, Rich Burridge

See Also: TopicRequestor
Constructor Summary

QueueRequestor(QueueSession session, Queue queue)
Constructor for the QueueRequestor class.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void close()</td>
<td>Closes the QueueRequestor and its session.</td>
</tr>
<tr>
<td>Message request(Message message)</td>
<td>Sends a request and waits for a reply.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public QueueRequestor(QueueSession session, Queue queue) throws JMSException

QueueRequestor

session AUTO_ACKNOWLEDGE DUPS_OK_ACKNOWLEDGE
session QueueSession
queue /

Throws JMSException: JMS
Throws InvalidDestinationException: QueueRequesto

QueueRequestor

public QueueRequestor(QueueSession session,
Constructor for the QueueRequestor class.

This implementation assumes the session parameter to be non-transacted, with a delivery mode of either AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE.

Parameters:
- session: the QueueSession the queue belongs to
- queue: the queue to perform the request/reply call on

Throws:
- JMSException: if the JMS provider fails to create the QueueRequestor due to some internal error.
- InvalidDestinationException: if an invalid queue is specified.

### Method Detail

**public Message request(Message message) throws JMSException**

```java
JMSReplyTo message
return
```

Throws: JMSException: JMS

Sends a request and waits for a reply. The temporary queue is used for the JMSReplyTo destination, and only one reply per request is expected.

Parameters:
- message: the message to send
Returns:
the reply message

Throws:

- [JMSException](#) - if the JMS provider fails to complete the request due to some internal error.

---

public void close() throws [JMSException](#)

QueueRequestor

---

QueueRequestor Java

QueueRequestor QueueSession

Throws [JMSException](#): JMS

---

close

close

public void close()

Throws [JMSException](#)

Closes the QueueRequestor and its session.

Since a provider may allocate some resources on behalf of a QueueRequestor outside the Java virtual machine, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Note that this method closes the QueueSession object passed to the QueueRequestor constructor.

Throws:

- [JMSException](#) - if the JMS provider fails to close the QueueRequestor due to some internal error.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface QueueSender

All Superinterfaces:
   MessageProducer

public interface QueueSender
extends MessageProducer

Implements: MessageProducer

QueueSender

Queue QueueSender QueueSender send
java.lang.UnsupportedOperationException

Queue QueueSender Queue send
java.lang.UnsupportedOperationException

send send

JMSDestination JMSDeliveryMode JMSExpiration JMSPriority JMSMessageID
JMSTimeStamp send
MessageProducer.setDisableMessageID
MessageProducer.setDisableMessageTimestamp send
JMSMessageID JMSTimeStamp

MessageProducer QueueSender MessageProducer

version 1.1 - February 2, 2002

See javax.jms.MessageProducer,
also createProducer(Destination), createSender(Queue)
A client uses a QueueSender object to send messages to a queue.

Normally, the queue is specified when a QueueSender is created. In this case, an attempt to use the send methods for an unidentified QueueSender will throw a java.lang.UnsupportedOperationException.

If the QueueSender is created with an unidentified Queue, an attempt to use the send methods that assume that the Queue has been identified will throw a java.lang.UnsupportedOperationException.

During the execution of its send method, a message must not be changed by other threads within the client. If the message is modified, the result of the send is undefined.

After sending a message, a client may retain and modify it without affecting the message that has been sent. The same message object may be sent multiple times.

The following message headers are set as part of sending a message: JMSDestination, JMSDeliveryMode, JMSExpiration, JMSPriority, JMSMessageID and JMSTimeStamp. When the message is sent, the values of these headers are ignored. After the completion of the send, the headers hold the values specified by the method sending the message. It is possible for the send method not to set JMSMessageID and JMSTimeStamp if the setting of these headers is explicitly disabled by the MessageProducer.setDisableMessageID or MessageProducer.setDisableMessageTimestamp method.

Creating a MessageProducer provides the same features as creating a QueueSender. A MessageProducer object is recommended when creating new code. The QueueSender is provided to support existing code.

Version:
   1.1 - February 2, 2002
Author:
   Mark Hapner, Rich Burridge, Kate Stout
See Also:
   MessageProducer, Session.createProducer(Destination), QueueSession.createSender(Queue)
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Queue</strong> <code>getQueue()</code></td>
<td>Gets the queue associated with this QueueSender.</td>
</tr>
<tr>
<td><code>send(Message message)</code></td>
<td>Sends a message to the queue.</td>
</tr>
<tr>
<td><code>send(Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Sends a message to the queue, specifying delivery mode, priority, and time to live.</td>
</tr>
<tr>
<td><code>send(Queue queue, Message message)</code></td>
<td>Sends a message to a queue for an unidentified message producer.</td>
</tr>
<tr>
<td><code>send(Queue queue, Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Sends a message to a queue for an unidentified message producer, specifying delivery mode, priority and time to live.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.jms.MessageProducer`:
- `close`, `getDeliveryMode`, `getDestination`, `getDisableMessageID`, `getDisableMessageTimestamp`, `getPriority`, `getTimeToLive`, `send`, `send`, `setDeliveryMode`, `setDisableMessageID`, `setDisableMessageTimestamp`, `setPriority`, `setTimeToLive`

## Method Detail

```java
public Queue getQueue() throws JMSException

    QueueSender

    return

    Throws  JMSException: JMS
```

**getQueue**

*Queue* `getQueue()`
throws JMSException

Gets the queue associated with this QueueSender.

**Returns:**
this sender's queue

**Throws:**
JMSException - if the JMS provider fails to get the queue for this QueueSender due to some internal error.

```java
public void send(Message message) throws JMSException
    QueueSender
message

Throws JMSException: JMS
Throws MessageFormatException: 
Throws InvalidDestinationException:
Throws UnsupportedOperationException:
See also getDeliveryMode(), getTimeToLive(), getPriority()
```

send

```java
void send(Message message)
    throws JMSException
```

Sends a message to the queue. Uses the QueueSender's default delivery mode, priority, and time to live.

**Specified by:**
send in interface MessageProducer

**Parameters:**
message - the message to send

**Throws:**
JMSException - if the JMS provider fails to send the message due to some internal error.
MessageFormatException - if an invalid message is specified.
InvalidDestinationException - if a client uses this method with a QueueSender with an invalid queue.
UnsupportedOperationException - if a client uses this method with a QueueSender that did not specify a queue at creation time.

See Also:

MessageProducer.getDeliveryMode(),
MessageProducer.getTimeToLive(),
MessageProducer.getPriority()

public void send(Message message, int deliveryMode, int priority, long timeToLive) throws JMSException

message
deliveryMode
priority
timeToLive

Throws JMSException: JMS
Throws MessageFormatException:
Throws InvalidDestinationException: Queue
Throws UnsupportedOperationException:

send

void send(Message message,
    int deliveryMode,
    int priority,
    long timeToLive)
    throws JMSException

Sends a message to the queue, specifying delivery mode, priority, and time to live.

Specified by:
    send in interface MessageProducer

Parameters:
    message - the message to send
deliveryMode - the delivery mode to use
priority - the priority for this message
timeToLive - the message's lifetime (in milliseconds)
Throws:

- `JMSException` - if the JMS provider fails to send the message due to some internal error.
- `MessageFormatException` - if an invalid message is specified.
- `InvalidDestinationException` - if a client uses this method with a `QueueSender` with an invalid queue.
- `UnsupportedOperationException` - if a client uses this method with a `QueueSender` that did not specify a queue at creation time.

See Also:

- `Session.createProducer(javax.jms.Destination)`

```java
public void send(Queue queue, Message message) throws JMSException
```

Sends a message to a queue for an unidentified message producer. Uses the `QueueSender`'s default delivery mode, priority, and time to live.

Typically, a message producer is assigned a queue at creation time; however, the JMS API also supports unidentified message
producers, which require that the queue be supplied every time a message is sent.

**Parameters:**
- `queue` - the queue to send this message to
- `message` - the message to send

**Throws:**
- `JMSException` - if the JMS provider fails to send the message due to some internal error.
- `MessageFormatException` - if an invalid message is specified.
- `InvalidDestinationException` - if a client uses this method with an invalid queue.

**See Also:**
- `MessageProducer.getDeliveryMode()`,
- `MessageProducer.getTimeToLive()`,
- `MessageProducer.getPriority()`

```java
public void send(Queue queue, Message message, int deliveryMode, int priority, long timeToLive) throws JMSException
```

**JMS API**

```java
    queue
    message
    deliveryMode
    priority
    timeToLive
```

**Throws**
- `JMSException`: JMS
- `MessageFormatException`: JMS
- `InvalidDestinationException`: JMS

send
void send(Queue queue,
   Message message,
   int deliveryMode,
   int priority,
   long timeToLive)
   throws JMSEException

Sends a message to a queue for an unidentified message producer, specifying delivery mode, priority and time to live.

Typically, a message producer is assigned a queue at creation time; however, the JMS API also supports unidentified message producers, which require that the queue be supplied every time a message is sent.

Parameters:
   queue - the queue to send this message to
   message - the message to send
   deliveryMode - the delivery mode to use
   priority - the priority for this message
   timeToLive - the message's lifetime (in milliseconds)

Throws:
   JMSEException - if the JMS provider fails to send the message due to some internal error.
   MessageFormatException - if an invalid message is specified.
   InvalidDestinationException - if a client uses this method with an invalid queue.

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES  NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface QueueSession

All Superinterfaces:
   Runnable, Session

public interface QueueSession
extends Session

Implements: Session
QueueSession QueueReceiver QueueSender QueueBrowser
TemporaryQueue QueueSession

QueueSession Session QueueSession Session
QueueSession / Session QueueSession
IllegalStateException

  • createDurableSubscriber
  • createTemporaryTopic
  • createTopic
  • unsubscribe

version 1.1 - April 2, 2002
See javax.jms.Session, createQueueSession(boolean, int),
getQueueSession()
redelivered when a consumer next accesses the queue.

A QueueSession is used for creating Point-to-Point specific objects. In general, use the Session object. The QueueSession is used to support existing code. Using the Session object simplifies the programming model, and allows transactions to be used across the two messaging domains.

A QueueSession cannot be used to create objects specific to the publish/subscribe domain. The following methods inherit from Session, but must throw an IllegalStateException if they are used from QueueSession:

- createDurableSubscriber
- createTemporaryTopic
- createTopic
- unsubscribe

Version:
1.1 - April 2, 2002

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
- Session, QueueConnection.createQueueSession(boolean, int),
- XAQueueSession.getQueueSession()

### Field Summary

#### Fields inherited from interface javax.jms.Session

AUTO_ACKNOWLEDGE, CLIENT_ACKNOWLEDGE, DUPS_OK_ACKNOWLEDGE, SESSION_TRANSACTED

### Method Summary

- **createBrowser(Queue queue)**
  Creates a QueueBrowser object to peek at the messages on the specified queue.
<table>
<thead>
<tr>
<th>Class</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueBrowser</td>
<td><code>createBrowser</code> (Queue queue, String messageSelector)</td>
<td>Creates a QueueBrowser object to peek at the messages on the specified queue using a message selector.</td>
</tr>
<tr>
<td>Queue</td>
<td><code>createQueue</code> (String queueName)</td>
<td>Creates a queue identity given a queue name.</td>
</tr>
<tr>
<td>QueueReceiver</td>
<td><code>createReceiver</code> (Queue queue)</td>
<td>Creates a QueueReceiver object to receive messages from the specified queue.</td>
</tr>
<tr>
<td>QueueReceiver</td>
<td><code>createReceiver</code> (Queue queue, String messageSelector)</td>
<td>Creates a QueueReceiver object to receive messages from the specified queue using a message selector.</td>
</tr>
<tr>
<td>QueueSender</td>
<td><code>createSender</code> (Queue queue)</td>
<td>Creates a QueueSender object to send messages to the specified queue.</td>
</tr>
<tr>
<td>TemporaryQueue</td>
<td><code>createTemporaryQueue</code> ()</td>
<td>Creates a TemporaryQueue object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.Session:
- close
- commit
- createBytesMessage
- createConsumer
- createConsumer
- createDurableSubscriber
- createDurableSubscriber
- createMapMessage
- createMessage
- createObjectMessage
- createProducer
- createStreamMessage
- createTemporaryTopic
- createTextMessage
- createTopic
- getAcknowledgeMode
- getMessageListener
- getTransacted
- recover
- rollback
- run
- setMessageListener
- unsubscribe

## Method Detail

```java
class Queue

public Queue createQueue(String queueName) throws JMSException
```

Queue
createQueue

**createQueue(String queueName)**

Throws **JMSException**

Creates a queue identity given a queue name.

This facility is provided for the rare cases where clients need to dynamically manipulate queue identity. It allows the creation of a queue identity with a provider-specific name. Clients that depend on this ability are not portable.

Note that this method is not for creating the physical queue. The physical creation of queues is an administrative task and is not to be initiated by the JMS API. The one exception is the creation of temporary queues, which is accomplished with the createTemporaryQueue method.

**Specified by:**
createQueue in interface Session

**Parameters:**

queueName - the name of this Queue

**Returns:**

a Queue with the given name

**Throws:**

**JMSException** - if the session fails to create a queue due to some internal error.

public **QueueReceiver** createReceiver(Queue queue)

throws **JMSException**
createReceiver

**QueueReceiver createReceiver(Queue queue)**

Throws **JMSException**: 
Throws **InvalidDestinationException**: 

Creates a **QueueReceiver** object to receive messages from the specified queue.

**Parameters:**
- **queue** - the Queue to access

**Throws:**
- **JMSException** - if the session fails to create a receiver due to some internal error.
- **InvalidDestinationException** - if an invalid queue is specified.

---

**public QueueReceiver createReceiver(Queue queue, String messageSelector)**

Throws **JMSException**

**QueueReceiver**

- **queue**
- **null**

**Throws:**
- **JMSException**: 
- **InvalidDestinationException**: 
- **InvalidSelectorException**: 

**createReceiver**

**QueueReceiver createReceiver(Queue queue, String messageSelector)**

Throws **JMSException**
Creates a QueueReceiver object to receive messages from the specified queue using a message selector.

**Parameters:**
- `queue` - the Queue to access
- `messageSelector` - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.

**Throws:**
- `JMSException` - if the session fails to create a receiver due to some internal error.
- `InvalidDestinationException` - if an invalid queue is specified.
- `InvalidSelectorException` - if the message selector is invalid.

```java
public QueueSender createSender(Queue queue) throws JMSException
{
  QueueSender queue
  Queue null
}

createSender
```

```java
QueueSender createSender(Queue queue) throws JMSException
{
  Creates a QueueSender object to send messages to the specified queue.

  **Parameters:**
  - `queue` - the Queue to access, or null if this is an unidentified producer

  **Throws:**
  - `JMSException` - if the session fails to create a sender due to some internal error.
```
InvalidDestinationException - if an invalid queue is specified.

**public QueueBrowser createBrowser(Queue queue) throws JMSException**

QueueBrowser

queue: Queue

Throws: JMSException:

Throws: InvalidDestinationException:

createBrowser

QueueBrowser createBrowser(Queue queue)

throws JMSException

Creates a QueueBrowser object to peek at the messages on the specified queue.

Specified by:
createBrowser in interface Session

Parameters:

queue - the Queue to access

Throws:

JMSException - if the session fails to create a browser due to some internal error.

InvalidDestinationException - if an invalid queue is specified.

**public QueueBrowser createBrowser(Queue queue, String messageSelector) throws JMSException**

QueueBrowser

queue: Queue

messageSelector: null

Throws: JMSException:

Throws: InvalidDestinationException:

Throws: InvalidSelectorException:
**createBrowser**

```
QueueBrowser createBrowser(Queue queue,
                           String messageSelector)
throws JMSException
```

Creates a `QueueBrowser` object to peek at the messages on the specified queue using a message selector.

**Specified by:**
`createBrowser` in interface `Session`

**Parameters:**
- `queue` - the Queue to access
- `messageSelector` - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.

**Throws:**
- `JMSException` - if the session fails to create a browser due to some internal error.
- `InvalidDestinationException` - if an invalid queue is specified.
- `InvalidSelectorException` - if the message selector is invalid.

**public TemporaryQueue createTemporaryQueue() throws JMSException**

```
TemporaryQueue createTemporaryQueue()
```

**Throws** `JMSException`: 

**createTemporaryQueue**

```
TemporaryQueue createTemporaryQueue()
throws JMSException
```

Creates a `TemporaryQueue` object. Its lifetime will be that of the
QueueConnection unless it is deleted earlier.

Specified by:

createTemporaryQueue in interface Session

Returns:

a temporary queue identity

Throws:

JMSException - if the session fails to create a temporary queue due to some internal error.
javax.mail  **Class Quota**

```java
java.lang.Object
   | javax.mail.Quota
```

public class **Quota** extends **Object**

**Inner classes:** **Quota.Resource**

(quotas root) Quota.Resource

RFC 2087

<table>
<thead>
<tr>
<th>since</th>
<th>JavaMail 1.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td>1.7, 07/05/04</td>
</tr>
</tbody>
</table>

This class represents a set of quotas for a given quota root. Each quota root has a set of resources, represented by the **Quota.Resource** class. Each resource has a name (for example, "STORAGE"), a current usage, and a usage limit. See RFC 2087.

**Since:** JavaMail 1.4
**Version:** 1.7, 07/05/04
**Author:** Bill Shannon

---

**Nested Class Summary**

<table>
<thead>
<tr>
<th>static class</th>
<th><strong>Quota.Resource</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>An individual resource in a quota root.</td>
</tr>
</tbody>
</table>

**Field Summary**
The name of the quota root.

The set of resources associated with this quota root.

**Constructor Summary**

```java
Quota(String quotaRoot)
```

Create a Quota object for the named quotaroot with no associated resources.

**Method Summary**

```java
void setResourceLimit(String name, long limit)
```

Set a resource limit for this quota root.

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Field Detail**

```java
public String quotaRoot
```

The name of the quota root.

```java
resources
```
public Quota.Resource[] resources

The set of resources associated with this quota root.

### Constructor Detail

```java
public Quota(String quotaRoot)
```

Create a Quota object for the named quotaroot with no associated resources.

**Parameters:**
- `quotaRoot` - the name of the quota root

### Method Detail

```java
public void setResourceLimit(String name, long limit)
```

Set a resource limit for this quota root.
Parameters:
  name - the name of the resource
  limit - the resource limit

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail Class Quota.Resource

java.lang.Object
   \_ javax.mail.Quota.Resource

Enclosing class:
   Quota

public static class Quota.Resource
extends Object

Contained within: Quota

since JavaMail 1.4

An individual resource in a quota root.

Since:
   JavaMail 1.4

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long</td>
<td>limit</td>
<td>The usage limit for the resource.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>The name of the resource.</td>
</tr>
<tr>
<td>long</td>
<td>usage</td>
<td>The current usage of the resource.</td>
</tr>
</tbody>
</table>

Constructor Summary

Quota.Resource(String name, long usage, long limit)
Construct a Resource object with the given name, usage, and limit.

Method Summary

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

name

public String name

The name of the resource.

usage

public long usage

The current usage of the resource.

limit

public long limit

The usage limit for the resource.
public Quota.Resource(String name, long usage, long limit)

Construct a Resource object with the given name, usage, and limit.

Parameters:
- name - the resource name
- usage - the current usage of the resource
- limit - the usage limit for the resource
javax.mail Interface QuotaAwareStore

public interface QuotaAwareStore

Store 2087 getQuota setQuota IMAP QUOTA

since JavaMail 1.4 en

An interface implemented by Stores that support quotas. The getQuota and setQuota methods support the quota model defined by the IMAP QUOTA extension. Refer to RFC 2087 for more information.

Since:
JavaMail 1.4

Method Summary

<table>
<thead>
<tr>
<th>Quota[]</th>
<th>getQuota(String root)</th>
<th>Get the quotas for the named quota root.</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>setQuota(Quota quota)</td>
<td>Set the quotas for the quota root specified in the quota argument.</td>
</tr>
</tbody>
</table>

Method Detail

public Quota[] getQuota(String root) throws MessagingException
getQuota

Quota[] getQuota(String root) throws MessagingException

Get the quotas for the named quota root. Quotas are controlled on the basis of a quota root, not (necessarily) a folder. The relationship between folders and quota roots depends on the server. Some servers might implement a single quota root for all folders owned by a user. Other servers might implement a separate quota root for each folder. A single folder can even have multiple quota roots, perhaps controlling quotas for different resources.

Parameters:
root - the name of the quota root

Returns:
array of Quota objects

Throws:
MessagingException - if the server doesn't support the QUOTA extension

public void setQuota(Quota quota) throws MessagingException

setQuota

void setQuota(Quota quota) throws MessagingException
Set the quotas for the quota root specified in the quota argument. Typically this will be one of the quota roots obtained from the getQuota method, but it need not be.

**Parameters:**
- `quota` - the quota to set

**Throws:**
- `MessagingException` - if the server doesn't support the QUOTA extension
javax.management.j2ee.statistics Interface RangeStatistic

All Superinterfaces:
Statistic

All Known Subinterfaces:
BoundedRangeStatistic

public interface RangeStatistic
extends Statistic

Implements: Statistic
Implemented by: BoundedRangeStatistic

Specifies standard measurements of the lowest and highest values an attribute has held as well as its current value.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>long getCurrent()</td>
<td>The current value of this attribute.</td>
</tr>
<tr>
<td>long getHighWaterMark()</td>
<td>The highest value this attribute has held since the beginning of the measurement.</td>
</tr>
<tr>
<td>long getLowWaterMark()</td>
<td>The lowest value this attribute has held since the beginning of the measurement.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.Statistic
getDescription, getLastSampleTime, getName, getStartTime, getUnit
public long getHighWaterMark()

getHighWaterMark

long getHighWaterMark()

The highest value this attribute has held since the beginning of the measurement.

public long getLowWaterMark()

getLowWaterMark

long getLowWaterMark()

The lowest value this attribute has held since the beginning of the measurement.

public long getCurrent()

getCurrent

long getCurrent()

The current value of this attribute.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail  **Class ReadOnlyFolderException**

*java.lang.Object*  
  └ *java.lang.Throwable*  
    └ *java.lang.Exception*  
      └ *javax.mail.MessagingException*  
        └ *javax.mail.ReadOnlyFolderException*

**All Implemented Interfaces:**  
  *Serializable*

---

**public class**  *ReadOnlyFolderException*

**Extends:**  *Throwable*  >  *Exception*  >  *MessagingException*

**getException()**

This exception is thrown when an attempt is made to open a folder read-write access when the folder is marked read-only.

The `getMessage()` method returns more detailed information about the error that caused this exception.

**Author:**  
  Jim Glennon

**See Also:**  
  *Serialized Form*

---

**Constructor Summary**

| *ReadOnlyFolderException*(*Folder* folder) |  |
Constructs a MessagingException with the specified folder.

**ReadOnlyFolderException**(Folder folder, String message)

Constructs a MessagingException with the specified folder and the specified detail message.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getFolder</td>
<td>Returns the dead Folder object.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.**MessagingException**

getCause, getNextException, setNextException, toString

Methods inherited from class java.lang.**Throwable**

fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace

Methods inherited from class java.lang.**Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail

### public ReadOnlyFolderException(Folder folder)

**MessagingException**

- **folder**
  - Folder
- **since**
  - JavaMail 1.2

### public ReadOnlyFolderException(Folder folder)

**ReadOnlyFolderException**
Constructs a MessagingException with the specified folder.

**Parameters:**
- folder - the Folder

**Since:**
JavaMail 1.2

```java
public ReadOnlyFolderException(Folder folder, String message)
```

**MessagingException**

```java
public ReadOnlyFolderException(Folder folder, String message)
```

ReadonlyFolderException

```java
public ReadonlyFolderException(Folder folder, String message)
```

Constructs a MessagingException with the specified folder and the specified detail message.

**Parameters:**
- folder - the Folder
- message - the detailed error message

**Since:**
JavaMail 1.2

### Method Detail

```java
public Folder getFolder()
```

`Folder`

**Since:**
JavaMail 1.2
public Folder getFolder()

Returns the dead Folder object.

Since:
JavaMail 1.2
**javax.mail.search Class ReceivedDateTerm**

**java.lang.Object**
- javax.mail.search.SearchTerm
  - javax.mail.search.ComparisonTerm
    - javax.mail.search.DateTerm
      - javax.mail.search.ReceivedDateTerm

All Implemented Interfaces:
- **Serializable**

public final class `ReceivedDateTerm`

extends `DateTerm`

**Extends:** `SearchTree` > `ComparisonTerm` > `DateTerm`

Message Received

This class implements comparisons for the Message Received date

**Author:**
  Bill Shannon, John Mani

**See Also:**
- `Serialized Form`

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.<code>DateTerm</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>date</code></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.<code>ComparisonTerm</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>comparison</code>, <code>EQ</code>, <code>GE</code>, <code>GT</code>, <code>LE</code>, <code>LT</code>, <code>NE</code></td>
</tr>
</tbody>
</table>
**Constructor Summary**

*ReceivedDateTerm*(int comparison, `Date` `date`)
  Constructor.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td><code>equals(Object obj)</code></td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>match(Message msg)</code></td>
<td>The match method.</td>
</tr>
</tbody>
</table>

**Methods inherited from class `javax.mail.search.DateTerm`**

`getComparison`, `getDate`, `hashCode`, `match`

**Methods inherited from class `java.lang.Object`**

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`

**Constructor Detail**

```java
public ReceivedDateTerm(int comparison, java.util.Date `date`)
```

```
  comparison
  `date`
  Comparison
```

**ReceivedDateTerm**

```java
public ReceivedDateTerm(int comparison, `Date` `date`)
```

Constructor.
Parameters:
  comparison - the Comparison type
date - the date to be compared

### Method Detail

**public boolean match(Message msg)**

```java
    msg  Message
    return true false
```

**match**

public boolean match(Message msg)

The match method.

**Specified by:**

match in class SearchTerm

**Parameters:**

msg - the date comparator is applied to this Message's sent date

**Returns:**

true if the comparison succeeds, otherwise false

---

**public boolean equals(Object obj)**

**equals**

public boolean equals(Object obj)

Equality comparison.

**Overrides:**
equals in class DateTerm

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.search  Class RecipientStringTerm

java.lang.Object  
  | javax.mail.search.SearchTerm
  |   | javax.mail.search.StringTerm
  |   | javax.mail.search.AddressStringTerm
  |   | javax.mail.search.RecipientStringTerm

All Implemented Interfaces:
  Serializable

public final class RecipientStringTerm

extends AddressStringTerm

Extends: SearchTerm > StringTerm > AddressStringTerm

Recipient Address

RecipientTerm Address

since JavaMail 1.1

This class implements string comparisons for the Recipient Address headers.

Note that this class differs from the RecipientTerm class in that this class does comparisons on address strings rather than Address objects. The string comparisons are case-insensitive.

Since:
  JavaMail 1.1

See Also:
  Serialized Form
### Field Summary

Fields inherited from class `javax.mail.search.StringTerm`
- `ignoreCase`, `pattern`

### Constructor Summary


### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean equals(Object obj)</code></td>
<td>Equality comparison.</td>
</tr>
<tr>
<td><code>Message.RecipientType getRecipientType()</code></td>
<td>Return the type of recipient to match with.</td>
</tr>
<tr>
<td><code>int hashCode()</code></td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td><code>boolean match(Message msg)</code></td>
<td>Check whether the address specified in the constructor is a substring of the recipient address of this Message.</td>
</tr>
</tbody>
</table>

Methods inherited from class `javax.mail.search.AddressStringTerm`
- `match`

Methods inherited from class `javax.mail.search.StringTerm`
- `getIgnoreCase`, `getPattern`, `match`

Methods inherited from class `java.lang.Object`
- `clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public RecipientStringTerm(Message.RecipientType type, String pattern)

Constructor.

Parameters:
  type - the recipient type
  pattern - the address pattern to be compared.

public Message.RecipientType getRecipientType()

getRecipientType

public Message.RecipientType getRecipientType()

Return the type of recipient to match with.

public boolean match(Message msg)
match

public boolean match(Message msg)

Check whether the address specified in the constructor is a substring of the recipient address of this Message.

Specified by:
match in class searchTerm
Parameters:
msg - The comparison is applied to this Message's recipient address.
Returns:
true if the match succeeds, otherwise false.

equals

public boolean equals(Object obj)

Equals

public boolean equals(Object obj)

Equality comparison.

Overrides:
equals in class AddressStringTerm

hashCode

public int hashCode()

hashCode
public int hashCode()

Compute a hashCode for this object.

Overrides:
hashCode in class StringTerm
javax.mail.search    Class RecipientTerm

java.lang.Object
  --- javax.mail.search.SearchTerm
       --- javax.mail.search.AddressTerm
            --- javax.mail.search.RecipientTerm

All Implemented Interfaces:
    Serializable

public final class RecipientTerm
    extends AddressTerm

Extends: searchTerm > AddressTerm

Recipient Address

This class implements comparisons for the Recipient Address headers.

Author:
    Bill Shannon, John Mani

See Also:
    Serialized Form

---

Field Summary

<table>
<thead>
<tr>
<th>protected</th>
<th>type</th>
<th>The recipient type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message.RecipientType</td>
<td>type</td>
<td>The recipient type.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.mail.search.AddressTerm
address

Constructor Summary
RecipientTerm(Message.RecipientType type, Address address)

Constructor.

Method Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th>equals(Object obj)</th>
<th>Equality comparison.</th>
</tr>
</thead>
<tbody>
<tr>
<td>getRecipientType()</td>
<td>Return the type of recipient to match with.</td>
<td></td>
</tr>
<tr>
<td>int</td>
<td>hashCode()</td>
<td>Compute a hashCode for this object.</td>
</tr>
<tr>
<td>boolean</td>
<td>match(Message msg)</td>
<td>The match method.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.mail.search.AddressTerm
getAddress, match

Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, toString, wait, wait

Field Detail

protected Message.RecipientType type

The recipient type.

Constructor Detail

public RecipientTerm(Message.RecipientType type,
Address(address)

type
address

RecipientTerm

public RecipientTerm(Message.RecipientType type,
        Address address)

Constructor.

Parameters:
    type - the recipient type
    address - the address to match for

Method Detail

public Message.RecipientType getRecipientType()

getRecipientType

public Message.RecipientType getRecipientType()

    Return the type of recipient to match with.

public boolean match(Message msg)

    msg        Message
    return     true false
**match**

```java
public boolean match(Message msg)
```

The match method.

**Specified by:**

match in class searchTerm

**Parameters:**

msg - The address match is applied to this Message's recipient address

**Returns:**

true if the match succeeds, otherwise false

---

**public boolean equals(Object obj)**

**equals**

```java
public boolean equals(Object obj)
```

Equality comparison.

**Overrides:**

equals in class AddressTerm

---

**public int hashCode()**

**hashCode**

```java
public int hashCode()
```

Compute a hashCode for this object.
Overrides:

hashCode in class AddressTerm
javax.resource.cci Interface Record

All Superinterfaces:
  Cloneable, Serializable

All Known Subinterfaces:
  IndexedRecord, MappedRecord, ResultSet

public interface Record
extends Cloneable, Serializable

Implements: Cloneable, java.io.Serializable
Implemented by: IndexedRecord, MappedRecord, ResultSet

javax.resource.cci.Record Interaction execute

Record

- MappedRecord- Collection
- IndexedRecord Collection
- JavaBean EIS ERP
- java.util.Map
- java.sql.ResultSet
- javax.resource.cci.ResultSet

MappedRecord IndexedRecord Record MappedRecord
IndexedRecord Java MappedRecord IndexedRecord
version 0.8
See also  javax.resource.cci.Interaction, java.sql.ResultSet

The javax.resource.cci.Record interface is the base interface for the representation of an input or output to the execute methods defined on an Interaction.

The Record interface can be extended to form a one of the following
representations:

- **MappedRecord**: A key-value pair based collection represents a record. This interface is based on the `java.util.Map`
- **IndexedRecord**: An ordered and indexed collection represents a record. This interface is based on the `java.util.List`.
- **JavaBean based representation of an EIS abstraction**: An example is a custom record generated to represent a purchase order in an ERP system.
- **javax.resource.cci.ResultSet**: This interface extends both `java.sql.ResultSet` and `javax.resource.cci.Record`. A ResultSet represents tabular data.

A MappedRecord or IndexedRecord can contain another Record. This means that MappedRecord and IndexedRecord can be used to create a hierarchical structure of any arbitrary depth. A basic Java type is used as the leaf element of a hierarchical structure represented by a MappedRecord or IndexedRecord.

**Version**: 0.8

**Author**: Rahul Sharma

**See Also**: Interaction, ResultSet

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone()</code></td>
<td>Creates and returns a copy of this object.</td>
</tr>
<tr>
<td><code>equals(Object other)</code></td>
<td>Check if this instance is equal to another Record.</td>
</tr>
<tr>
<td><code>getRecordName()</code></td>
<td>Gets the name of the Record.</td>
</tr>
<tr>
<td><code>getRecordShortDescription()</code></td>
<td>Gets a short description string for the Record.</td>
</tr>
<tr>
<td><code>hashCode()</code></td>
<td></td>
</tr>
</tbody>
</table>
Returns the hash code for the Record instance.

void setRecordName(String name)
Sets the name of the Record.

void setRecordShortDescription(String description)
Sets a short description string for the Record.

Method Detail

public String getRecordName()
Record
return
Record

getRecordName

String getRecordName()

Gets the name of the Record.

Returns:
String representing name of the Record

public void setRecordName(String name)
Record
name
Record

setRecordName

void setRecordName(String name)

Sets the name of the Record.

Parameters:
public void setRecordShortDescription(String description)

Sets a short description string for the Record. This property is used primarily by application development tools.

Parameters:

description - Description of the Record

public String getRecordShortDescription()

Gets a short description string for the Record. This property is used primarily by application development tools.

Returns:

String representing a short description of the Record

public boolean equals(Object other)

name - Name of the Record
equals

boolean equals(Object other)

Check if this instance is equal to another Record.

Overrides: equals in class Object

Returns: true if two instances are equal

public int hashCode()
Record

return

hashCode

int hashCode()

Returns the hash code for the Record instance.

Overrides: hashCode in class Object

Returns: hash code

public Object clone() throws CloneNotSupportedException
“”
return

Throws CloneNotSupportedException: Cloneable clone
**clone**

```java
Object clone()
    throws CloneNotSupportedException
```

Creates and returns a copy of this object. The precise meaning of "copy" may depend on the class of the object.

**Returns:**
a clone of this instance.

**Throws:**
- `CloneNotSupportedException` - If the object's class does not support the Cloneable interface. Subclasses that override the clone method can also throw this exception to indicate that an instance cannot be cloned.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
public interface RecordFactory

RecordFactory MappedRecord IndexedRecord
RecordFactory CCI RecordFactory

since 0.8
See [javax.resource.cci.IndexedRecord](#), [javax.resource.cci.MappedRecord](#)
also [javax.resource.cci.IndexedRecord](#), [javax.resource.cci.MappedRecord](#)

The RecordFactory interface is used for creating MappedRecord and IndexedRecord instances. Note that the RecordFactory is only used for creation of generic record instances. A CCI implementation provides an implementation class for the RecordFactory interface.

Since: 0.8
Author: Rahul Sharma
See Also: IndexedRecord, MappedRecord

<table>
<thead>
<tr>
<th>Class</th>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>IndexedRecord</td>
<td><code>createIndexedRecord(String recordName)</code> Creates a IndexedRecord.</td>
</tr>
<tr>
<td>MappedRecord</td>
<td><code>createMappedRecord(String recordName)</code> Creates a MappedRecord.</td>
</tr>
</tbody>
</table>
public **MappedRecord** createMappedRecord(String recordName) throws **ResourceException**

**MappedRecord** RecordFactory

```java
recordName
return
```

<table>
<thead>
<tr>
<th>Throws</th>
</tr>
</thead>
<tbody>
<tr>
<td>•</td>
</tr>
</tbody>
</table>

**Throws** ```java
NotFoundException: MappedRecord
```

**createMappedRecord**

```java
MappedRecord createMappedRecord(String recordName)
throws ResourceException
```

Creates a MappedRecord. The method takes the name of the record that is to be created by the RecordFactory. The name of the record acts as a pointer to the meta information (stored in the metadata repository) for a specific record type.

**Parameters:**

- **recordName** - Name of the Record

**Returns:**

- MappedRecord

**Throws:**

- **ResourceException** - Failed to create a MappedRecord. Example error cases are:
  - Invalid specification of record name
  - Resource adapter internal error
  - Failed to access metadata repository
- **NotSupportedException** - Operation not supported

```java
public IndexedRecord createIndexedRecord(String
```

---
recordName) throws ResourceException
IndexedRecord
RecordFactory

createIndexedRecord

createIndexedRecord(String recordName) throws ResourceException

Creates a IndexedRecord. The method takes the name of the record that is to be created by the RecordFactory. The name of the record acts as a pointer to the meta information (stored in the metadata repository) for a specific record type.

Parameters:
recordName - Name of the Record

Returns:
IndexedRecord

Throws:
ResourceException - Failed to create an IndexedRecord.
Example error cases are:
- Invalid specification of record name
- Resource adapter internal error
- Failed to access metadata repository

NotSupportedException - Operation not supported
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
| SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLASS | FRAME: NO FRAMES | DETAIL: FIELD | CONSTR | METHOD | PREV CLASS | NEXT CLAS
The Referenceable interface extends the javax.naming.Referenceable interface. It enables support for JNDI Reference mechanism for the registration of the connection factory in the JNDI name space. Note that the implementation and structure of Reference is specific to an application server.

The implementation class for a connection factory interface is required to implement both java.io.Serializable and javax.resource.Referenceable interfaces to support JNDI registration.

Version:
0.9

Author:
Rahul Sharma
### Method Summary

<table>
<thead>
<tr>
<th>void setReference(Reference reference)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the Reference instance.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void setReference(javax.naming.Reference reference)
Reference getReference
javax.naming.Referenceable Reference
```

#### setReference

```java
void setReference(Reference reference)
```

Sets the Reference instance. This method is called by the deployment code to set the Reference that can be later returned by the getReference method (as defined in the javax.naming.Referenceable interface).

**Parameters:**

- reference - A Reference instance

**See Also:**

- `Referenceable.getReference()`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :

**javax.faces.el Class ReferenceSyntaxException**

```java
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.faces.FacesException
                  └ javax.faces.el.EvaluationException
                      └ javax.faces.el.ReferenceSyntaxException
```

**All Implemented Interfaces:**
- Serializable

---

**Deprecated.** *This has been replaced by ELException.*

**Extends:** Throwable > Exception > RuntimeException > FacesException > EvaluationException

```java
public class ReferenceSyntaxException
extends EvaluationException
```

An exception reporting a syntax error in a method binding expression or value binding expression.

**See Also:**
- Serialized Form

---

**Constructor Summary**

```java
ReferenceSyntaxException()
```

**Deprecated.** Construct a new exception with no detail message
or root cause.

ReferenceSyntaxException(String message)

    Deprecated. Construct a new exception with the specified detail message and no root cause.

ReferenceSyntaxException(String message, Throwable cause)

    Deprecated. Construct a new exception with the specified detail message and root cause.

ReferenceSyntaxException(Throwable cause)

    Deprecated. Construct a new exception with the specified root cause.

---

Method Summary

Methods inherited from class javax.faces.FacesException
getCause

Methods inherited from class java.lang.Throwable
fillInStackTrace, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

Constructor Detail

public ReferenceSyntaxException()
ReferenceSyntaxException

public ReferenceSyntaxException()

    Deprecated.

    Construct a new exception with no detail message or root cause.

public ReferenceSyntaxException(String message)

    message

ReferenceSyntaxException

public ReferenceSyntaxException(String message)

    Deprecated.

    Construct a new exception with the specified detail message and no root cause.

    Parameters:
    
    message - The detail message for this exception

public ReferenceSyntaxException(Throwables cause)

    (cause == null ? null : cause.toString())

    cause

ReferenceSyntaxException
public ReferenceSyntaxException(Throwable cause)

Depreciated.

Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

Parameters:
cause - The root cause for this exception

public ReferenceSyntaxException(String message, Throwable cause)

message
cause

ReferenceSyntaxException

public ReferenceSyntaxException(String message, Throwable cause)

Depreciated.

Construct a new exception with the specified detail message and root cause.

Parameters:
message - The detail message for this exception
cause - The root cause for this exception
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.registry.infomodel Interface RegistryEntry

All Superinterfaces:
   ExtensibleObject, RegistryObject, Versionable

All Known Subinterfaces:
   ClassificationScheme, ExtrinsicObject, RegistryPackage, Service

public interface RegistryEntry
   extends RegistryObject, Versionable

Implements: RegistryObject, Versionable
Implemented by: ClassificationScheme, ExtrinsicObject, RegistryPackage, Service

RegistryEntry RegistryObject

The RegistryEntry interface is a base interface for interfaces in the model that require additional metadata beyond what is provided by the RegistryObject interface. A few interfaces in the model represent high level (coarse grain) objects in the registry that require additional metadata such as version information and indication of the stability or volatility of the information.

Author:
   Farrukh S. Najmi

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
</tr>
<tr>
<td>RegistryEntry may change at any time.</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>RegistryEntry may change at any time, however the changes will be backward compatible.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getExpiration()</code></td>
<td>Gets expirationDate attribute of the RegistryEntry within the Registry.</td>
</tr>
<tr>
<td><code>getStability()</code></td>
<td>Gets the stability indicator for the RegistryEntry within the Registry.</td>
</tr>
<tr>
<td><code>getStatus()</code></td>
<td>Gets the life cycle status of the RegistryEntry within the registry.</td>
</tr>
<tr>
<td><code>setExpiration(Date expiration)</code></td>
<td>Sets the expirationDate.</td>
</tr>
<tr>
<td><code>setStability(int stability)</code></td>
<td>Sets the stability indicator for the RegistryEntry.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface javax.xml.registry.infomodel.RegistryObject

- `addAssociation`, `addAssociations`, `addClassification`, `addClassifications`, `addExternalIdentifier`, `addExternalIdentifiers`, `addExteralnallLink`, `addExteralnallLinks`, `getAssociatedObjects`, `getAssociations`, `getAuditTrail`, `getClassifications`, `getDescription`, `getExternalIdentifiers`, `getExteralnallLinks`, `getKey`, `getLifeCycleManager`, `getName`, `getObjectType`, `getRegistryPackages`, `getSubmittingOrganization`, `removeAssociation`, `removeAssociations`, `removeClassification`, `removeClassifications`
### Field Detail

**STATUS_SUBMITTED**

static final int STATUS_SUBMITTED

RegistryEntry has been submitted.

See Also:

[Constant Field Values](#)

**STATUS_APPROVED**

static final int STATUS_APPROVED

RegistryEntry has been submitted and approved.

See Also:
Constant Field Values

STATUS_DEPRECATED

static final int STATUS_DEPRECATED

RegistryEntry has been deprecated.

See Also:
   Constant Field Values

STATUS_WITHDRAWN

static final int STATUS_WITHDRAWN

RegistryEntry has been withdrawn by the submitter.

See Also:
   Constant Field Values

STABILITY_DYNAMIC

static final int STABILITY_DYNAMIC

RegistryEntry may change at any time.

See Also:
   Constant Field Values
STABILITY_DYNAMIC_COMPATIBLE

static final int STABILITY_DYNAMIC_COMPATIBLE

RegistryEntry may change at any time, however the changes will be backward compatible.

See Also:
Constant Field Values

STABILITY_STATIC

static final int STABILITY_STATIC

RegistryEntry will not change.

See Also:
Constant Field Values

Method Detail

public int getStatus() throws JAXRException
RegistryEntry

UnsupportedCapabilityException

See
return
Throws
JAXRException: JAXR

getStatus
public int getStability() throws JAXRException

Registry RegistryEntry

1

UnsupportedCapabilityException

Get the stability indicator for the RegistryEntry within the Registry. The stability indicator is provided by the submitter as an indication of the level of stability for the content.

Capability Level: 1
This method must throw UnsupportedCapabilityException in lower capability levels.
Returns:
the stability indicator as an integer enumeration

Throws:
   JAXRException - If the JAXR provider encounters an internal error

See Also:
   STABILITY_DYNAMIC

public void setStability(int stability) throws JAXRException
RegistryEntry

1

UnsupportedCapabilityException

stability

Throws
   JAXRException: JAXR

setStability

void setStability(int stability)
   throws JAXRException

Sets the stability indicator for the RegistryEntry.

Capability Level: 1
   This method must throw UnsupportedCapabilityException in lower capability levels.

Parameters:
   stability - the stability indicator

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public java.util.Date getExpiration() throws JAXRException
Registry RegistryEntry expirationDate
expirationDate STABILITY_DYNAMIC null

1

UnsupportedCapabilityException

return Date
Throws JAXRException: JAXR

getExpiration

Date getExpiration() throws JAXRException

Gets expirationDate attribute of the RegistryEntry within the Registry. This attribute defines a time limit upon the stability indication provided by the stability attribute. Once the expirationDate has been reached the stability attribute in effect becomes STABILITY_DYNAMIC implying that content can change at any time and in any manner. A null value implies that there is no expiration on stability attribute.

Capability Level: 1
This method must throw UnsupportedCapabilityException in lower capability levels.

Returns:
the expiration Date for the stability indicator

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setExpiration(java.util.Date expiration) throws JAXRException
expirationDate
setExpiration

void setExpiration(Date expiration)
  throws JAXRException

Sets the expirationDate.

Capability Level: 1

Parameters:
  expiration - the expiration Date for the stability indicator

Throws:
  JAXRException - If the JAXR provider encounters an internal error
javax.xml.registry Class RegistryException

java.lang.Object
    ↓ java.lang.Throwable
        ↓ java.lang.Exception
            ↓ javax.xml.registry.JAXRException
                ↓ javax.xml.registry.RegistryException

All Implemented Interfaces:
    Serializable, JAXRResponse

Direct Known Subclasses:
    DeleteException, FindException, SaveException

public class RegistryException
extends JAXRException

Extends: Throwable > Exception > JAXRException
Extended by: DeleteException, FindException, SaveException

JAXR Exception

This is the common base class for all Exceptions that are detected on the registry provider side rather than the JAXR client side.

Author:
    Farrukh S. Najmi
See Also:
    Serialized Form

Field Summary

Fields inherited from class javax.xml.registry.JAXRException
cause
Fields inherited from interface javax.xml.registry.\texttt{JAXRResponse}
\begin{itemize}
\item \texttt{STATUS\_FAILURE}, \texttt{STATUS\_SUCCESS}, \texttt{STATUS\_UNAVAILABLE}, \texttt{STATUS\_WARNING}
\end{itemize}

## Constructor Summary

\begin{itemize}
\item \texttt{RegistryException()}
\hspace{1em} Constructs a JAXRException object with no reason or embedded Throwable.
\item \texttt{RegistryException(String reason)}
\hspace{1em} Constructs a JAXRException object with the given String as the reason for the exception being thrown.
\item \texttt{RegistryException(String reason, Throwable cause)}
\hspace{1em} Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.
\item \texttt{RegistryException(Throwable cause)}
\hspace{1em} Constructs a JAXRException object initialized with the given Throwable object.
\end{itemize}

## Method Summary

\begin{itemize}
\item \texttt{getErrorObjectKey()} \hspace{1em} Gets the Key to the first object that encountered an error in the registry.
\item \texttt{setErrorObjectKey(Key key)} \hspace{1em} Sets the Key to the first object that encountered an error in the registry.
\end{itemize}

Methods inherited from class javax.xml.registry.\texttt{JAXRException}
\begin{itemize}
\item \texttt{getCause}, \texttt{getMessage}, \texttt{getRequestId}, \texttt{getStatus}, \texttt{initCause}, \texttt{isAvailable}
\end{itemize}

Methods inherited from class java.lang.\texttt{Throwable}
\begin{itemize}
\item \texttt{fillInStackTrace}, \texttt{getLocalizedMessage}, \texttt{getStackTrace}, \texttt{printStackTrace}, \texttt{printStackTrace}, \texttt{printStackTrace}, \texttt{setStackTrace},
**Constructor Detail**

**public RegistryException()**

Throws: JAXRException

Constructs a JAXRException object with no reason or embedded Throwable.

**public RegistryException(String reason)**

Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**Parameters:**

- reason - a description of what caused the exception
public RegistryException(String reason, Throwable cause)

    JAXRException     String     Throwable

Throwable

    reason
    cause

JAXRException

public RegistryException(String reason, Throwable cause)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:

    reason - a description of what caused the exception
    cause - a Throwable object that is to be embedded in this JAXRException object

---

public RegistryException(Throwable cause)

    Throwable     JAXRException

    cause     Exception     Throwable

RegistryException

public RegistryException(Throwable cause)

    Constructs a JAXRException object initialized with the given Throwable object.

Parameters:

    cause - the Throwable that caused this Exception
**Method Detail**

```java
class getErrorObjectKey()

public Key getErrorObjectKey() throws JAXRException

Key return
```

**getErrorObjectKey**

```java
public Key getErrorObjectKey() throws JAXRException

Gets the Key to the first object that encountered an error in the registry.

Returns:
the key to the first object in error

Throws:
JAXRException
```

```java
public void setErrorObjectKey(Key key) throws JAXRException

Key key
```

**setErrorObjectKey**

```java
public void setErrorObjectKey(Key key) throws JAXRException

Sets the Key to the first object that encountered an error in the registry.

Parameters:
key - the key to the first object in error
```
Throws:

`JAXRException`

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.registry.infomodel Interface RegistryObject

All Superinterfaces:
- ExtensibleObject

All Known Subinterfaces:
- Association, AuditableEvent, Classification, ClassificationScheme,
  Concept, ExternalIdentifier, ExternalLink, ExtrinsicObject,
  Organization, RegistryEntry, RegistryPackage, Service,
  ServiceBinding, SpecificationLink, User

public interface RegistryObject
extends ExtensibleObject

Implements: ExtensibleObject
Implemented by: Association, AuditableEvent, Classification, Concept,
ExternalIdentifier, ExternalLink, Organization, RegistryEntry,
ServiceBinding, SpecificationLink, User

RegistryObject
See also javax.xml.registry.infomodel.RegistryEntry

The RegistryObject class is an abstract base class used by most classes
in the model. It provides minimal metadata for registry objects. It also
provides methods for accessing related objects that provide additional
dynamic metadata for the registry object.

Author:
Farrukh S. Najmi
See Also:
RegistryEntry

Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addAssociation(Association association)</td>
<td>Adds specified Association to use this object as source.</td>
</tr>
<tr>
<td>void addAssociations(Collection associations)</td>
<td>Adds specified Associations to use this object as source.</td>
</tr>
<tr>
<td>void addClassification(Classification classification)</td>
<td>Adds specified Classification to this object.</td>
</tr>
<tr>
<td>void addClassifications(Collection classifications)</td>
<td>Adds specified Classifications to this object.</td>
</tr>
<tr>
<td>void addExternalIdentifier(ExternalIdentifier externalIdentifier)</td>
<td>Adds specified ExternalIdentifier as an external identifier to this object.</td>
</tr>
<tr>
<td>void addExternalIdentifiers(Collection externalIdentifiers)</td>
<td>Adds specified ExternalIdentifiers as an external identifiers to this object.</td>
</tr>
<tr>
<td>void addExternalLink(ExternalLink externalLink)</td>
<td>Adds specified ExternalLink to this object.</td>
</tr>
<tr>
<td>void addExternalLinks(Collection externalLinks)</td>
<td>Adds specified ExternalLinks to this object.</td>
</tr>
<tr>
<td>Collection getAssociatedObjects()</td>
<td>Returns the collection of RegistryObject instances associated with this object.</td>
</tr>
<tr>
<td>Collection getAssociations()</td>
<td>Gets all Associations where this object is source.</td>
</tr>
<tr>
<td>Collection getAuditTrail()</td>
<td>Returns the complete audit trail of all requests that state change in this object as an ordered Collection of AuditableEvent objects.</td>
</tr>
<tr>
<td>Collection getClassifications()</td>
<td>Gets the Classification instances that classify this object.</td>
</tr>
<tr>
<td>InternationalString getDescription()</td>
<td>Gets the textual description for this object.</td>
</tr>
<tr>
<td>Collection getExternalIdentifiers()</td>
<td>Returns the ExternalIdentifiers associated with this object. are external identifiers for this object.</td>
</tr>
<tr>
<td>Collection getExternalLinks()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>getKey()</td>
<td>Gets the key representing the universally unique ID (UUID) for this object.</td>
</tr>
<tr>
<td>getLifeCycleManager()</td>
<td>Returns the LifeCycleManager that created this object.</td>
</tr>
<tr>
<td>getName()</td>
<td>Gets the user-friendly name of this object.</td>
</tr>
<tr>
<td>getObjectType()</td>
<td>Gets the object type that best describes the RegistryObject.</td>
</tr>
<tr>
<td>getRegistryPackages()</td>
<td>Returns the Package associated with this object.</td>
</tr>
<tr>
<td>getSubmittingOrganization()</td>
<td>Gets the Organization that submitted this RegistryObject.</td>
</tr>
<tr>
<td>removeAssociation(Association association)</td>
<td>Removes specified Association from this object.</td>
</tr>
<tr>
<td>removeAssociations(Collection associations)</td>
<td>Removes specified Associations from this object.</td>
</tr>
<tr>
<td>removeClassification(Classification classification)</td>
<td>Removes specified Classification from this object.</td>
</tr>
<tr>
<td>removeClassifications(Collection classifications)</td>
<td>Removes specified Classifications from this object.</td>
</tr>
<tr>
<td>removeExternalIdentifier(ExternalIdentifier externalIdentifier)</td>
<td>Removes specified ExternalIdentifier as an external identifier from this object.</td>
</tr>
<tr>
<td>removeExternalIdentifiers(Collection externalIdentifiers)</td>
<td>Removes specified ExternalIdentifiers as an external identifiers from this object.</td>
</tr>
<tr>
<td>removeExternallLink(Externallink externalLink)</td>
<td>Removes specified Externallink from this object.</td>
</tr>
<tr>
<td>removeExternallinks(Collection externallinks)</td>
<td>Removes specified Externallinks from this object.</td>
</tr>
<tr>
<td>setAssociations(Collection associations)</td>
<td>Replaces all previous Associations from this object with specified Associations.</td>
</tr>
</tbody>
</table>
void **setClassifications**(Collection classifications)  
Replaces all previous Classifications with specified Classifications.

void **setDescription**(InternationalString description)  
Sets the context independent textual description for this object.

void **setExternalIdentifiers**(Collection externalIdentifiers)  
Replaces all previous external identifiers with specified Collection of ExternalIdentifiers as an external identifier.

void **setExternalLinks**(Collection externalLinks)  
Replaces all previous ExternalLinks with specified ExternalLinks.

void **setKey**(Key key)  
Sets the key representing the universally unique ID (UUID) for this object.

void **setName**(InternationalString name)  
Sets user-friendly name of object in repository.

String **toXML** ()  
Returns a registry provider specific XML representation of this Object.

Methods inherited from interface  
javax.xml.registry.infomodel.**ExtensibleObject**  
addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

---

**Method Detail**

public **Key** getKey() throws **JAXRException**  
ID (UUID)

0

    return **Key**  
    Throws **JAXRException**: JAXR
getKey

public Key getKey() throws JAXRException

Gets the key representing the universally unique ID (UUID) for this object.

**Capability Level:** 0

**Returns:**
the Key for this object

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

getDescription

public InternationalString getDescription() throws JAXRException

0

return null

**Throws:**
JAXRException: JAXR
the description for this object which must not be null

Throws:  
JAXRException - If the JAXR provider encounters an internal error

public void setDescription(InternationalString description) 
throws JAXRException

Sets the context independent textual description for this object.

Capability Level: 0

Parameters:  
description - the description for this object

Throws: 
JAXRException - If the JAXR provider encounters an internal error

public InternationalString getName() throws JAXRException

0
return null
Throws JAXRException: JAXR

getName

**InternationalString** getName()
throws JAXRException

Gets the user-friendly name of this object.

**Capability Level:** 0

**Returns:**
the name for this object which must not be null.

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

```java
public void setName(**InternationalString** name) throws JAXRException
```

Sets user-friendly name of object in repository.

**Capability Level:** 0
Parameters:
   name - the name for this object

Throws:
   JAXRException - If the JAXR provider encounters an internal error

```
public void setKey(Key key) throws JAXRException
ID (UUID)
```

```
key
```

Throws
   JAXRException: JAXR

setKey

```
void setKey(Key key)
throws JAXRException
```

Sets the key representing the universally unique ID (UUID) for this object.

**Capability Level: 0**

Parameters:
   key - the key for this object

Throws:
   JAXRException - If the JAXR provider encounters an internal error

```
public String toXML() throws JAXRException
Object XML API
UnsupportedCapabilityException
```

0
toXML

```java
String toXML() throws JAXRException
```

Returns a registry provider specific XML representation of this Object. This may be used as a last resort back door way to get to a provider specific information element that is not accessible via the API. Implementation may choose to throw a UnsupportedCapabilityException.

**Capability Level: 0**

**Returns:**
the String containing the XML representation for this object

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

---

```java
public void addClassification(Classification classification) throws JAXRException
```

Classification Classification classifiedObject

0

**classification**

**Throws**
JAXRException: JAXR

---

```java
void addClassification(Classification classification) throws JAXRException
```

addClassification
Adds specified Classification to this object. Silently replaces the classifiedObject in Classification with reference to this object.

**Capability Level:** 0

**Parameters:**
- classification - the Classification being added

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

```java
public void addClassifications(java.util.Collection<? extends Classification> classifications) throws JAXRException
```

**addClassifications**

```java
void addClassifications(Collection classifications) throws JAXRException
```

Adds specified Classifications to this object. Silently replaces the classifiedObject in Classifications with reference to this object.

**Capability Level:** 0

**Parameters:**
- classifications - the Collection of Classifications being added

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error
public void removeClassification(Classification classification) throws JAXRException

Removes specified Classification from this object.

Capability Level: 0

Parameters:

classification - the Classification being removed

Throws:

JAXRException - If the JAXR provider encounters an internal error

public void removeClassifications(java.util.Collection<E> classifications) throws JAXRException

Removes specified Classifications from this object.

Capability Level: 0

Parameters:

classifications - the Classifications being removed

Throws:

JAXRException - If the JAXR provider encounters an internal error
void removeClassifications(Collection classifications) throws JAXRException

Removes specified Classifications from this object.

**Capability Level: 0**

**Parameters:**
- classifications - the Collection of Classifications being removed

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

public void setClassifications(java.util.Collection<E> classifications) throws JAXRException

Classification Classification

0

classifications Classification Collection

**Throws**
- JAXRException: JAXR

**setClassifications**

void setClassifications(Collection classifications) throws JAXRException

Replaces all previous Classifications with specified Classifications.

**Capability Level: 0**

**Parameters:**
- classifications - the Collection of Classifications being set

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error
public java.util.Collection\<E\> getClassifications() throws JAXRException

Classification

0

return Classification Collection Collection null

Throws JAXRException: JAXR

link aggregationByValue

associates 

supplierCardinality 

label

See also javax.xml.registry.infomodel.Classification

gOfClassifications

Collection getClassifications() throws JAXRException

Gets the Classification instances that classify this object.

Capability Level: 0

Returns: Collection of Classification instances. The Collection may be empty but not null.

Throws: JAXRException - If the JAXR provider encounters an internal error

See Also: Classification

public java.util.Collection\<E\> getAuditTrail() throws
JAXRException
AuditableEvent Collection

1
UnsupportedCapabilityException

return AuditableEvent CollectionCollection null

Throws JAXRException: JAXR

link aggregationByValue

associates <{AuditableEvent}>

supplierCardinality 1..*

label auditTrail

See also javax.xml.registry.infomodel.AuditableEvent

getauditTrail

Collection getAuditTrail() throws JAXRException

Returns the complete audit trail of all requests that effected a state change in this object as an ordered Collection of AuditableEvent objects.

Capability Level: 1
This method must throw UnsupportedCapabilityException in lower capability levels.

Returns:
Collection of AuditableEvent instances. The Collection may be empty but not null.

Throws:
JBAXRException - If the JAXR provider encounters an internal error

See Also:
AuditableEvent
public void addAssociation(Association association) throws JAXRException

Adding specified Association to use this object as source. Silently replaces the sourceObject in Association with reference to this object.

Capability Level: 0

Parameters:

- association - the Association being added

Throws:

- JAXRException - If the JAXR provider encounters an internal error

public void addAssociations(java.util.Collection<E> associations) throws JAXRException

Adding specified Associations to use this object as source. Silently replaces the sourceObject in Association with reference to this object.

Capability Level: 0

Parameters:

- associations - the Collection of Associations being added

Throws:

- JAXRException - If the JAXR provider encounters an internal error
addAssociations

void addAssociations(Collection associations) throws JAXRException

Adds specified Associations to use this object as source. Silently replaces the sourceObject in Associations with reference to this object.

Capability Level: 0

Parameters:
associations - the Collection of Associations being added

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void removeAssociation(Association association) throws JAXRException

Association

void removeAssociation(Association association) throws JAXRException

Removes specified Association from this object.

Capability Level: 0
Parameters:
  association - the Association being removed

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void removeAssociations(java.util.Collection<? extends Association> associations) throws JAXRException

Removes specified Associations from this object.

Capability Level: 0

Parameters:
  associations - the Collection of Associations being removed

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void setAssociations(java.util.Collection<? extends Association> associations) throws JAXRException

This method sets the specified Associations to this object.

Capability Level: 0

Parameters:
  associations - the Collection of Associations being set

Throws:
  JAXRException - If the JAXR provider encounters an internal error
setAssociations

void setAssociations(Collection associations) throws JAXRException

Replaces all previous Associations from this object with specified Associations.

Capability Level: 0

Parameters:

associations - the Collection of Associations being set

Throws:

JAXRException - If the JAXR provider encounters an internal error

---

public java.util.Collection&lt;E&gt; getAssociations() throws JAXRException

Association

0

return Association CollectionCollection null

Throws JAXRException: JAXR

See also javax.xml.registry.infomodel.Association

getAssociations

Collection getAssociations() throws JAXRException

Gets all Associations where this object is source.
Capability Level: 0

Returns:
Collection of Association instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
Association

```java
public java.util.Collection<E> getAssociatedObjects()
throws JAXRException

RegistryObject Collection

UnsupportedCapabilityException

return
RegistryObject Collection
null

Throws
JAXRException: JAXR
associationAsClass
{Association>
0..*

supplierCardinality
0..*

clientCardinality

associates

undirected

label

See also
javax.xml.registry.infomodel.RegistryObject
```
this object.

**Capability Level: 1**
This method must throw UnsupportedCapabilityException in lower capability levels.

**Returns:**
Collection of RegistryObject instances. The Collection may be empty but not null.

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

**See Also:**
RegistryObject

---

```java
public void addExternalIdentifier(ExternalIdentifier externalIdentifier) throws JAXRException
    ExternalIdentifier registryObject

0

    externalIdentifier Externallldentifier
    Throws JAXRException: JAXR
```

**addExternalIdentifier**

```java
void addExternalIdentifier(ExternalIdentifier externalIdentifier)
    throws JAXRException
```

Adds specified ExternalIdentifier as an external identifier to this object. Silently replaces the registryObject in ExternalIdentifier with reference to this object.

**Capability Level: 0**

**Parameters:**
public void addExternalIdentifiers(java.util.Collection<E> externalIdentifiers) throws JAXRException

Parameters:

- **externalIdentifiers**: the Collection of ExternalIdentifiers being added

Throws:

- **JAXRException**: If the JAXR provider encounters an internal error

---

**addExternalIdentifiers**

void addExternalIdentifiers(Collection externalIdentifiers)

throws JAXRException

Adds specified ExternalIdentifiers as an external identifiers to this object. Silently replaces the registryObject in ExternalIdentifiers with reference to this object.

**Capability Level**: 0

**Parameters**: 

- **externalIdentifiers**: the Collection of ExternalIdentifiers being added

**Throws**: 

- **JAXRException**: If the JAXR provider encounters an internal error

---

public void removeExternalIdentifier(ExternalIdentifier externalIdentifier) throws JAXRException

**Throws**: 

- **JAXRException**: If the JAXR provider encounters an internal error

---

**public void addExternalIdentifiers(ExternalIdentifier being added**

**Throws**: 

- **JAXRException**: If the JAXR provider encounters an internal error
ExternalIdentifier

0

Throws

JAXRException: JAXR

removeExternalIdentifier

void removeExternalIdentifier(ExternalIdentifier externalIdentifier)
throws JAXRException

Removes specified ExternalIdentifier as an external identifier from this object.

Capability Level: 0

Parameters:

externalIdentifier - the ExternalIdentifier being removed

Throws:

JAXRException - If the JAXR provider encounters an internal error

public void removeExternalIdentifiers(java.util.Collection<E> externalIdentifiers) throws JAXRException

ExternalIdentifier

0

Throws

JAXRException: JAXR

removeExternalIdentifiers
void removeExternalIdentifiers(Collection externalIdentifiers) throws JAXRException

Removes specified ExternalIdentifiers as an external identifiers from this object.

**Capability Level:** 0

**Parameters:**
- externalIdentifiers - the Collection of ExternalIdentifiers being removed

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

public void setExternalIdentifiers(java.util.Collection\<E\> externalIdentifiers) throws JAXRException

ExternalIdentifier Collection

0

externalIdentifiers - ExternalIdentifier Collection

**Throws**
- JAXRException: JAXR

---

**setExternalIdentifiers**

void setExternalIdentifiers(Collection externalIdentifiers) throws JAXRException

Replaces all previous external identifiers with specified Collection of ExternalIdentifiers as an external identifier.

**Capability Level:** 0

**Parameters:**
- externalIdentifiers - the Collection of ExternalIdentifiers being set
Throws:

`JAXRException` - If the JAXR provider encounters an internal error

```java
public java.util.Collection<E> getExternalIdentifiers()
throws JAXRException
```

Returns the `ExternalIdentifier` instances that are external identifiers for this object.

**Capability Level:** 0

**Returns:**

Collection of `ExternalIdentifier` instances. The Collection may be empty but not null.

**Throws:**

`JAXRException` - If the JAXR provider encounters an internal error
public void addExternalLink(ExternalLink externalLink) throws JAXRException
ExternalLink

0

addExternalLink

void addExternalLink(ExternalLink externalLink)
   throws JAXRException

    Adds specified ExternalLink to this object.

    Capability Level: 0

    Parameters:
        externalLinks - the ExternalLink being added

    Throws:
        JAXRException - If the JAXR provider encounters an internal error

public void addExternalLinks(java.util.Collection<E> externalLinks) throws JAXRException
ExternalLink

0

addExternalLinks
addExternalLinks

void addExternalLinks(Collection externalLinks)
throws JAXRException

Adds specified ExternalLinks to this object.

Capability Level: 0

Parameters:
   externalLinks - the Collection of ExternalLinks being added

Throws:
   JAXRException - If the JAXR provider encounters an internal error

removeExternalLink

public void removeExternalLink(ExternalLink externalLink)
throws JAXRException

Removes specified ExternalLink from this object.

Capability Level: 0

Parameters:
public void removeExternalLinks(java.util.Collection\<E\> externalLinks) throws JAXRException

Removes specified ExternalLinks from this object.

Capability Level: 0

Parameters:
- externalLinks - the Collection of ExternalLinks being removed

Throws:
- JAXRException - If the JAXR provider encounters an internal error
setExternalLinks

void setExternalLinks(Collection externalLinks)

throws JAXRException

Replaces all previous ExternalLinks with specified ExternalLinks.

Capability Level: 0

Parameters:

externalLink - the Collection of ExternalLinks being set

Throws:

JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getExternalLinks() throws JAXRException

ExternalLink

0

return

ExternalLink CollectionCollection

null

JAXRException: JAXR

associationAsClass

associates

supplierCardinality

clientCardinality

undirected

supplierRole

clientRole

See also

javax.xml.registry.infomodel.ExternalLink
getExternalLinks

`Collection getExternalLinks()` throws `JAXRException`

Returns the ExternalLinks associated with this object.

**Capability Level: 0**

**Returns:**
Collection of ExternalLink instances. The Collection may be empty but not null.

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error

**See Also:**
`ExternalLink`

---

public `Concept getObjectType()` throws `JAXRException`

`RegistryObject`

1

`UnsupportedCapabilityException`

```
return ObjectType ClassificationScheme Concept
```  

**Throws:**
`JAXRException`: JAXR

---

getObjectType

`Concept getObjectType()` throws `JAXRException`

Gets the object type that best describes the RegistryObject.

**Capability Level: 1**
This method must throw UnsupportedCapabilityException in lower capability levels.

**Returns:**
the object type as a Concept within the pre-defined ClassificationScheme named ObjectType

**Throws:**
  - JAXRException - If the JAXR provider encounters an internal error

```java
public Organization getSubmittingOrganization() throws JAXRException
```

**getSubmittingOrganization**

**Returns:**
the Organization that submitted this object to the registry

**Throws:**
  - JAXRException - If the JAXR provider encounters an internal error

```java
public java.util.Collection<E> getRegistryPackages() throws JAXRException
```

1
UnsupportedCapabilityException

\[
\text{return} \qquad \begin{array}{l}
\text{RegistryPackage}\quad \text{Collection}\quad \text{null} \\
\text{Throws} \qquad \text{JAXRException}:\quad \text{JAXR} \\
\text{supplierCardinality} \quad 0..* \\
\text{clientCardinality} \quad 1..* \\
\text{associates} \quad \langle\text{RegistryPackage}\rangle \\
\text{undirected} \quad \text{en} \\
\text{clientRole} \quad \text{memberObjects} \\
\text{supplierRole} \quad \text{packages} \\
\text{associationAsClass} \quad \langle\text{Association}\rangle \\
\text{See also} \quad \text{javax.xml.registry.infomodel.RegistryPackage} \\
\end{array}
\]

getRegistryPackages

\[
\text{Collection} \quad \text{getRegistryPackages}() \quad \text{throws} \quad \text{JAXRException}
\]

Returns the Package associated with this object.

**Capability Level: 1**

This method must throw UnsupportedCapabilityException in lower capability levels.

**Returns:**
Collection of RegistryPackage instances. The Collection may be empty but not null.

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

**See Also:**
RegistryPackage

\[
\text{public} \quad \text{LifeCycleManager} \quad \text{getLifeCycleManager}() \quad \text{throws}
\]
JAXRException

LifeCycleManager

0

    return LifeCycleManager

Throws JAXRException: JAXR

getLifeCycleManager

LifeCycleManager getLifeCycleManager() throws JAXRException

Returns the LifeCycleManager that created this object.

Capability Level: 0

Returns:
the LifeCycleManager objet that created this object

Throws:
    JAXRException - If the JAXR provider encounters an internal error

PS:
javax.xml.registry.infomodel Interface RegistryPackage

All Superinterfaces:
   ExtensibleObject, RegistryEntry, RegistryObject, Versionable

public interface RegistryPackage
   extends RegistryEntry

Implements: RegistryEntry

RegistryPackage  RegistryEntry  RegistryEntry
RegistryObjectRegistryObject  Package

See also   javax.xml.registry.infomodel.RegistryObject

RegistryPackage instances are RegistryEntries that group logically related RegistryEntries together. A package may contain any number of RegistryObjects. A RegistryObject may be a member of any number of Packages.

Author:
   Farrukh S. Najmi

See Also:
   RegistryObject

Field Summary

Fields inherited from interface
javax.xml.registry.infomodel.RegistryEntry
STABILITY_DYNAMIC, STABILITY_DYNAMIC_COMPATIBLE, STABILITY_STATIC,
STATUS_APPROVED, STATUS_DEPRECATED, STATUS_SUBMITTED,
STATUS_WITHDRAWN
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addRegistryObject(RegistryObject registryObject)</code></td>
<td>Adds a child RegistryObject as member.</td>
</tr>
<tr>
<td><code>addRegistryObjects(Collection registryObjects)</code></td>
<td>Adds a Collection of RegistryObject children as members.</td>
</tr>
<tr>
<td><code>getRegistryObjects()</code></td>
<td>Gets the collection of member RegistryObjects of this RegistryPackage.</td>
</tr>
<tr>
<td><code>removeRegistryObject(RegistryObject registryObject)</code></td>
<td>Removes a child RegistryObject from membership.</td>
</tr>
<tr>
<td><code>removeRegistryObjects(Collection registryObjects)</code></td>
<td>Removes a Collection of children RegistryObjects from membership.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.registry.infomodel.RegistryEntry`:

- `getExpiration`, `getStability`, `getStatus`, `setExpiration`, `setStability`  

Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`:

- `addAssociation`, `addAssociations`, `addClassification`, `addClassifications`, `addExternalIdentifier`, `addExternalIdentifiers`, `addExternalLink`, `addExternalLinks`, `getAssociatedObjects`, `getAssociations`, `getAuditTrail`, `getClassifications`, `getDescription`, `getExternalIdentifiers`, `getExternalLinks`, `getKey`, `getLifeCycleManager`, `getName`, `getObjectType`, `getRegistryPackages`, `getSubmittingOrganization`, `removeAssociation`, `removeAssociations`, `removeClassification`, `removeClassifications`, `removeExternalIdentifier`, `removeExternalIdentifiers`, `removeExternalLink`, `removeExternalLinks`, `setAssociations`, `setClassifications`, `setDescription`, `setExternalIdentifiers`, `setExternalLinks`, `setKey`, `setName`, `toXML`  

Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`:

- `addSlot`, `addSlots`, `getSlot`, `getSlots`, `removeSlot`, `removeSlots`
Methods inherited from interface javax.xml.registry.infomodel.Versionable

Method Detail

public void addRegistryObject(RegistryObject registryObject) throws JAXRException
RegistryObject

1

registryObject RegistryObject
Throws JAXRException: JAXR

addRegistryObject

void addRegistryObject(RegistryObject registryObject) throws JAXRException

Adds a child RegistryObject as member.

Capability Level: 1

Parameters:
registryObject - the RegistryObject being added

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void addRegistryObjects(java.util.Collection<E> registryObjects) throws JAXRException
RegistryObject Collection
addRegistryObjects

void addRegistryObjects(Collection registryObjects)
throws JAXRException

Adds a Collection of RegistryObject children as members.

Capability Level: 1

Parameters:
registryObjects - the Collection of RegistryObjects being added

Throws:
JAXRException - If the JAXR provider encounters an internal error

removeRegistryObject

void removeRegistryObject(RegistryObject registryObject)
throws JAXRException

Removes a child RegistryObject from membership.
Capability Level: 1

Parameters:
   registryObject - the RegistryObject being removed

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public void removeRegistryObjects(java.util.Collection<E> registryObjects) throws JAXRException
RegistryObject Collection

removeRegistryObjects

void removeRegistryObjects(Collection registryObjects)
   throws JAXRException

Removes a Collection of children RegistryObjects from membership.

Capability Level: 1

Parameters:
   registryObjects - the Collection of RegistryObject being removed

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public java.util.Set<E> getRegistryObjects() throws JAXRException
return RegistryObject Set
Throws JAXRException: JAXR

getRegistryObjects

Set getRegistryObjects()
throws JAXRException

Gets the collection of member RegistryObjects of this RegistryPackage.

Capability Level: 1

Returns: the Set of RegistryObjects that are members of this object
Throws: JAXRException - If the JAXR provider encounters an internal error

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
public interface RegistryService

JAXR Connection JAXR

See also javax.xml.registry.Connection

This is the principal interface implemented by a JAXR provider. A registry client can get this interface from a Connection to a registry. It provides the methods that are used by the client to discover various capability specific interfaces implemented by the JAXR provider.

Author:
Farrukh S. Najmi

See Also:
Connection

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BulkResponse</strong>&lt;br&gt;getBulkResponse(String requestId)</td>
</tr>
<tr>
<td><strong>BusinessLifeCycleManager</strong>&lt;br&gt;getBusinessLifeCycleManager()</td>
</tr>
<tr>
<td><strong>BusinessQueryManager</strong>&lt;br&gt;getBusinessQueryManager()</td>
</tr>
<tr>
<td><strong>CapabilityProfile</strong>&lt;br&gt;getCapabilityProfile()</td>
</tr>
<tr>
<td><strong>DeclarativeQueryManager</strong>&lt;br&gt;getDeclarativeQueryManager()</td>
</tr>
</tbody>
</table>
object implemented by the JAXR provider.

<table>
<thead>
<tr>
<th>ClassificationScheme</th>
<th>getDefaultPostalScheme()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the default user-defined postal scheme for codifying the attributes of PostalAddress.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>makeRegistrySpecificRequest(String request)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Takes a String that is an XML request in a registry-specific format, sends the request to the registry, and returns a String that is the registry-specific XML response.</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public CapabilityProfile getCapabilityProfile() throws JAXRException
JAXR CapabilityProfile

0

return JAXR CapabilityProfile

Throws JAXRException: JAXR

associates <{javax.xml.registry.CapabilityProfile}>

See also javax.xml.registry.LifeCycleManager
```

getCapabilityProfile

`CapabilityProfile getCapabilityProfile()` throws `JAXRException`

Returns the CapabilityProfile for the JAXR provider.

**Capability Level:** 0

**Returns:**

the CapabilityProfile for a JAXR provider
public **BusinessLifeCycleManager**

getBusinessLifeCycleManager() throws **JAXRException**

JAXR BusinessLifeCycleManager

0

```java
return BusinessLifeCycleManager
Throws JAXRException: JAXR
associates <javax.xml.registry.BusinessLifeCycleManager>
See also javax.xml.registry.LifeCycleManager
```

**getBusinessLifeCycleManager**

**BusinessLifeCycleManager** getBusinessLifeCycleManager() throws **JAXRException**

Returns the BusinessLifeCycleManager object implemented by the JAXR provider.

**Capability Level:** 0

**Returns:**
the BusinessLifeCycleManager

**Throws:**

**JAXRException** - If the JAXR provider encounters an internal error

**See Also:**
**LifeCycleManager**
public **BusinessQueryManager** getBusinessQueryManager() throws **JAXRException**

JAXR **BusinessQueryManager**

```java
return BusinessQueryManager
Throws JAXRException: JAXR
associates <{BusinessQueryManager}> directed
```

**getBusinessQueryManager**

```java
BusinessQueryManager getBusinessQueryManager() throws JAXRException

Returns the BusinessQueryManager object implemented by the JAXR provider.

**Capability Level: 0**

**Returns:**
the BusinessQueryManager

**Throws:**
JAXRException - If the JAXR provider encounters an internal error
```

public **DeclarativeQueryManager** getDeclarativeQueryManager() throws **JAXRException**, **UnsupportedCapabilityException**

JAXR **DeclarativeQueryManager**

```java
1
return DeclarativeQueryManager
```
**getDeclarativeQueryManager**

`DeclarativeQueryManager getDeclarativeQueryManager()` throws `JAXRException`

Returns the DeclarativeQueryManager object implemented by the JAXR provider.

**Capability Level: 1**

**Returns:**
the DeclarativeQueryManager

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error
- `UnsupportedCapabilityException`

```java
public BulkResponse getBulkResponse(String requestId) throws InvalidRequestException, JAXRException
    requestId BulkResponse requestId
BulkResponse requestId Bulkresponse
InvalidRequestException
```

**requestId** ID
**return** BulkResponse
**Throws** `InvalidRequestException`: requestId
**Throws** `JAXRException`: JAXR
getBulkResponse

BulkResponse getBulkResponse(String requestId) throws InvalidRequestException, JAXRException

Returns the BulkResponse associated with specified requestId. Once a client retrieves a BulkResponse for a particular requestId any subsequent calls to retrieve the Bulkresponse for the same requestId should result in an InvalidRequestException.

Capability Level: 0

Parameters:
requestId - the id for a previous asynchronous request

Returns:
The BulkResponse that contains the result for the specified request

Throws:
InvalidRequestException - if no responses exist for specified requestId
JAXRException - If the JAXR provider encounters an internal error

public ClassificationScheme getDefaultPostalScheme() throws JAXRException

PostalAddress

0

return ClassificationScheme

Throws JAXRException: JAXR

getDefaultPostalScheme

ClassificationScheme getDefaultPostalScheme() throws JAXRException
Gets the default user-defined postal scheme for codifying the attributes of PostalAddress.

**Capability Level: 0**

**Returns:**
the ClassificationScheme that is the default postal scheme

**Throws:**
  - `JAXRException` - If the JAXR provider encounters an internal error

```java
public String makeRegistrySpecificRequest(String request) throws JAXRException {
    String returnString = ...;
    return returnString;
}
```

**makeRegistrySpecificRequest**

`String makeRegistrySpecificRequest(String request) throws JAXRException` - Takes a String that is an XML request in a registry-specific format, sends the request to the registry, and returns a String that is the registry-specific XML response.

**Capability Level: 0**

**Parameters:**
request - the registry-specific request in a String representation

**Returns:**
the String that is the XML response in a registry-specific manner
Throws:

`JAXRException` - If the JAXR provider encounters an internal error
javax.ejb  Annotation Type Remote

```
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Remote
```

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean value()

Declares the remote business interface(s) for a session bean. When used on an interface, designates that interface as a remote business interface. In this case, no value() is provided.

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>value</th>
</tr>
</thead>
</table>

**abstract public Class<T>[] value()**

**value**

public abstract `Class[]` value

**Default:**

`{}`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  Annotation Type RemoteHome

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface RemoteHome

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean Remote Home Remote Home

Declares the Remote Home or adapted Remote Home interface for a session bean.

---

**Required Element Summary**

<table>
<thead>
<tr>
<th>Class</th>
<th>value</th>
</tr>
</thead>
</table>

---

**Element Detail**

abstract public Class<T> value()

value

public abstract Class value
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb  Annotation Type Remove

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface Remove

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Bean  Bean

Applied to a business method of a stateful session bean class. Indicates that the stateful session bean is to be removed by the container after completion of the method.

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boolean retainIfException</strong></td>
</tr>
</tbody>
</table>

**abstract public boolean retainIfException()**
true  Bean

**retainIfException**

public abstract boolean retainIfException

   If true, the stateful session bean will not be removed if an exception is thrown from the designated method.

**Default:**
false
javax.ejb  Class RemoveException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.ejb.RemoveException

All Implemented Interfaces:
   Serializable

public class RemoveException
   extends Exception

Extends: Throwable > Exception

Bean  EJB  EJB  RemoveException

The RemoveException exception is thrown at an attempt to remove an EJB object when the enterprise Bean or the container does not allow the EJB object to be removed.

See Also:
   Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RemoveException()</td>
<td>Constructs an RemoveException with no detail message.</td>
</tr>
<tr>
<td>RemoveException(String message)</td>
<td>Constructs an RemoveException with the specified detail message.</td>
</tr>
</tbody>
</table>

---

### Method Summary
Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public RemoveException()
RemoveException

RemoveException

public RemoveException()

Constructs an RemoveException with no detail message.

public RemoveException(String message)
RemoveException

RemoveException

public RemoveException(String message)

Constructs an RemoveException with the specified detail message.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS : 
A `Renderer` converts the internal representation of `UIComponent`s into the output stream (or writer) associated with the response we are creating for a particular request. Each `Renderer` knows how to render one or more `UIComponent` types (or classes), and advertises a set of render-dependent attributes that it recognizes for each supported `UIComponent`.

Families of `Renderer`s are packaged as a `RenderKit`, and together support the rendering of all of the `UIComponent`s in a view associated with a `FacesContext`. Within the set of `Renderer`s for a particular `RenderKit`, each must be uniquely identified by the `rendererType` property.

Individual `Renderer` instances will be instantiated as requested during the rendering process, and will remain in existence for the remainder of the lifetime of a web application. Because each instance may be invoked from more than one request processing thread simultaneously, they MUST be programmed in a thread-safe manner.
### Constructor Summary

**Renderer()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| **String** | `convertClientId(FacesContext context, String clientId)`  
Convert the component generated client id to a form suitable for transmission to the client. |
| **void** | `decode(FacesContext context, UIComponent component)`  
Decode any new state of the specified `UIComponent` from the request contained in the specified `FacesContext`, and store that state on the `UIComponent`. |
| **void** | `encodeBegin(FacesContext context, UIComponent component)`  
Render the beginning specified `UIComponent` to the output stream or writer associated with the response we are creating. |
| **void** | `encodeChildren(FacesContext context, UIComponent component)`  
Render the child components of this `UIComponent`, following the rules described for `encodeBegin()` to acquire the appropriate value to be rendered. |
| **void** | `encodeEnd(FacesContext context, UIComponent component)`  
Render the ending of the current state of the specified `UIComponent`, following the rules described for `encodeBegin()` to acquire the appropriate value to be rendered. |
| **Object** | `getConvertedValue(FacesContext context, UIComponent component, Object submittedValue)`  
Attempt to convert previously stored state information into an object of the type required for this component (optionally using the registered `Converter` for this component, if there is one). |
| **boolean** | `getRendersChildren()`  
Return a flag indicating whether this `Renderer` is responsible for rendering the children the component it is asked to render. |
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public Renderer()

Renderer

public Renderer()

Method Detail

public void decode(FacesContext context, UIComponent component)

FacesContext UIComponent UIComponent

UIComponent queueEvent()

context component

Throws NullPointerException: context component null

decode

public void decode(FacesContext context, UIComponent component)

Decode any new state of the specified UIComponent from the request contained in the specified FacesContext, and store that state on the UIComponent.
During decoding, events may be enqueued for later processing (by event listeners that have registered an interest), by calling queueEvent() on the associated UIComponent.

**Parameters:**
- **context** - FacesContext for the request we are processing
- **component** - UIComponent to be decoded.

**Throws:**
- NullPointerException - if context or component is null

```java
public void encodeBegin(FacesContext context, UIComponent component) throws java.io.IOException

UIComponent writer getConvertedValue()
decode()

context FacesContext
component UIComponent

Throws java.io.IOException: /
Throws NullPointerException: context component null
```

**encodeBegin**

```java
public void encodeBegin(FacesContext context, UIComponent component)
throws IOException

Render the beginning specified UIComponent to the output stream or writer associated with the response we are creating. If the conversion attempted in a previous call to getConvertedValue() for this component failed, the state information saved during execution of decode() should be used to reproduce the incorrect input.

**Parameters:**
- **context** - FacesContext for the request we are processing
- **component** - UIComponent to be rendered
public void encodeChildren(FacesContext context, UICOMPONENT component) throws java.io.IOException

    UICOMPONENT encodeBegin() rendersChildren true

    context FacesContext
    component UICOMPONENT

    Throws java.io.IOException: /
    Throws NullPointerException: context component null

**encodeChildren**

public void encodeChildren(FacesContext context, UICOMPONENT component) throws IOException

Render the child components of this UICOMPONENT, following the rules described for encodeBegin() to acquire the appropriate value to be rendered. This method will only be called if the rendersChildren property of this component is true.

**Parameters:**

context - FacesContext for the response we are creating
component - UICOMPONENT whose children are to be rendered

**Throws:**

IOException - if an input/output error occurs while rendering
NullPointerException - if context or component is null

public void encodeEnd(FacesContext context, UICOMPONENT component) throws java.io.IOException
**encodeBegin()**

```
context
component
```

**Throws**
- `java.io.IOException: /`
- `NullPointerException: context component null`

**encodeEnd**

```
public void encodeEnd(FacesContext context, UIComponent component)
throws IOException
```

Render the ending of the current state of the specified `UIComponent`, following the rules described for `encodeBegin()` to acquire the appropriate value to be rendered.

**Parameters:**
- `context` - `FacesContext` for the response we are creating
- `component` - `UIComponent` to be rendered

**Throws:**
- `IOException` - if an input/output error occurs while rendering
- `NullPointerException` - if context Or component is null

```
public String convertClientId(FacesContext context, String clientId)
```

**ID**

```
clientId
```

**Throws**
- `NullPointerException: context clientId null`
**convertClientId**

```java
public String convertClientId(FacesContext context, String clientId)
```

Convert the component generated client id to a form suitable for transmission to the client.

The default implementation returns the argument `clientId` unchanged.

**Parameters:**
- `context` - `FacesContext` for the current request
- `clientId` - the client identifier to be converted to client a specific format.

**Throws:**
- `NullPointerException` - if `context` or `clientId` is null

---

**public boolean getRendersChildren()**

```java
Renderer false
```

**getRendersChildren**

```java
public boolean getRendersChildren()
```

Return a flag indicating whether this `Renderer` is responsible for rendering the children the component it is asked to render. The default implementation returns `false`.

---

**public Object getConvertedValue(FacesContext context, UIComponent component, Object submittedValue) throws ConverterException**
javax.faces.convert.Converter

ConverterException

context context
component component
submittedValue submittedValue

Throws ConverterException:
Throws NullPointerException:

getConvertedValue

public Object getConvertedValue(FacesContext context, UIComponent component, Object submittedValue)
throws ConverterException

Attempt to convert previously stored state information into an object of the type required for this component (optionally using the registered Converter for this component, if there is one). If conversion is successful, the new value should be returned from this method; if not, a ConverterException should be thrown.

Parameters:
context - FacesContext for the request we are processing
component - UIComponent to be decoded.
submittedValue - a value stored on the component during decode.

Throws:
ConverterException - if the submitted value cannot be converted successfully.
NullPointerException - if context or component is null
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.render Class RenderKit

javax.lang.Object
  └─javax.faces.render.RenderKit

public abstract class RenderKit
  extends Object

<table>
<thead>
<tr>
<th>RenderKit</th>
<th>Renderer</th>
<th>Collection JavaServer Faces UICOMPONENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>RenderKit / Locale</td>
<td>RenderKit</td>
<td>Renderer</td>
</tr>
</tbody>
</table>

Factory

JavaServer Faces Web RenderKit RenderKitFactory
getRenderKit() RenderKit RenderKit
javax.faces.application.ViewHandler

RenderKit ResponseStateManager

RenderKit represents a collection of Renderer instances that, together, know how to render JavaServer Faces UICOMPONENT instances for a specific client. Typically, RenderKits are specialized for some combination of client device type, markup language, and/or user Locale. A RenderKit also acts as a Factory for associated Renderer instances, which perform the actual rendering process for each component.

A typical JavaServer Faces implementation will configure one or more RenderKit instances at web application startup. They are made available through calls to the getRenderKit() methods of RenderKitFactory. Because RenderKit instances are shared, they must be implemented in a thread-safe manner. Due to limitations in the current specification having multiple RenderKit instances at play in the same application requires a custom ViewHandler instance that is aware of how to deal with this case. This limitation will be lifted in a future version of the spec.
The RenderKit instance must also vend a ResponseStateManager instance, which is used in the process of saving and restoring tree structure and state.

## Constructor Summary

**RenderKit()**

## Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>addRenderer(String family, String rendererType, Renderer renderer)</strong></td>
<td>Register the specified Renderer instance, associated with the specified component family and rendererType, to the set of Renderers registered with this RenderKit, replacing any previously registered Renderer for this combination of identifiers.</td>
</tr>
<tr>
<td><strong>createResponseStream(OutputStream out)</strong></td>
<td>Use the provided OutputStream to create a new ResponseStream instance.</td>
</tr>
<tr>
<td><strong>createResponseWriter(Writer writer, String contentTypeList, String characterEncoding)</strong></td>
<td>Use the provided Writer to create a new ResponseWriter instance for the specified (optional) content type, and character encoding.</td>
</tr>
<tr>
<td><strong>getRenderer(String family, String rendererType)</strong></td>
<td>Return the Renderer instance most recently registered for the specified component family and rendererType, if any; otherwise, return null.</td>
</tr>
<tr>
<td><strong>getResponseStateManager()</strong></td>
<td>Return an instance of ResponseStateManager to handle rendering technology specific state management decisions.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Constructor Detail

public RenderKit()

Method Detail

abstract public void addRenderer(String family, String rendererType, Renderer renderer)

family rendererType Renderer Renderer
Renderer Renderer

renderer

Throws

NullPointerException: family rendererType renderer null

addRenderer

public abstract void addRenderer(String family,
String rendererType,
Renderer renderer)

Register the specified Renderer instance, associated with the specified component family and rendererType, to the set of
Renderer s registered with this RenderKit, replacing any previously registered Renderer for this combination of identifiers.

Parameters:
- family - Component family of the Renderer to register
- rendererType - Renderer type of the Renderer to register
- renderer - Renderer instance we are registering

Throws:
- NullPointerException - if family or rendererType or renderer is null

abstract public Renderer getRenderer(String family, String rendererType)

family  rendererType  Renderer  null

family  Rendered  Renderer
rendererType  Rendered  Renderer

Throws  NullPointerException:  family  rendererType  null

getRenderer

public abstract Renderer getRenderer(String family, String rendererType)

Return the Renderer instance most recently registered for the specified component family and rendererType, if any; otherwise, return null.

Parameters:
- family - Component family of the requested Renderer instance
- rendererType - Renderer type of the requested Renderer instance

Throws:
- NullPointerException - if family or rendererType is null
abstract public **ResponseStateManager** getResponseStateManager()

**ResponseStateManager**

getResponseStateManager

public abstract **ResponseStateManager** getResponseStateManager()

Return an instance of **ResponseStateManager** to handle rendering technology specific state management decisions.

abstract public **ResponseWriter**

createResponseWriter(java.io.Writer writer, String contentTypeList, String characterEncoding)

**ResponseWriter**

```
javax.servlet.ServletResponse getCharacterEncoding()
```

writer **ResponseWriter** writer

```
"Accept (Accept header style)"
```

RenderKit **null** RenderKit "Accept"

```
"String String
text/htmlapplication/xhtml+xml
```

text/xml **RenderKit** Accept HTTP

```
Accept Accept
```

RFC 2616_14.1

```
"ISO-8859-1"
```

ResponseWriter **null** the IANA

characterEncoding

```
null
```

**ResponseWriter**

`Throws` **IllegalArgumentException**: contentTypeList

characterEncoding
createResponseWriter

public abstract ResponseWriter createResponseWriter(Writer writer, String contentTypeList, String characterEncoding)

Use the provided Writer to create a new ResponseWriter instance for the specified (optional) content type, and character encoding.

Implementors are advised to consult the getCharacterEncoding() method of class ServletResponse to get the required value for the characterEncoding for this method. Since the Writer for this response will already have been obtained (due to it ultimately being passed to this method), we know that the character encoding cannot change during the rendering of the response.

Parameters:
- writer - the Writer around which this ResponseWriter must be built.
- contentTypeList - an "Accept header style" list of content types for this response, or null if the RenderKit should choose the best fit. As of the current version, the values accepted by the Standard render-kit for this parameter include any valid "Accept header style" String that includes the String text/html, application/xhtml+xml, application/xml or text/xml. This may change in a future version. The RenderKit must support a value for this argument that comes straight from the Accept HTTP header, and therefore requires parsing according to the specification of the Accept header. Please see Section 14.1 of RFC 2616 for the specification of the Accept header.
- characterEncoding - such as "ISO-8859-1" for this ResponseWriter, or null if the RenderKit should choose the best fit. Please see the IANA for a list of character encodings.

Returns: a new ResponseWriter.

Throws:
- IllegalArgumentException - if no matching content type can be found in contentTypeList, no appropriate content type can be found with the implementation dependent best fit algorithm, or no matching character encoding can be found for the argument.
abstract public ResponseStream createResponseStream(java.io.OutputStream out)

OutputStream ResponseStream

createResponseStream

public abstract ResponseStream createResponseStream(OutputStream out)

Use the provided OutputStream to create a new ResponseStream instance.
**javax.faces.render** Class **RenderKitFactory**

**java.lang.Object**
   ∧ **javax.faces.render.RenderKitFactory**

```
public abstract class RenderKitFactory
extends Object
```

**RenderKitFactory**  
**RenderKit**  
**JavaServer Faces**  
**Web**  
**RenderKitFactory**

```
RenderKitFactory factory = (RenderKitFactory) FactoryFinder.getFactory(FactoryFinder.RENDER_KIT_FACTORY);
```

**RenderKitFactory** is a factory object that registers and returns **RenderKit** instances. Implementations of JavaServer Faces must provide at least a default implementation of **RenderKit**. Advanced implementations (or external third party libraries) may provide additional **RenderKit** implementations (keyed by render kit identifiers) for performing different types of rendering for the same components.

There must be one **RenderKitFactory** instance per web application that is utilizing JavaServer Faces. This instance can be acquired, in a portable manner, by calling:

```
RenderKitFactory factory = (RenderKitFactory) FactoryFinder.getFactory(FactoryFinder.RENDER_KIT_FACTORY);
```

---

**Field Summary**
<table>
<thead>
<tr>
<th>Field Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HTML_BASIC_RENDER_KIT</strong></td>
</tr>
</tbody>
</table>

The render kit identifier of the default `RenderKit` instance for this JavaServer Faces implementation.

### Constructor Summary

#### `RenderKitFactory()`

### Method Summary

#### `abstract void addRenderKit(String renderKitId, RenderKit renderKit)`

- Register the specified `RenderKit` instance, associated with the specified `renderKitId`, to be supported by this `RenderKitFactory`, replacing any previously registered `RenderKit` for this identifier.

#### `abstract RenderKit getRenderKit(FacesContext context, String renderKitId)`

- Return a `RenderKit` instance for the specified render kit identifier, possibly customized based on dynamic characteristics of the specified `FacesContext`, if non-null.

#### `abstract Iterator<String> getRenderKitIds()`

- Return an `Iterator` over the set of render kit identifiers registered with this factory.

*Methods inherited from class java.lang.Object*

- `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`
public static final String HTML_BASIC_RENDER_KIT

The render kit identifier of the default RenderKit instance for this JavaServer Faces implementation.

See Also:
Constant Field Values

Constructor Detail

public RenderKitFactory()

Method Detail

abstract public void addRenderKit(String renderKitId, RenderKit renderKit)

void addRenderKit(String renderKitId, RenderKit renderKit)

Throws

NullPointerException: renderKitId null

addRenderKit

public abstract void addRenderKit(String renderKitId, RenderKit renderKit)

null

Register the specified RenderKit instance, associated with the
specified renderKitId, to be supported by this RenderKitFactory, replacing any previously registered RenderKit for this identifier.

Parameters:
- renderKitId - Identifier of the RenderKit to register
- renderKit - RenderKit instance that we are registering

Throws:
- NullPointerException - if renderKitId or renderKit is null

abstract public RenderKit getRenderKit(FacesContext context, String renderKitId)

Return a RenderKit instance for the specified render kit identifier, possibly customized based on dynamic characteristics of the specified FacesContext, if non-null. If there is no registered RenderKit for the specified identifier, return null. The set of available render kit identifiers is available via the getRenderKitIds() method.

Parameters:
- context - FacesContext for the request currently being processed, or null if none is available.
- renderKitId - Render kit identifier of the requested RenderKit instance
Throws:

- `IllegalArgumentException` - if no `RenderKit` instance can be returned for the specified identifier
- `NullPointerException` - if `renderKitId` is `null`

abstract public java.util.Iterator<E> getRenderKitIds()

```
Iterator RenderKitFactory.HTML_BASIC_RENDER_KIT
```

getRenderKitIds

code

class

public abstract Iterator<String> getRenderKitIds()

Return an `Iterator` over the set of render kit identifiers registered with this factory. This set must include the value specified by `RenderKitFactory.HTML_BASIC_RENDER_KIT`.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.servlet Interface RequestDispatcher

public interface RequestDispatcher

servletHTML  JSP servlet

servlet servlet RequestDispatcher

See also
getRequestDispatcher(java.lang.String),
getNamedDispatcher(java.lang.String),
getRequestDispatcher(java.lang.String)

Defines an object that receives requests from the client and sends them to any resource (such as a servlet, HTML file, or JSP file) on the server. The servlet container creates the RequestDispatcher object, which is used as a wrapper around a server resource located at a particular path or given by a particular name.

This interface is intended to wrap servlets, but a servlet container can create RequestDispatcher objects to wrap any type of resource.

Author:
Various

See Also:
ServletContext.getRequestDispatcher(java.lang.String),
ServletContext.getNamedDispatcher(java.lang.String),
ServletRequest.getRequestDispatcher(java.lang.String)

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>forward</strong> (ServletRequest request, ServletResponse response)</td>
</tr>
<tr>
<td>Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.</td>
</tr>
</tbody>
</table>
**Method Detail**

```java
class MethodDetail {
    public void forward(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException {
        ServletRequestWrapper request = (ServletRequestWrapper) request;
        ServletResponseWrapper response = (ServletResponseWrapper) response;

        ServletRequest wrappedRequest = request.getWrappedRequest();
        ServletResponse wrappedResponse = response.getWrappedResponse();

        // method implementation...
    }
}
```

Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server. This method allows one servlet to do...
preliminary processing of a request and another resource to
generate the response.

For a RequestDispatcher obtained via getRequestDispatcher(), the
ServletRequest object has its path elements and parameters
adjusted to match the path of the target resource.

forward should be called before the response has been committed to
the client (before response body output has been flushed). If the
response already has been committed, this method throws an
IllegalStateException. Uncommitted output in the response buffer is
automatically cleared before the forward.

The request and response parameters must be either the same
objects as were passed to the calling servlet's service method or be
subclasses of the ServletRequestWrapper or ServletResponseWrapper
classes that wrap them.

Parameters:
request - a ServletRequest object that represents the request
the client makes of the servlet
response - a ServletResponse object that represents the
response the servlet returns to the client

Throws:
ServletException - if the target resource throws this exception
IOException - if the target resource throws this exception
IllegalStateException - if the response was already committed

public void include(ServletRequest request,
ServletResponse response) throws ServletException,
java.io.IOException
servletJSP  HTML

ServletResponse  servlet

request  response  servlet service
ServletRequestWrapper  ServletResponseWrapper
include

void include(ServletRequest request,
              ServletResponse response)
  throws ServletException,
         IOException

Includes the content of a resource (servlet, JSP page, HTML file) in the response. In essence, this method enables programmatic server-side includes.

The ServletResponse object has its path elements and parameters remain unchanged from the caller's. The included servlet cannot change the response status code or set headers; any attempt to make a change is ignored.

The request and response parameters must be either the same objects as were passed to the calling servlet's service method or be subclasses of the ServletRequestWrapper or ServletResponseWrapper classes that wrap them.

Parameters:

request - a ServletRequest object that contains the client's request
response - a ServletResponse object that contains the servlet's response

Throws:

ServletException - if the included resource throws this exception
IOException - if the included resource throws this exception
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.ws  Annotation Type RequestWrapper

@Target(value=METHOD)  
@Retention(value=RUNTIME)  
@Documented

public @interface RequestWrapper

Implements: Annotation
@Target(value=METHOD)  
@Retention(value=RUNTIME)  
@Documented

bean  WebMethod  localName  operationName  
targetNamespace  SEI

Java  className

   since  JAX-WS 2.0

See also  javax.jws.WebMethod

Usage:

Used to annotate methods in the Service Endpoint Interface with the request wrapper bean to be used at runtime. The default value of the localName is the operationName, as defined in WebMethod annotation and the targetNamespace is the target namespace of the SEI.

When starting from Java this annotation is used to resolve overloading conflicts in document literal mode. Only the className is required in this case.

Since:
   JAX-WS 2.0

See Also:
   WebMethod

Optional Element Summary
String **className**

*Request wrapper bean name.*

String **localName**

*Elements local name.*

String **targetNamespace**

*Elements namespace name.*

---

abstract public String **localName**()

**localName**

public abstract String **localName**

*Elements local name.*

**Default:**

```
```

---

abstract public String **targetNamespace**()

**targetNamespace**

public abstract String **targetNamespace**

*Elements namespace name.*

**Default:**

```
```

---

abstract public String **className**()
bean

className

public abstract String className

Request wrapper bean name.

Default:

"

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.annotation  Annotation Type Resource

@Target(value={TYPE, FIELD, METHOD})
@Retention(value=RUNTIME)
public @interface Resource

Implements: Annotation
Inner classes: Resource.AuthenticationType
@Target(value={TYPE, FIELD, METHOD})
@Retention(value=RUNTIME)

Resource

Inherited private

The Resource annotation marks a resource that is needed by the application. This annotation may be applied to an application component class, or to fields or methods of the component class. When the annotation is applied to a field or method, the container will inject an instance of the requested resource into the application component when the component is initialized. If the annotation is applied to the component class, the annotation declares a resource that the application will look up at runtime.

Even though this annotation is not marked Inherited, deployment tools are required to examine all superclasses of any component class to discover all uses of this annotation in all superclasses. All such annotation instances specify resources that are needed by the application component. Note that this annotation may appear on private fields and methods of superclasses; the container is required to perform injection in these cases as well.

Optional Element Summary

[authentication]
<table>
<thead>
<tr>
<th><strong>Resource.AuthenticationType</strong></th>
<th>The authentication type to use for this resource.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>String</strong></td>
<td><strong>description</strong></td>
</tr>
<tr>
<td></td>
<td>Description of this resource.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><strong>mappedName</strong></td>
</tr>
<tr>
<td></td>
<td>A product specific name that this resource should be mapped to.</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><strong>name</strong></td>
</tr>
<tr>
<td></td>
<td>The JNDI name of the resource.</td>
</tr>
<tr>
<td><strong>boolean</strong></td>
<td><strong>shareable</strong></td>
</tr>
<tr>
<td></td>
<td>Indicates whether this resource can be shared between this component and other components.</td>
</tr>
<tr>
<td><strong>class</strong></td>
<td><strong>type</strong></td>
</tr>
<tr>
<td></td>
<td>The Java type of the resource.</td>
</tr>
</tbody>
</table>

```java
abstract public String name()

JNDI  JavaBean
```

**name**

```
public abstract String name
```

The JNDI name of the resource. For field annotations, the default is the field name. For method annotations, the default is the JavaBeans property name corresponding to the method. For class annotations, there is no default and this must be specified.

**Default:**

```
"
```

---

```java
abstract public Class<T> type()

Java  JavaBean
```
type

public abstract Class type

The Java type of the resource. For field annotations, the default is the type of the field. For method annotations, the default is the type of the JavaBeans property. For class annotations, there is no default and this must be specified.

Default:
java.lang.Object.class

abstract public Resource.AuthenticationType authentication()

authentication

public abstract Resource.AuthenticationType authentication

The authentication type to use for this resource. This may be specified for resources representing a connection factory of any supported type, and must not be specified for resources of other types.

Default:
CONTAINER

abstract public boolean shareable()

shareable
public abstract boolean shareable

    Indicates whether this resource can be shared between this component and other components. This may be specified for resources representing a connection factory of any supported type, and must not be specified for resources of other types.

    Default:
    true

abstract public String mappedName()

    A product specific name that this resource should be mapped to. The name of this resource, as defined by the name element or defaulted, is a name that is local to the application component using the resource. (It's a name in the JNDI java:comp/env namespace.) Many application servers provide a way to map these local names to names of resources known to the application server. This mapped name is often a global JNDI name, but may be a name of any form.

    Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

    Default:
    ""
abstract public String description()

description

public abstract String description

Description of this resource. The description is expected to be in the default language of the system on which the application is deployed. The description can be presented to the Deployer to help in choosing the correct resource.

Default:

"""

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.annotation Enum Resource.AuthenticationType

java.lang.Object
  ▼ java.lang.Enum<Resource.AuthenticationType>
    ▼ javax.annotation.Resource.AuthenticationType

All Implemented Interfaces:
  Serializable, Comparable<Resource.AuthenticationType>

Enclosing class:
  Resource

public static enum Resource.AuthenticationType
  extends Enum<Resource.AuthenticationType>

  Extends: Enum<E>
  Contained within: Resource

The two possible authentication types for a resource.

### Enum Constant Summary

<table>
<thead>
<tr>
<th>APPLICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONTAINER</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>static Resource.AuthenticationType</th>
<th>valueOf(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returns the enum constant of this type with the specified name.</td>
<td></td>
</tr>
</tbody>
</table>
static Resource.AuthenticationType[] values() Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

CONTAINER
public static final Resource.AuthenticationType CONTAINER

APPLICATION
public static final Resource.AuthenticationType APPLICATION

Method Detail

final public static Resource.AuthenticationType[] values()
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(Resource.AuthenticationType c : Resource.AuthenticationType.values())
    System.out.println(c);
```

**Returns:**

an array containing the constants of this enum type, in the order they're declared

---

```java
public static Resource.AuthenticationType valueOf(String name)
```

**valueOf**

```java
public static Resource.AuthenticationType valueOf(String name)
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**

- `name` - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**

`IllegalArgumentException` - if this enum type has no constant with the specified name
to license terms.

PS:
javax.resource.spi  Interface ResourceAdapter

public interface ResourceAdapter

JavaBean

version 1.0

This represents a resource adapter instance and contains operations for lifecycle management and message endpoint setup. A concrete implementation of this interface is required to be a JavaBean.

Version: 1.0
Author: Ram Jeyaraman

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>endpointActivation(MessageEndpointFactory endpointFactory, ActivationSpec spec)</code></td>
<td>This is called during the activation of a message endpoint.</td>
</tr>
<tr>
<td><code>endpointDeactivation(MessageEndpointFactory endpointFactory, ActivationSpec spec)</code></td>
<td>This is called when a message endpoint is deactivated.</td>
</tr>
<tr>
<td><code>getXAResources(ActivationSpec[] specs)</code></td>
<td>This method is called by the application server during crash recovery.</td>
</tr>
<tr>
<td><code>start(BootstrapContext ctx)</code></td>
<td>This is called when a resource adapter instance is bootstrapped.</td>
</tr>
<tr>
<td><code>stop()</code></td>
<td>This is called when a resource adapter instance is undeployed or during application server shutdown.</td>
</tr>
</tbody>
</table>
public void start(BootstrapContext ctx) throws ResourceAdapterInternalException

ResourceAdapter JavaBean

ctx

Throws ResourceAdapterInternalException:

start

void start(BootstrapContext ctx)
    throws ResourceAdapterInternalException

This is called when a resource adapter instance is bootstrapped. This may be during resource adapter deployment or application server startup. This is a startup notification from the application server, and this method is called by an application server thread. The application server thread executes in an unspecified context.

During this method call a ResourceAdapter JavaBean is responsible for initializing the resource adapter instance. Any exception thrown during this method call causes the application server to abort the bootstrap procedure for this specific resource adapter instance.

Parameters:
    ctx - a bootstrap context containing references to useful facilities that could be used by a resource adapter instance.

Throws:
    ResourceAdapterInternalException - indicates bootstrap failure. The resource adapter instance is unusable and must be discarded.
public void stop()

ResourceAdapter JavaBean

stop

void stop()

This is called when a resource adapter instance is undeployed or during application server shutdown. This is a shutdown notification from the application server, and this method is called by an application server thread. The application server thread executes in an unspecified context.

During this method call, a ResourceAdapter JavaBean is responsible for performing an orderly shutdown of the resource adapter instance. Any exception thrown by this method call does not alter the processing of the application server shutdown or resource adapter undeployment that caused this method call. The application server may log the exception information for error reporting purposes.

public void endpointActivation(MessageEndpointFactory endpointFactory, ActivationSpec spec) throws ResourceException

endpointFactory
  spec
  Throws

spec
  JavaBean
  NotSupportedException:
endpointActivation

void endpointActivation(MessageEndpointFactory endpointFactory, 
    ActivationSpec spec) 
    throws ResourceException

This is called during the activation of a message endpoint. This causes the resource adapter instance to do the necessary setup (ie., setup message delivery for the message endpoint with a message provider). Note that message delivery to the message endpoint might start even before this method returns.

Endpoint activation is deemed successful only when this method completes successfully without throwing any exceptions.

Parameters:
    endpointFactory - a message endpoint factory instance.
    spec - an activation spec JavaBean instance.

Throws:
    NotSupportedException - indicates message endpoint activation rejection due to incorrect activation setup information.
    ResourceException

public void endpointDeactivation(MessageEndpointFactory endpointFactory, ActivationSpec spec)

endpointActivation

endpointFactory
    spec JavaBean

endpointDeactivation

void endpointDeactivation(MessageEndpointFactory endpointFactory, 
    ActivationSpec spec)
This is called when a message endpoint is deactivated. The instances passed as arguments to this method call should be identical to those passed in for the corresponding endpointActivation call. This causes the resource adapter to stop delivering messages to the message endpoint.

Any exception thrown by this method is ignored. After this method call, the endpoint is deemed inactive.

**Parameters:**
- endpointFactory - a message endpoint factory instance.
- spec - an activation spec JavaBean instance.

```java
public XAResource[] getXAResources(ActivationSpec[] specs) throws ResourceException
```

**Throws** ResourceException:

<table>
<thead>
<tr>
<th>XAResource null</th>
<th>XAResource</th>
</tr>
</thead>
<tbody>
<tr>
<td>specs</td>
<td>ActivationSpec JavaBean JavaBean</td>
</tr>
</tbody>
</table>

This method is called by the application server during crash recovery. This method takes in an array of ActivationSpec JavaBeans and returns an array of XAResource objects each of which represents a unique resource manager. The resource adapter may return null if it does not implement the XAResource interface. Otherwise, it must return an array of XAResource objects, each of which represents a unique resource manager that was used by the endpoint applications. The application server uses the XAResource objects to query each resource manager for a list of in-doubt
transactions. It then completes each pending transaction by sending the commit decision to the participating resource managers.

**Parameters:**

specs - an array of ActivationSpec JavaBeans each of which corresponds to an deployed endpoint application that was active prior to the system crash.

**Returns:**

an array of XAResource objects each of which represents a unique resource manager.

**Throws:**

ResourceException - generic exception if operation fails due to an error condition.
javax.resource.spi  Interface  ResourceAdapterAssociation

All Known Subinterfaces:
   ActivationSpec

public interface ResourceAdapterAssociation

Implemented by:  ActivationSpec

<table>
<thead>
<tr>
<th>ResourceAdapter</th>
<th>ManagedConnectionFactory</th>
<th>ActivationSpec</th>
</tr>
</thead>
<tbody>
<tr>
<td>version</td>
<td></td>
<td>1.0</td>
</tr>
</tbody>
</table>

This interface specifies the methods to associate a ResourceAdapter object with other objects that implement this interface like ManagedConnectionFactory and ActivationSpec.

Version:
   1.0

Author:
   Ram Jeyaraman

### Method Summary

<table>
<thead>
<tr>
<th>ResourceAdapter</th>
<th>getRA()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Get the associated ResourceAdapter object.</td>
<td></td>
</tr>
</tbody>
</table>

| void | setRA(ResourceAdapter ra) |
| Associate this object with a ResourceAdapter object. |

### Method Detail
public ResourceAdapter getResourceAdapter()

ResourceAdapter

return ResourceAdapter

getResourceAdapter

ResourceAdapter getResourceAdapter()

Get the associated ResourceAdapter object.

Returns:

the associated ResourceAdapter object.

public void setResourceAdapter(ResourceAdapter ra)
throws ResourceException

ResourceAdapter

ra

ResourceAdapter

Throws ResourceException:

Throws ResourceAdapterInternalException:

Throws IllegalStateException:

setResourceAdapter

void setResourceAdapter(ResourceAdapter ra)
throws ResourceException

Associate this object with a ResourceAdapter object. Note, this method must be called exactly once. That is, the association must not change during the lifetime of this object.

Parameters:

ra - ResourceAdapter object to be associated with.

Throws:

ResourceException - generic exception.

ResourceAdapterInternalException - resource adapter related
error condition.

IllegalStateException - indicates that this object is in an illegal state for the method invocation. For example, this occurs when this method is called more than once on the same object.
Class

javax.resource.spi Class ResourceAdapterInternalException

java.lang.Object
  \  \ java.lang.Throwable
    \  \ java.lang.Exception
      \  \ javax.resource.ResourceException
        \  \ javax.resource.spi.ResourceAdapterInternalException

All Implemented Interfaces:
  Serializable

public class ResourceAdapterInternalException extends ResourceException

Extends: Throwable > Exception > ResourceException

ResourceAdapterInternalException

  • EIS
  • EIS
  •

version 1.0

A ResourceAdapterInternalException indicates any system-level error conditions related to a resource adapter. The common conditions indicated by this exception type are:

  • Invalid configuration for creation of a new physical connection. An example is invalid server name for a target EIS instance.
  • Failure to create a physical connection to a EIS instance due to communication protocol error or any resource adapter implementation specific error.
  • Error conditions internal to resource adapter implementation.
Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ResourceAdapterInternalException()</code></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><code>ResourceAdapterInternalException(String message)</code></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><code>ResourceAdapterInternalException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and an error code.</td>
</tr>
<tr>
<td><code>ResourceAdapterInternalException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>ResourceAdapterInternalException(Throwable cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class `javax.resource.ResourceException`
- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCause`, `setLinkedException`

Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, 
Constructor Detail

public ResourceAdapterInternalException()

null

ResourceAdapterInternalException

public ResourceAdapterInternalException()

Constructs a new instance with null as its detail message.

public ResourceAdapterInternalException(String message)

message

ResourceAdapterInternalException

public ResourceAdapterInternalException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.

public ResourceAdapterInternalException(Throwable cause)

cause throwable

cause Throwable
ResourceAdapterInternalException

public ResourceAdapterInternalException(Throwable cause)

Constructs a new throwable with the specified cause.

**Parameters:**
- cause - a chained exception of type Throwable.

---

public ResourceAdapterInternalException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

**Parameters:**
- message - the detail message.
- cause - a chained exception of type Throwable.

---

public ResourceAdapterInternalException(String message, String errorCode)

Constructs a new throwable with the specified message and cause.

**Parameters:**
- message - the detail message.
- errorCode - the error code.
ResourceAdapterInternalException

public ResourceAdapterInternalException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  SUMMARY: NESTED | FIELD | CONSTR | METHOD  FRAMES  NO FRAMES  DETAIL: FIELD | CONSTR | METHOD
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The interface `javax.resource.cci.ResourceAdapterMetaData` provides information about capabilities of a resource adapter implementation. Note that this interface does not provide information about an EIS instance that is connected through the resource adapter.

A CCI client uses a `ConnectionFactory.getMetaData` to get metadata information about the resource adapter. The `getMetaData` method does not require that an active connection to an EIS instance should have been established.

The `ResourceAdapterMetaData` can be extended to provide more information specific to a resource adapter implementation.

Since:

0.8

Version:

0.8

Author:

Rahul Sharma

See Also:

`javax.resource.cci.ConnectionFactory`
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getAdapterName()</code></td>
<td>Gets a tool displayable name of the resource adapter.</td>
</tr>
<tr>
<td><code>String getAdapterShortDescription()</code></td>
<td>Gets a tool displayable short description of the resource adapter.</td>
</tr>
<tr>
<td><code>String getAdapterVendorName()</code></td>
<td>Gets the name of the vendor that has provided the resource adapter.</td>
</tr>
<tr>
<td><code>String getAdapterVersion()</code></td>
<td>Gets the version of the resource adapter.</td>
</tr>
<tr>
<td><code>String[] getInteractionSpecsSupported()</code></td>
<td>Returns an array of fully-qualified names of InteractionSpec types supported by the CCI implementation for this resource adapter.</td>
</tr>
<tr>
<td><code>String getSpecVersion()</code></td>
<td>Returns a string representation of the version of the connector architecture specification that is supported by the resource adapter.</td>
</tr>
<tr>
<td><code>boolean supportsExecuteWithInputAndOutputRecord()</code></td>
<td>Returns true if the implementation class for the Interaction interface implements public boolean execute(InteractionSpec ispec, Record input, Record output) method; otherwise the method returns false.</td>
</tr>
<tr>
<td><code>boolean supportsExecuteWithInputRecordOnly()</code></td>
<td>Returns true if the implementation class for the Interaction interface implements public Record execute(InteractionSpec ispec, Record input) method; otherwise the method returns false.</td>
</tr>
<tr>
<td><code>boolean supportsLocalTransactionDemarcation()</code></td>
<td>Returns true if the resource adapter implements the LocalTransaction interface and supports local transaction demarcation on the underlying EIS instance through the LocalTransaction interface.</td>
</tr>
</tbody>
</table>
public String getAdapterVersion()

    return

getAdapterVersion

String getAdapterVersion()

    Gets the version of the resource adapter.

    Returns:
    String representing version of the resource adapter

public String getAdapterVendorName()

    return

getAdapterVendorName

String getAdapterVendorName()

    Gets the name of the vendor that has provided the resource adapter.

    Returns:
    String representing name of the vendor that has provided the resource adapter

public String getAdapterName()
getAdapterName

String getAdapterName()

Gets a tool displayable name of the resource adapter.

Returns:
String representing the name of the resource adapter

getAdapterShortDescription

String getAdapterShortDescription()

Gets a tool displayable short description of the resource adapter.

Returns:
String describing the resource adapter

getSpecVersion

String getSpecVersion()

Returns a string representation of the version of the connector
architecture specification that is supported by the resource adapter.

**Returns:**
String representing the supported version of the connector architecture

```
public String[] getInteractionSpecsSupported()
CCI InteractionSpec InteractionSpec
InteractionSpec CCI InteractionSpec 0

return CCI InteractionSpec
See alsojavax.resource.cci.InteractionSpec
```

getInteractionSpecsSupported

```
String[] getInteractionSpecsSupported()

Returns an array of fully-qualified names of InteractionSpec types supported by the CCI implementation for this resourceSpec adapter. Note that the fully-qualified class name is for the implementation class of an InteractionSpec. This method may be used by tools vendor to find information on the supported InteractionSpec types. The method should return an array of length 0 if the CCI implementation does not define specific InteractionSpec types.

**Returns:**
Array of fully-qualified class names of InteractionSpec classes supported by this resource adapter's CCI implementation

See Also:
InteractionSpec
```

generic supportsExecuteWithInputAndOutputRecord()

```
public boolean supportsExecuteWithInputAndOutputRecord()
```

supportsExecuteWithInputAndOutputRecord

boolean supportsExecuteWithInputAndOutputRecord()

Returns true if the implementation class for the Interaction interface implements public boolean execute(InteractionSpec ispec, Record input, Record output) method; otherwise the method returns false.

Returns:
boolean depending on method support

See Also:
Interaction
public boolean supportsLocalTransactionDemarcation()

    LocalTransaction LocalTransaction  EIS  true

        return  LocalTransaction  true false

See also  javax.resource.cci.LocalTransaction

supportsLocalTransactionDemarcation

boolean  supportsLocalTransactionDemarcation()

    Returns  true if the resource adapter implements the
    LocalTransaction interface and supports local transaction
    demarcation on the underlying EIS instance through the
    LocalTransaction interface.

    Returns:
    true if resource adapter supports resource manager local
    transaction demarcation through LocalTransaction interface;
    false otherwise

    See Also:
    LocalTransaction

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to
license terms.

PS :
javax.jms Class ResourceAllocationException

java.lang.Object
  └ java.lang.Throwable
    └ java.lang.Exception
      └ javax.jms.JMSException
        └ javax.jms.ResourceAllocationException

All Implemented Interfaces:
  Serializable

public class ResourceAllocationException
  extends JMSException

Extends: Throwable > Exception > JMSException

TopicConnectionFactory.createTopicConnection JMS

version 26 August 1998

This exception is thrown when a provider is unable to allocate the resources required by a method. For example, this exception should be thrown when a call to TopicConnectionFactory.createTopicConnection fails due to a lack of JMS provider resources.

Version:
  26 August 1998

Author:
  Rahul Sharma

See Also:
  Serialized Form
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResourceAllocationException</strong></td>
<td>Constructs a ResourceAllocationException with the specified reason.</td>
</tr>
<tr>
<td><strong>ResourceAllocationException</strong></td>
<td>Constructs a ResourceAllocationException with the specified reason and error code.</td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class javax.jms.JMSException

- `getErrorCode`, `getLinkedException`, `setLinkedException`

### Methods inherited from class java.lang.Throwable

- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public ResourceAllocationException(String reason, String errorCode)

Constructs a ResourceAllocationException with the specified reason and error code.

Parameters:
- reason - a description of the exception
- errorCode - a string specifying the vendor-specific error code

public ResourceAllocationException(String reason)

ResourceAllocationException null reason

ResourceAllocationException

public ResourceAllocationException(String reason)

Constructs a ResourceAllocationException with the specified reason. The error code defaults to null.

Parameters:
- reason - a description of the exception

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi  Class ResourceAllocationException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.resource.ResourceException
              └ javax.resource.spi.ResourceAllocationException

All Implemented Interfaces:
  Serializable

public class ResourceAllocationException
  extends RuntimeException

Extends: Throwable > Exception > RuntimeException

ResourceAllocationException
  version 1.0

A ResourceAllocationException can be thrown by an application server or resource adapter to indicate any failure to allocate system resources (example: threads, physical connections). An example is error condition when an upper bound is reached on the maximum number of physical connections that can be managed by an application server specific connection pool.

Version: 1.0
Author:
  Rahul Sharma, Ram Jeyaraman
See Also:
 Serialized Form

Constructor Summary
<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceAllocationException()</td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td>ResourceAllocationException(String message)</td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td>ResourceAllocationException(String message, String errorCode)</td>
<td>Constructs a new throwable with the specified detail message and error code.</td>
</tr>
<tr>
<td>ResourceAllocationException(String message, Throwable cause)</td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td>ResourceAllocationException(Throwable cause)</td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

**Method Summary**

**Methods inherited from class javax.resource.ResourceException**
- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`

**Methods inherited from class java.lang.Throwable**
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

**Methods inherited from class java.lang.Object**
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
ResourceAllocationException

public ResourceAllocationException()

    Constructs a new instance with null as its detail message.

-------------------

public ResourceAllocationException(String message)

    message

ResourceAllocationException

public ResourceAllocationException(String message)

    Constructs a new instance with the specified detail message.

    Parameters:
    message - the detail message.

-------------------

public ResourceAllocationException(Throwables acuse)

    cause Throwable

ResourceAllocationException

public ResourceAllocationException(Throwable cause)

    Constructs a new throwable with the specified cause.

    Parameters:
    cause - a chained exception of type Throwable.

-------------------

public ResourceAllocationException(String message,
ResourceAllocationException

public ResourceAllocationException(String message,
        Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public ResourceAllocationException(String message,
        String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.el  **Class ResourceBundleELResolver**

java.lang.Object  
  ↓  javax.el.ELResolver  
  ↓  javax.el.ResourceBundleELResolver

public class **ResourceBundleELResolver**

extends **ELResolver**

**Extends:** **ELResolver**

| java.util.ResourceBundle |
|---------------------------|---------------------------|
| java.util.ResourceBundle  | base                      |
| java.lang.String          |                           |
| **getObject(java.lang.String)** |

<table>
<thead>
<tr>
<th><strong>javadoc</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>since</strong></td>
</tr>
<tr>
<td>JSP 2.1</td>
</tr>
<tr>
<td><strong>See also</strong></td>
</tr>
<tr>
<td>javax.el.CompositeELResolver, javax.el.ELResolver, java.util.ResourceBundle</td>
</tr>
</tbody>
</table>

**Defines property resolution behavior on instances of** **ResourceBundle.**

This resolver handles base objects of type java.util.ResourceBundle. It accepts any object as a property and coerces it to a java.lang.String for invoking **ResourceBundle.getObject(java.lang.String).**

This resolver is read only and will throw a **PropertyNotWritableException** if setValue is called.

**ELResolvers** are combined together using **CompositeELResolvers**, to define rich semantics for evaluating an expression. See the javadocs for
**ELResolver** for details.

Since: JSP 2.1

See Also: CompositeELResolver, ELResolver, ResourceBundle

## Field Summary

Fields inherited from class javax.el.ELResolver

| RESOLVABLE_AT_DESIGN_TIME, TYPE |

## Constructor Summary

**ResourceBundleELResolver()**

## Method Summary

<table>
<thead>
<tr>
<th>Class&lt;?&gt;</th>
<th>getCommonPropertyType(ELContext context, Object base)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the base object is a ResourceBundle, returns the most general type that this resolver accepts for the property argument.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iterator</th>
<th>getFeatureDescriptors(ELContext context, Object base)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the base object is a ResourceBundle, returns an Iterator containing the set of keys available in the ResourceBundle.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class&lt;?&gt;</th>
<th>getType(ELContext context, Object base, Object property)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the base object is an instance of ResourceBundle, return null, since the resolver is read only.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>getValue(ELContext context, Object base, Object property)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the base object is an instance of ResourceBundle, the provided property will first be coerced to a String.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>boolean</th>
<th>isReadOnly(ELContext context, Object base, Object property)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the base object is not null and an instance of</td>
</tr>
</tbody>
</table>
void setValue(ELContext context, Object base, Object property, Object value)

    If the base object is a ResourceBundle, throw a PropertyNotWritableException.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public ResourceBundleELResolver()

ResourceBundleELResolver

public ResourceBundleELResolver()

Method Detail

public Object getValue(ELContext context, Object base, Object property)

    base ResourceBundle String base
    ResourceBundle getObject Object
    base ResourceBundle ELContext propertyResolved true
    true

    context
    base ResourceBundle
    property String
    ELContext propertyResolved true null

    return null ResourceBundle String Object
**Throws**

NullPointerException: context null

**Throws** **ELException**: cause

---

**getValue**

```java
public Object getValue(ELContext context,
                      Object base,
                      Object property)
```

If the base object is an instance of ResourceBundle, the provided property will first be coerced to a String. The object returned by `getObject` on the base ResourceBundle will be returned.

If the base is ResourceBundle, the `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver, before returning. If this property is not `true` after this method is called, the caller should ignore the return value.

**Specified by:**

`getValue` in class **ELResolver**

**Parameters:**

- `context` - The context of this evaluation.
- `base` - The ResourceBundle to analyze.
- `property` - The name of the property to analyze. Will be coerced to a String.

**Returns:**

If the `propertyResolved` property of `ELContext` was set to `true`, then `null` if property is `null`; otherwise the object for the given key (property coerced to String) from the `ResourceBundle`. If no object for the given key can be found, then the String "??" + key + "??".

**Throws:**

- NullPointerException - if context is `null`
- ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.
public Class<T> getType(ELContext context, Object base, Object property)

Throws

If the base object is an instance of ResourceBundle, return null, since the resolver is read only.

If the base is ResourceBundle, the propertyResolved property of the ELContext object must be set to true by this resolver, before returning. If this property is not true after this method is called, the caller should ignore the return value.

Specified by:

g getType in class ELResolver

Parameters:

c context - The context of this evaluation.
b base - The ResourceBundle to analyze.
p property - The name of the property to analyze.

Returns:

If the propertyResolved property of ELContext was set to true, then null; otherwise undefined.

Throws:
public void setValue(ELContext context, Object base, Object property, Object value)

Throws

NullPointersException - if context is null

PropertyNotWritableException - Always thrown if base is an instance of ResourceBundle.

If the base object is a ResourceBundle, throw a

PropertyNotWritableException.

Specified by:

setValue in class ELResolver

Parameters:

class context - The context of this evaluation.

Object base - The ResourceBundle to be modified. Only bases that are of type ResourceBundle are handled.

Object property - The String property to use.

Object value - The value to be set.

Throws:

NullPointersException - if context is null.

PropertyNotWritableException - Always thrown if base is an instance of ResourceBundle.
public boolean isReadOnly(ELContext context, Object base, Object property)

<table>
<thead>
<tr>
<th>context</th>
<th>ResourceBundle</th>
<th>true</th>
</tr>
</thead>
<tbody>
<tr>
<td>base</td>
<td>ResourceBundle</td>
<td>base</td>
</tr>
<tr>
<td>property</td>
<td>String</td>
<td></td>
</tr>
<tr>
<td>return</td>
<td>ELContext</td>
<td>propertyResolved</td>
</tr>
</tbody>
</table>

Throws: NullPointerException: context null

isReadOnly

public boolean isReadOnly(ELContext context, Object base, Object property)

If the base object is not null and an instance of ResourceBundle, return true.

Specified by: isReadOnly in class ELResolver

Parameters:
- context - The context of this evaluation.
- base - The ResourceBundle to be modified. Only bases that are of type ResourceBundle are handled.
- property - The String property to use.

Returns:
- If the propertyResolved property of ELContext was set to true, then true; otherwise undefined.

Throws:
- NullPointerException - if context is null

public java.util.Iterator<E> getFeatureDescriptors(ELContext context, Object base)

<table>
<thead>
<tr>
<th>base</th>
<th>ResourceBundle</th>
<th>Iterator</th>
<th>Resource</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>null</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Iterator 0     java.beans.FeatureDescriptor  info

ResourceBundle

- displayName - String
- name – displayName
- shortDescription –
- expert - false
- hidden - false
- preferred - true

FeatureDescriptor

- ELResolver#TYPE - String.class
- ELResolver#RESOLVABLE_AT_DESIGN_TIME - true

class context
  base ResourceBundle base

return 0

FeatureDescriptor Iterator base
  ResourceBundle null

getFeatureDescriptors

public Iterator getFeatureDescriptors(ELContext context,
                                      Object base)

If the base object is a ResourceBundle, returns an Iterator
containing the set of keys available in the ResourceBundle.
Otherwise, returns null.

The Iterator returned must contain zero or more instances of
FeatureDescriptor. Each info object contains information about a key
in the ResourceBundle, and is initialized as follows:

- displayName - The string key
- name - Same as displayName property.
- shortDescription - Empty string
• expert - false
• hidden - false
• preferred - true

In addition, the following named attributes must be set in the returned FeatureDescriptorS:
• ELResolver.TYPE - String.class
• ELResolver.RESOLVABLE_AT_DESIGN_TIME - true

Specified by:
  getFeatureDescriptors in class ELResolver

Parameters:
  context - The context of this evaluation.
  base - The bundle whose keys are to be iterated over. Only bases of type ResourceBundle are handled by this resolver.

Returns:
  An Iterator containing zero or more (possibly infinitely more) FeatureDescriptor objects, each representing a key in this bundle, or null if the base object is not a ResourceBundle.

See Also:
  FeatureDescriptor

public Class<T> getCommonPropertyType(ELContext context, Object base)
  base ResourceBundle property null

  base ResourceBundle String.class
  context
  base ResourceBundle base
  return base ResourceBundle null String.class

getCommonPropertyType

public Class<? super Object> getCommonPropertyType(ELContext context, Object base)
If the base object is a ResourceBundle, returns the most general type that this resolver accepts for the property argument. Otherwise, returns null.

Assuming the base is a ResourceBundle, this method will always return String.class.

**Specified by:**

getCommonPropertyType in class ELResolver

**Parameters:**

context - The context of this evaluation.

base - The bundle to analyze. Only bases of type ResourceBundle are handled by this resolver.

**Returns:**

null if base is not a ResourceBundle; otherwise String.class.
javax.resource  Class ResourceException

java.lang.Object  
    ▼ java.lang.Throwable  
        ▼ java.lang.Exception  
            ▼ javax.resource.ResourceException

All Implemented Interfaces:
    Serializable

Direct Known Subclasses:
    ApplicationServerInternalException, CommException,
    EISSystemException, IllegalStateException,
    InvalidPropertyException, LocalTransactionException,
    NotSupportedException, ResourceAdapterInternalException,
    ResourceAllocationException, ResourceWarning, SecurityException,
    SharingViolationException, UnavailableException, WorkException

public class ResourceException

extends Exception

Extends: Throwable > Exception
Extended by: ApplicationServerInternalException, CommException,
EISSystemException, IllegalStateException, InvalidPropertyException,
LocalTransactionException, NotSupportedException,
ResourceAdapterInternalException, ResourceAllocationException,
ResourceWarning, SecurityException, SharingViolationException,
UnavailableException, WorkException

Connector  ResourceException

- Java getMessage()
- 
- ResourceException J2SE  1.4

version  1.0
This is the root interface of the exception hierarchy defined for the Connector architecture. The ResourceException provides the following information:

- A resource adapter vendor specific string describing the error. This string is a standard Java exception message and is available through getMessage() method.
- resource adapter vendor specific error code.
- reference to another exception. Often a resource exception will be result of a lower level problem. If appropriate, this lower level exception can be linked to the ResourceException. Note, this has been deprecated in favor of J2SE release 1.4 exception chaining facility.

Version:
  1.0
Author:
  Rahul Sharma, Ram Jeyaraman
See Also:
  Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResourceException()</td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td>ResourceException(String message)</td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td>ResourceException(String message, String errorCode)</td>
<td>Create a new throwable with the specified message and error code.</td>
</tr>
<tr>
<td>ResourceException(String message, Throwable cause)</td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td>ResourceException(Throwable cause)</td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>
# Method Summary

<table>
<thead>
<tr>
<th>String getErrorCode()</th>
<th>Get the error code.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exception getLinkedException()</td>
<td>Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to getCause and initCause methods of the java.lang.Throwable class.</td>
</tr>
<tr>
<td>String getMessage()</td>
<td>Returns a detailed message string describing this exception.</td>
</tr>
<tr>
<td>void setErrorCode(String errorCode)</td>
<td>Set the error code.</td>
</tr>
<tr>
<td>void setLinkedException(Exception ex)</td>
<td>Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to getCause and initCause methods of the java.lang.Throwable class.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Throwable
- fillInStackTrace
- getCause
- getLocalizedMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

## Methods inherited from class java.lang.Object
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

## Constructor Detail

```java
public ResourceException()
null
```
ResourceException

public ResourceException()

    Constructs a new instance with null as its detail message.

public ResourceException(String message)

    message

ResourceException

public ResourceException(String message)

    Constructs a new instance with the specified detail message.

    Parameters:
        message - the detail message.

ResourceException

public ResourceException(Throwable cause)

    cause  throwable

    cause

ResourceException

public ResourceException(Throwable cause)

    Constructs a new throwable with the specified cause.

    Parameters:
        cause - a chained exception of type Throwable.
public ResourceException(String message, Throwable cause)
cause throwable
    message
    cause

ResourceException

public ResourceException(String message, Throwable cause)

    Constructs a new throwable with the specified detail message and cause.

    Parameters:
    message - the detail message.
    cause - a chained exception of type Throwable.

public ResourceException(String message, String errorCode)
throwable
    message
    errorCode

ResourceException

public ResourceException(String message, String errorCode)

    Create a new throwable with the specified message and error code.

    Parameters:
    message - a description of the exception.
    errorCode - a string specifying the vendor specific error code.
public void setErrorCode(String errorCode)

    errorCode

setErrorCode

public void setErrorCode(String errorCode)

    Set the error code.

Parameters:
    errorCode - the error code.

public String getErrorCode()

    return

gGetErrorCode

public String getErrorCode()

    Get the error code.

Returns:
    the error code.

public Exception getLinkedException()

    return Exception null

deprecated

    J2SE 1.4 thrower throwable

    java.lang.Throwable getCause

    initCause
getLinkedExceptionHandler

```java
public Exception getLinkedExceptionHandler()
```

**Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown.** Refer to `getCause` and `initCause` methods of the `java.lang.Throwable` class.

Get the exception linked to this ResourceException

**Returns:**
- linked Exception, null if none

---

setLinkedExceptionHandler

```java
public void setLinkedExceptionHandler(Exception ex)
```

**Exception ResourceException**

```java
ex Exception
```

**deprecated J2SE 1.4** throwable throwable

```java
java.lang.Throwable getCause initCause
```

Add a linked Exception to this ResourceException.

**Parameters:**
- ex - linked Exception
public String getMessage()

    return

getMessage

public String getMessage()

    Returns a detailed message string describing this exception.

    Overrides: getMessage in class Throwable

    Returns: 
        a detailed message string.
javax.annotation Annotation Type Resources

@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface Resources

Implementes: Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)

since 1.0
See also javax.annotation.Resource

This class is used to allow multiple resources declarations.

Since:
1.0
See Also:
Resource

Required Element Summary

<table>
<thead>
<tr>
<th>Resource[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Array used for multiple resource declarations.</td>
</tr>
</tbody>
</table>

Element Detail

abstract public Resource[] value()
value

public abstract Resource[] value

Array used for multiple resource declarations.
<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Field</th>
</tr>
</thead>
</table>

| Class | Method | Field |
javax.resource.cci  **Class ResourceWarning**

`java.lang.Object`  
  ▼ `java.lang.Throwable`  
    ▼ `java.lang.Exception`  
      ▼ `javax.resource.ResourceException`  
        ▼ `javax.resource.cci.ResourceWarning`

All Implemented Interfaces:  
  `Serializable`

---

**public class** ResourceWarning  
**extends** ResourceException

**Extends:** Throwable > Exception > **ResourceException**

**ResourceWarning** EIS

**See also**  
getWarnings

A `ResourceWarning` provides information on warnings related to execution of an interaction with an EIS. Warnings are silently chained to the object whose method caused it to be reported.

**See Also:**  
`Interaction.getWarnings()`, **Serialized Form**

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ResourceWarning()</strong></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><strong>ResourceWarning(String message)</strong></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><strong>ResourceWarning(String message, String errorCode)</strong></td>
<td>Constructs a new throwable with the specified detail message</td>
</tr>
</tbody>
</table>
and an error code.

<table>
<thead>
<tr>
<th>ResourceWarning(String message, Throwable cause)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ResourceWarning(Throwable cause)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

### Method Summary

**ResourceWarning**

<table>
<thead>
<tr>
<th>getLinkedWarning()</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deprecated.</strong> J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to <code>getCause</code> and <code>initCause</code> methods of the <code>java.lang Throwable</code> class.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>setLinkedWarning(ResourceWarning warning)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deprecated.</strong> J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to <code>getCause</code> and <code>initCause</code> methods of the <code>java.lang Throwable</code> class.</td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.resource.ResourceException**

getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException

**Methods inherited from class java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
**Constructor Detail**

```java
public ResourceWarning()
null

ResourceWarning

public ResourceWarning()

    Constructs a new instance with null as its detail message.

public ResourceWarning(String message)

    message

ResourceWarning

public ResourceWarning(String message)

    Constructs a new instance with the specified detail message.

    Parameters:
    message - the detail message.

public ResourceWarning(Throwable cause)

cause  throwable

ResourceWarning

public ResourceWarning(Throwable cause)
```
Constructs a new throwable with the specified cause.

**Parameters:**
- `cause` - a chained exception of type `Throwable`.

---

```java
public ResourceWarning(String message, Throwable cause)
```

**cause throwable**

- `message` - the detail message.
- `cause` - a chained exception of type `Throwable`.

---

```java
public ResourceWarning(String message, String errorCode)
```

**throwable**

- `message` - the detail message.
- `errorCode` - the error code.

---

```java
ResourceWarning
```

**Constructs a new throwable with the specified detail message and cause.**

**Parameters:**
- `message` - the detail message.
- `cause` - a chained exception of type `Throwable`.

---

```java
ResourceWarning
```

**Constructs a new throwable with the specified detail message and error code.**

**Parameters:**
- `message` - the detail message.
- `errorCode` - the error code.
Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.

Method Detail

public ResourceWarning getLinkedWarning()

ResourceWarning
return ResourceWarning
null

getLinkedWarning

public ResourceWarning getLinkedWarning()

getLinkedWarning

getLinkedWarning

Public. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to getCause and initCause methods of the java.lang.Throwable class.

Retrieves the warning chained to this ResourceWarning object.

Returns:
next ResourceWarning in the chain; null if none.

public void setLinkedWarning(ResourceWarning warning)

ResourceWarning

setLinkedWarning

setLinkedWarning

setLinkedWarning

Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to getCause and initCause methods of the java.lang.Throwable class.
setLinkedWarning

public void setLinkedWarning(ResourceWarning warning)

**Deprecated.** J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to `getCause` and `initCause` methods of the `java.lang.Throwable` class.

Adds an `ResourceWarning` object to the end of the chain.

**Parameters:**

- `warning` - `ResourceWarning` to be added to the chain.
javax.xml.ws Interface Response<T>

All Superinterfaces:
  Future<T>

public interface Response<T>
extends Future<T>

Implements: java.util.concurrent.Future<V>

Response

  get(...) cause RemoteException
  WebServiceException

since JAX-WS 2.0

The Response interface provides methods used to obtain the payload and context of a message sent in response to an operation invocation.

For asynchronous operation invocations it provides additional methods to check the status of the request. The get(...) methods may throw the standard set of exceptions and their cause may be a RemoteException or a WebServiceException that represents the error that occurred during the asynchronous method invocation.

Since:
  JAX-WS 2.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map&lt;String, Object&gt;</td>
</tr>
<tr>
<td>Gets the contained response context.</td>
</tr>
</tbody>
</table>
public java.util.Map<K, V> getContextMenu()

    return null

getContext

Map<String, Object> getContextMenu()

    Gets the contained response context.

    Returns:
    The contained response context. May be null if a response is not yet available.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>DETAIL</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
public abstract class ResponseStateManager
extends Object

ResponseStateManager is the helper class to StateManager that knows the specific rendering technology being used to generate the response. It is a singleton abstract class, vended by the RenderKit. This class knows the mechanics of saving state, whether it be in hidden fields, session, or some combination of the two.

Field Summary

| static String | RENDER_KIT_ID_PARAM | The name of the request parameter used by the default implementation of ViewHandler.calculateRenderKitId(javax.faces.context.FacesContext) to derive a RenderKit ID. |
| static String | VIEW_STATE_PARAM | Implementations must use this value as the name and id of the client parameter in which to save the state between requests. |

Constructor Summary

ResponseStateManager()
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getComponentStateToRestore(FacesContext context)</code></td>
<td><strong>Deprecated.</strong> This method has been replaced by <code>getState(javax.faces.context.FacesContext, java.lang.String)</code>. The default implementation returns <code>null</code>.</td>
</tr>
<tr>
<td><code>getState(FacesContext context, String viewId)</code></td>
<td>The implementation must inspect the current request and return an Object representing the tree structure and component state passed in to a previous invocation of <code>writeState(javax.faces.context.FacesContext, java.lang.Object)</code></td>
</tr>
<tr>
<td><code>getTreeStructureToRestore(FacesContext context, String viewId)</code></td>
<td><strong>Deprecated.</strong> This method has been replaced by <code>getState(javax.faces.context.FacesContext, java.lang.String)</code>. The default implementation returns <code>null</code>.</td>
</tr>
<tr>
<td><code>isPostback(FacesContext context)</code></td>
<td>Return true if the current request is a postback.</td>
</tr>
<tr>
<td><code>writeState(FacesContext context, Object state)</code></td>
<td></td>
</tr>
<tr>
<td><code>writeState(FacesContext context, StateManager.SerializedView state)</code></td>
<td><strong>Deprecated.</strong> This method has been replaced by <code>writeState(javax.faces.context.FacesContext, java.lang.Object)</code> The default implementation of this method does nothing.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`
The name of the request parameter used by the default implementation of ViewHandler.calculateRenderKitId(java.util.FacesContext) to derive a RenderKit ID.

See Also:
Constant Field Values

Implementations must use this value as the name and id of the client parameter in which to save the state between requests.

Since:
1.2
See Also:
Constant Field Values

Constructor Detail

public ResponseStateManager()

ResponseStateManager

public ResponseStateManager()
public void writeState(FacesContext context, Object state)
throws java.io.IOException

writeState

public void writeState(FacesContext context, Object state)
throws IOException

Throws:
IOException

public void writeState(FacesContext context, StateManager.SerializedView state) throws java.io.IOException

state ResponseWriter writer

javax.faces.application.StateManager.SerializedView

String HTTP

javax.faces.application.StateManager#STATE_SAVING_METHOD_CLI!

deprecated #writeState(javax.faces.context.FacesContext,javax.lang.Object)

context FacesContext

writeState

public void writeState(FacesContext context, StateManager.SerializedView state)
throws IOException

**Deprecated.** *This method has been replaced by* `writeState(javax.faces.context.FacesContext, java.lang.Object)`.* The default implementation of this method does nothing.*

Take the argument `state` and write it into the output using the current `ResponseWriter`, which must be correctly positioned already.

If the `StateManager.SerializedView` is to be written out to hidden fields, the implementation must take care to make all necessary character replacements to make the Strings suitable for inclusion as an HTTP request parameter.

If the state saving method for this application is `StateManager.STATE_SAVING_METHOD_CLIENT`, the implementation may encrypt the state to be saved to the client. We recommend that the state be unreadable by the client, and also be tamper evident. The reference implementation follows these recommendations.

**Parameters:**
- `context` - The `FacesContext` instance for the current request
- `state` - The serialized state information previously saved

**Throws:**
- IOException

```java
public Object getState(FacesContext context, String viewId)
```

**Object**

- `writeState(javax.faces.context.FacesContext, java.lang.Object)`
- `getTreeStructureToRestore`
- `getComponentStateToRestore`  
  - `ResponseStateManager`  
  - `getComponentStateToRestore`  
  - `SerializedView`

since 1.2
getState

public Object getState(FacesContext context, String viewId)

The implementation must inspect the current request and return an Object representing the tree structure and component state passed in to a previous invocation of writeState(FacesContext, java.lang.Object).

For backwards compatibility with existing ResponseStateManager implementations, the default implementation of this method calls getTreeStructureToRestore(javax.faces.context.FacesContext, java.lang.String) and getComponentStateToRestore(javax.faces.context.FacesContext) and creates and returns a two element Object array with element zero containing the structure property and element one containing the state property of the SerializedView.

Parameters:
context - The FacesContext instance for the current request
viewId - View identifier of the view to be restored

Returns:
the tree structure and component state Object passed in to writeState. If this is an initial request, this method returns null.

Since:
1.2

public Object getTreeStructureToRestore(FacesContext context, String viewId)

writeState() Object
getTreeStructureToRestore

```java
public Object getTreeStructureToRestore(FacesContext context, String viewId)
```

**Deprecated. This method has been replaced by**

The default implementation returns `null`.

The implementation must inspect the current request and return the tree structure Object passed to it on a previous invocation of `writeState()`.

**Parameters:**
- context - The `FacesContext` instance for the current request
- viewId - View identifier of the view to be restored

---

getComponentStateToRestore

```java
public Object getComponentStateToRestore(FacesContext context)
```

**Deprecated. This method has been replaced by**
The default implementation returns `null`.

The implementation must inspect the current request and return the component state object passed to it on a previous invocation of `writeState()`.

**Parameters:**

```java
public boolean isPostback(FacesContext context)
```

**postback true**

```java
javax.faces.application.ViewHandler#restoreView
javax.faces.application.ViewHandler#createView
ResponseStateManager postback true false
```

**HTML RenderKit**

```java
javax.faces.context.ExternalContext requestParameterMap #VIEW_STATE_PARAM true
```

**JSF 1.2**

```java
ResponseStateManager
javax.faces.context.ExternalContext requestParameterMap 0 true
```

since 1.2

**isPostback**

```java
public boolean isPostback(FacesContext context)
```

Return true if the current request is a postback. This method is leveraged from the `Restore View Phase` to determine if `ViewHandler.restoreView(javax.faces.context.FacesContext, java.lang.String)` or `ViewHandler.createView(javax.faces.context.FacesContext, java.lang.String)` should be called. The default implementation must return true if this `ResponseStateManager` instance wrote out
state on a previous request to which this request is a postback, false otherwise.

The implementation if this method for the Standard HTML RenderKit must consult the \texttt{ExternalContext}'s \texttt{requestParameterMap} and return \texttt{true} if and only if there is a key equal to the value of the symbolic constant \texttt{VIEW\_STATE\_PARAM}.

For backwards compatibility with implementations of \texttt{ResponseStateManager} prior to JSF 1.2, a default implementation is provided that consults the \texttt{ExternalContext}'s \texttt{requestParameterMap} and return \texttt{true} if its size is greater than 0.

\textbf{Since:} 1.2
javax.faces.context  **Class ResponseStream**

**java.lang.Object**

**java.io.OutputStream**

**javax.faces.context.ResponseStream**

**All Implemented Interfaces:**

[Closeable, Flushable]

---

```java
public abstract class ResponseStream
extends OutputStream
```

**Extends:** java.io.OutputStream

**ResponseStream**

**ResponseStream** is an interface describing an adapter to an underlying output mechanism for binary output.

---

**Constructor Summary**

`ResponseStream()`

---

**Method Summary**

**Methods inherited from class java.io.OutputStream**

`close, flush, write, write, write`

**Methods inherited from class java.lang.Object**

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,`

Constructor Detail

public ResponseStream()

ResponseStream

public ResponseStream()
javax.xml.ws  Annotation Type ResponseWrapper

@Target(value=METHOD)
@Retention(value=RUNTIME)
@Documented
public @interface ResponseWrapper

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)
@Documented

bean Response WebMethod localName
operationName targetNamespace SEI

Java className

since JAX-WS 2.0

See also javax.jws.WebMethod

Used to annotate methods in the Service Endpoint Interface with the response wrapper bean to be used at runtime. The default value of the localName is the operationName as defined in WebMethod annotation appended with Response and the targetNamespace is the target namespace of the SEI.

When starting from Java this annotation is used resolve overloading conflicts in document literal mode. Only the className is required in this case.

**Since:**
JAX-WS 2.0

**See Also:**
WebMethod
Optional Element Summary

<table>
<thead>
<tr>
<th>String className</th>
<th>Response wrapper bean name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String localName</td>
<td>Elements local name.</td>
</tr>
<tr>
<td>String targetNamespace</td>
<td>Elements namespace name.</td>
</tr>
</tbody>
</table>

abstract public String localName()

localName

public abstract String localName

    Elements local name.

    Default:"

abstract public String targetNamespace()

targetNamespace

public abstract String targetNamespace

    Elements namespace name.

    Default:"


abstract public String className()
bean

className

public abstract String className

Response wrapper bean name.

Default:

"

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.context  Class ResponseWriter

java.lang.Object
  ▼ java.io.Writer
    ▼ javax.faces.context.ResponseWriter

All Implemented Interfaces:
  Closeable, Flushable, Appendable

Direct Known Subclasses:
  ResponseWriterWrapper

public abstract class ResponseWriter
  extends Writer

  Extends: java.io.Writer
  Extended by: ResponseWriterWrapper

ResponseWriter  java.io.Writer  write()
 HTML  XML

ResponseWriter is an abstract class describing an adapter to an underlying output mechanism for character-based output. In addition to the low-level write() methods inherited from java.io.Writer, this class provides utility methods that are useful in producing elements and attributes for markup languages like HTML and XML.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class java.io.Writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock</td>
</tr>
</tbody>
</table>
## Constructor Summary

*ResponseWriter()*

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>abstract</strong> <code>ResponseWriter cloneWithWriter(Writer writer)</code></td>
<td>Create and return a new instance of this <code>ResponseWriter</code>, using the specified <code>Writer</code> as the output destination.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>endDocument()</code></td>
<td>Write whatever text should end a response.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>endElement(String name)</code></td>
<td>Write the end of an element, after closing any open element created by a call to <code>startElement()</code>.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>flush()</code></td>
<td>Flush any output buffered by the output method to the underlying Writer or OutputStream.</td>
</tr>
<tr>
<td><strong>abstract String getCharacterEncoding()</strong></td>
<td>Return the character encoding (such as &quot;ISO-8859-1&quot;) for this <code>ResponseWriter</code>.</td>
</tr>
<tr>
<td><strong>abstract String getContentType()</strong></td>
<td>Return the content type (such as &quot;text/html&quot;) for this <code>ResponseWriter</code>.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>startDocument()</code></td>
<td>Write whatever text should begin a response.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>startElement(String name, UIComponent component)</code></td>
<td>Write the start of an element, up to and including the element name.</td>
</tr>
<tr>
<td><strong>abstract void</strong> <code>writeAttribute(String name, Object value, String property)</code></td>
<td>Write an attribute name and corresponding value, after converting that text to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered.</td>
</tr>
<tr>
<td>Method</td>
<td>Signature</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td><code>writeComment</code></td>
<td><code>Object comment</code></td>
</tr>
<tr>
<td><code>writeText</code></td>
<td><code>char[] text, int off, int len</code></td>
</tr>
<tr>
<td><code>writeText</code></td>
<td><code>Object text, String property</code></td>
</tr>
<tr>
<td><code>writeText</code></td>
<td><code>Object text, UIComponent component, String property</code></td>
</tr>
<tr>
<td><code>writeURIAttribute</code></td>
<td><code>String name, Object value, String property</code></td>
</tr>
</tbody>
</table>

Methods inherited from class `java.io.Writer`:
- `append`, `append`, `append`, `close`, `write`, `write`, `write`, `write`, `write`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Constructor Detail**

public `ResponseWriter()`
ResponseWriter

public ResponseWriter()

Method Detail

abstract public String getContentType()

ResponseWriter "text/html" "charset="

getContentType

public abstract String getContentType()

Return the content type (such as "text/html") for this ResponseWriter. Note: this must not include the "charset=" suffix.

abstract public String getCharacterEncoding()

ResponseWriter "ISO-8859-1" the IANA

getCharacterEncoding

public abstract String getCharacterEncoding()

Return the character encoding (such as "ISO-8859-1") for this ResponseWriter. Please see the IANA for a list of character encodings.
abstract public void flush() throws java.io.IOException

Writer  OutputStream  Writer  OutputStream
ResponseWriter

flush

public abstract void flush()
    throws IOException

Flush any output buffered by the output method to the underlying Writer or OutputStream. This method will not flush the underlying Writer or OutputStream; it simply clears any values buffered by this ResponseWriter.

Specified by:
    flush in interface Flushable
Specified by:
    flush in class Writer
Throws:
    IOException

abstract public void startDocument() throws java.io.IOException

throws java.io.IOException: /

startDocument

public abstract void startDocument()
    throws IOException
Write whatever text should begin a response.

**Throws:**
- IOException - if an input/output error occurs

---

```java
abstract public void endDocument() throws java.io.IOException
```

**startElement()**

**Throws**
- java.io.IOException: /

---

**endDocument**

```java
public abstract void endDocument() throws IOException
```

Write whatever text should end a response. If there is an open element that has been created by a call to `startElement()`, that element will be closed first.

**Throws:**
- IOException - if an input/output error occurs

---

```java
abstract public void startElement(String name, UIComponent component) throws java.io.IOError
```

```java
writeAttribute() writeURIAttribute() startElement()writeComment()writeText()endElement()endDocument()write() "">
```

- `name`
- `component`

**Throws**
- java.io.IOException: /
startElement

public abstract void startElement(String name, UIComponent component) throws IOException

Write the start of an element, up to and including the element name. Once this method has been called, clients can call the writeAttribute() or writeURIAttribute() methods to add attributes and corresponding values. The starting element will be closed (that is, the trailing '>' character added) on any subsequent call to startElement(), writeComment(), writeText(), endElement(), endDocument(), close(), flush(), or write().

Parameters:

- name - Name of the element to be started
- component - The UIComponent (if any) to which this element corresponds

Throws:

- IOException - if an input/output error occurs
- NullPointerException - if name is null

abstract public void endElement(String name) throws java.io.IOException

startElement()
public abstract void endElement(String name) throws IOException

Write the end of an element, after closing any open element created by a call to startElement(). Elements must be closed in the inverse order from which they were opened; it is an error to do otherwise.

Parameters:
   name - Name of the element to be ended

Throws:
   IOException - if an input/output error occurs
   NullPointerException - if name is null

abstract public void writeAttribute(String name, Object value, String property) throws java.io.IOException

String startElement()

   name
   value
   property
   UIComponent

Throws
   IllegalStateException:
   java.io.IOException: /
   NullPointerException: name null

writeAttribute

public abstract void writeAttribute(String name, Object value, String property) throws IOException

Write an attribute name and corresponding value, after converting that text to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered. This method may only be called after a call to startElement(), and before the opened element has been closed.
Parameters:

- **name** - Attribute name to be added
- **value** - Attribute value to be added
- **property** - Name of the property or attribute (if any) of the UIComponent associated with the containing element, to which this generated attribute corresponds

Throws:

- **IllegalStateException** - if this method is called when there is no currently open element
- **IOException** - if an input/output error occurs
- **NullPointerException** - if name is null

abstract public void writeURIAttribute(String name, Object value, String property) throws java.io.IOException

String URI

```
startElement()
```

<table>
<thead>
<tr>
<th>name</th>
<th>value</th>
<th>property</th>
<th>Throws</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IllegalStateException:</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>java.io.IOException: /</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>NullPointerException: name null</td>
</tr>
</tbody>
</table>

writeURIAttribute

public abstract void writeURIAttribute(String name, Object value, String property) throws IOException

Write a URI attribute name and corresponding value, after converting that text to a String (if necessary), and after performing any encoding appropriate to the markup language being rendered. This method may only be called after a call to startElement(), and before the opened element has been closed.
Parameters:
  name - Attribute name to be added
  value - Attribute value to be added
  property - Name of the property or attribute (if any) of the UIComponent associated with the containing element, to which this generated attribute corresponds

Throws:
  IllegalStateException - if this method is called when there is no currently open element
  IOException - if an input/output error occurs
  NullPointerException - if name is null

abstract public void writeComment(Object comment)
throws java.io.IOException

String startElement()

comment

Throws java.io.IOException: /

Throws NullPointerException: comment null

writeComment

public abstract void writeComment(Object comment)
throws IOException

Write a comment containing the specified text, after converting that text to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered. If there is an open element that has been created by a call to startElement(), that element will be closed first.

Parameters:
  comment - Text content of the comment

Throws:
  IOException - if an input/output error occurs
abstract public void writeText(Object text, String property)
throws java.io.IOException

Parameters:
  text - Text to be written
  property - Name of the property or attribute (if any) of the UIComponent associated with the containing element, to which this generated text corresponds

Throws:
  IOException - if an input/output error occurs
  NullPointerException - if text is null

public void writeText(Object text, UIComponent component, String property) throws java.io.IOException
String

```java
public void writeText(Object text, UIComponent component, String property)
```

Parameters:
- `text` - Text to be written
- `component` - The `UIComponent` (if any) to which this element corresponds
- `property` - Name of the property or attribute (if any) of the `UIComponent` associated with the containing element, to which
abstract public void writeText(char[] text, int off, int len) throws java.io.IOException

   writeText

   public abstract void writeText(char[] text,
       int off,
       int len)
   throws IOException

Write text from a character array, after any performing any escaping appropriate for the markup language being rendered. If there is an open element that has been created by a call to startElement(), that element will be closed first.

Parameters:
   text - Text to be written
   off - Starting offset (zero-relative)
   len - Number of characters to be written

Throws:
**IndexOutOfBoundsException** - if the calculated starting or ending position is outside the bounds of the character array

**IOException** - if an input/output error occurs

**NullPointerException** - if text is null

---

**abstract public ResponseWriter cloneWithWriter(java.io.Writer writer)**

**Parameters:**

- **writer** - The **Writer** that is the output destination

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.faces.context  Class ResponseWriterWrapper


All Implemented Interfaces:
   Closeable, Flushable, Appendable

public abstract class ResponseWriterWrapper
extends ResponseWriter

Extends: java.io.Writer > ResponseWriter

   ResponseWriter       ResponseWriter       ResponseWriter
   #getWrapped

   since                1.2

Provides a simple implementation of ResponseWriter that can be subclassed by developers wishing to provide specialized behavior to an existing ResponseWriter instance. The default implementation of all methods is to call through to the wrapped ResponseWriter.

Usage: extend this class and override getWrapped() to return the instance we are wrapping.

Since:
   1.2

Field Summary
### Fields inherited from class java.io.Writer

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>lock</td>
</tr>
</tbody>
</table>

### Constructor Summary

**ResponseWriterWrapper()**

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cloneWithWriter(Writer writer)</td>
<td>The default behavior of this method is to call ResponseWriter.cloneWithWriter(java.io.Writer) on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>close()</td>
<td>The default behavior of this method is to call Writer.close() on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>endDocument()</td>
<td>The default behavior of this method is to call ResponseWriter.endDocument() on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>endElement(String name)</td>
<td>The default behavior of this method is to call ResponseWriter.endElement(String) on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>flush()</td>
<td>The default behavior of this method is to call ResponseWriter.flush() on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>getCharacterEncoding()</td>
<td>The default behavior of this method is to call ResponseWriter.getCharacterEncoding() on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>getContentType()</td>
<td>The default behavior of this method is to call ResponseWriter.getContentType() on the wrapped ResponseWriter object.</td>
</tr>
</tbody>
</table>
String ResponseWriter.getContentType() on the wrapped ResponseWriter object.

protected abstract ResponseWriter getWrapped()

void startDocument()
   The default behavior of this method is to call ResponseWriter.startDocument() on the wrapped ResponseWriter object.

void startElement(String name, UIComponent component)
   The default behavior of this method is to call ResponseWriter.startElement(String, javax.faces.component.UIComponent) on the wrapped ResponseWriter object.

void write(char[] cbuf, int off, int len)
   The default behavior of this method is to call Writer.write(char[], int, int) on the wrapped ResponseWriter object.

void writeAttribute(String name, Object value, String property)
   The default behavior of this method is to call ResponseWriter.writeAttribute(String, Object, String) on the wrapped ResponseWriter object.

void writeComment(Object comment)
   The default behavior of this method is to call ResponseWriter.writeComment(Object) on the wrapped ResponseWriter object.

void writeText(char[] text, int off, int len)
   The default behavior of this method is to call ResponseWriter.writeText(char[], int, int) on the wrapped ResponseWriter object.

void writeText(Object text, String property)
   The default behavior of this method is to call ResponseWriter.writeText(Object, String) on the wrapped ResponseWriter object.

void writeText(Object text, UIComponent component, String property)
   The default behavior of this method is to call
<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResponseWriter.writeText(Object, UIComponent, String) on the wrapped ResponseWriter object.</td>
</tr>
<tr>
<td>writeURIAttribute(String name, Object value, String property)</td>
</tr>
<tr>
<td>The default behavior of this method is to call ResponseWriter.writeURIAttribute(String, Object, String) on the wrapped ResponseWriter object.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.io.Writer**

append, append, append, write, write, write, write

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

public ResponseWriterWrapper()

**ResponseWriterWrapper**

public ResponseWriterWrapper()

**Method Detail**

abstract protected ResponseWriter getWrapped()

return

getWrapped

protected abstract ResponseWriter getWrapped()
Returns:
the instance that we are wrapping.

public String getContentType()

    ResponseWriter getContentType()

    since 1.2
    See also getContentType()

getContentType

public String getContentType()

    The default behavior of this method is to call
    ResponseWriter.getContentType() on the wrapped ResponseWriter
    object.

    Specified by:
    getContentType in class ResponseWriter

    Since:
    1.2

    See Also:
    ResponseWriter.getContentType()

public String getCharacterEncoding()

    ResponseWriter getCharacterEncoding()

    since 1.2
    See also getCharacterEncoding()

getCharacterEncoding
public String getCharacterEncoding()

The default behavior of this method is to call ResponseWriter.getCharacterEncoding() on the wrapped ResponseWriter object.

Specified by: getCharacterEncoding in class ResponseWriter
Since: 1.2
See Also: ResponseWriter.getCharacterEncoding()

--------------------------------------------------------------------------------

public void flush() throws java.io.IOException

ResponseWriter flush()

since 1.2
See also flush()

flush

public void flush() throws IOException

The default behavior of this method is to call ResponseWriter.flush() on the wrapped ResponseWriter object.

Specified by: flush in interface Flushable
Specified by: flush in class ResponseWriter
Throws: IOException
Since: 1.2
See Also: ResponseWriter.flush()
public void startDocument() throws java.io.IOException

ResponseWriter startDocument()

since 1.2
See also startDocument()

startDocument

class:
public void startDocument()
throws IOException

The default behavior of this method is to call

Specified by:
startDocument in class ResponseWriter

Throws:
IOException - if an input/output error occurs

Since:
1.2

See Also:
ResponseWriter.startDocument()

public void endDocument() throws java.io.IOException

ResponseWriter endDocument()

since 1.2
See also endDocument()

endDocument
public void endDocument()  
throws IOException

The default behavior of this method is to call
ResponseWriter.endDocument() on the wrapped ResponseWriter
object.

Specified by:
    endDocument in class ResponseWriter

Throws:
    IOException - if an input/output error occurs

Since:
    1.2

See Also:
    ResponseWriter.endDocument()

public void startElement(String name, UIComponent component)
throws java.io.IOException

ResponseWriter startElement(String, javax.faces.component.UIComponent)

since
1.2

See also startElement(String, javax.faces.component.UIComponent)

startElement

public void startElement(String name, UIComponent component)
throws IOException

The default behavior of this method is to call
ResponseWriter.startElement(String, javax.faces.component.UIComponent) on the wrapped ResponseWriter
object.

Specified by:
**startElement** in class `ResponseWriter`

**Parameters:**
- `name` - Name of the element to be started
- `component` - The `UIComponent` (if any) to which this element corresponds

**Throws:**
- `IOException` - if an input/output error occurs

**Since:**
1.2

**See Also:**
- `ResponseWriter.startElement(String, javax.faces.component.UIComponent)`

```java
class ResponseWriter {
    public void endElement(String name) throws IOException {
        // The default behavior of this method is to call
        ResponseWriter.endElement(String) on the wrapped ResponseWriter object.
    }
}
```

**endElement**

```java
public void endElement(String name)
    throws IOException
```

**Specified by:**
- `endElement` in class `ResponseWriter`

**Parameters:**
- `name` - Name of the element to be ended

**Throws:**
- `IOException` - if an input/output error occurs

**Since:**
public void writeAttribute(String name, Object value, String property) throws java.io.IOException

Since: 1.2
See also: writeAttribute(String, Object, String)

writeAttribute

public void writeAttribute(String name,
Object value,
String property)
throws IOException

The default behavior of this method is to call ResponseWriter.writeAttribute(String, Object, String) on the wrapped ResponseWriter object.

Specified by:
writeAttribute in class ResponseWriter

Parameters:
name - Attribute name to be added
value - Attribute value to be added
property - Name of the property or attribute (if any) of the UICOMPONENT associated with the containing element, to which this generated attribute corresponds

Throws:
IOException - if an input/output error occurs

Since: 1.2
See Also:
ResponseWriter.writeAttribute(String, Object, String)
public void writeURIAttribute(String name, Object value, String property) throws IOException

Since 1.2
See also writeURIAttribute(String, Object, String)

writeURIAttribute

public void writeURIAttribute(String name, Object value, String property) throws IOException

The default behavior of this method is to call ResponseWriter.writeURIAttribute(String, Object, String) on the wrapped ResponseWriter object.

Specified by:
writeURIAttribute in class ResponseWriter

Parameters:
name - Attribute name to be added
value - Attribute value to be added
property - Name of the property or attribute (if any) of the UIComponent associated with the containing element, to which this generated attribute corresponds

Throws:
IOException - if an input/output error occurs

Since 1.2
See Also:
ResponseWriter.writeURIAttribute(String, Object, String)
public void writeComment(Object comment) throws IOException

ResponseWriter writeComment(Object)

since 1.2
See also writeComment(Object)

writeComment

public void writeComment(Object comment)
throws IOException

The default behavior of this method is to call
ResponseWriter.writeComment(Object) on the wrapped
ResponseWriter object.

Specified by:
writeComment in class ResponseWriter

Parameters:
comment - Text content of the comment

Throws:
IOException - if an input/output error occurs

Since:
1.2

See Also:
ResponseWriter.writeComment(Object)

public void writeText(Object text, String property) throws IOException

ResponseWriter writeText(Object, String)

since 1.2
See also writeText(Object, String)
writeText

public void writeText(Object text, String property)
throws IOException

The default behavior of this method is to call
ResponseWriter.writeText(Object, String) on the wrapped
ResponseWriter object.

Specified by:
   writeText in class ResponseWriter

Parameters:
   text - Text to be written
   property - Name of the property or attribute (if any) of the
   UIComponent associated with the containing element, to which
   this generated text corresponds

Throws:
   IOException - if an input/output error occurs

Since:
   1.2

See Also:
   ResponseWriter.writeText(Object, String)

public void writeText(Object text, UIComponent component, String property) throws java.io.IOException

ResponseWriter writeText(Object, UIComponent, String)

since
   1.2

See also
   writeText(Object, String)

writeText

public void writeText(Object text, UIComponent component, String property)
throws IOException

The default behavior of this method is to call
ResponseWriter.writeText(Object, UIComponent, String) on the
wrapped ResponseWriter object.

Overrides:
   writeText in class ResponseWriter

Parameters:
   text - Text to be written
   component - The UIComponent (if any) to which this element
               corresponds
   property - Name of the property or attribute (if any) of the
              UIComponent associated with the containing element, to which
              this generated text corresponds

Throws:
   IOException - if an input/output error occurs

Since: 1.2

See Also:
   ResponseWriter.writeText(Object, String)

public void writeText(char[] text, int off, int len) throws
java.io.IOException

   ResponseWriter   writeText(char[], int, int)

   since            1.2
   See also         writeText(char[], int, int)

writeText

public void writeText(char[] text, int off, int len)
   throws IOException
The default behavior of this method is to call `ResponseWriter.writeText(char[], int, int)` on the wrapped `ResponseWriter` object.

**Specified by:**
`writeText` in class `ResponseWriter`

**Parameters:**
- `text` - Text to be written
- `off` - Starting offset (zero-relative)
- `len` - Number of characters to be written

**Throws:**
- `IOException` - if an input/output error occurs

**Since:**
1.2

**See Also:**
`ResponseWriter.writeText(char[], int, int)`

```java
public ResponseWriter cloneWithWriter(java.io.Writer writer)
```

**ResponseWriter**  
`cloneWithWriter(java.io.Writer)`

- **since**
  1.2
- **See also**
  `cloneWithWriter(java.io.Writer)`

`cloneWithWriter`

```java
public ResponseWriter cloneWithWriter(Writer writer)
```

The default behavior of this method is to call `ResponseWriter.cloneWithWriter(java.io.Writer)` on the wrapped `ResponseWriter` object.

**Specified by:**
`cloneWithWriter` in class `ResponseWriter`

**Parameters:**
- `writer` - Writer to be used for writing
public void close() throws java.io.IOException

ResponseWriter close()

since 1.2
See also close()

close

public void close() throws IOException

The default behavior of this method is to call Writer.close() on the wrapped ResponseWriter object.

Specified by:
   close in interface Closeable
Specified by:
   close in class Writer
Throws:
   IOException
Since: 1.2
See Also:
   Writer.close()

public void write(char[] cbuf, int off, int len) throws java.io.IOException

ResponseWriter write(char[], int, int)
write

public void write(char[] cbuf,
        int off,
        int len)
    throws IOException

The default behavior of this method is to call Writer.write(char[],
        int, int) on the wrapped ResponseWriter object.

Specified by: write in class Writer

Throws: IOException

Since: 1.2

See Also: Writer.write(char[], int, int)
javax.faces.model Class ResultDataModel

java.lang.Object
   javax.faces.model.DataModel
   javax.faces.model.ResultDataModel

public class ResultDataModel
extends DataModel

Extends: DataModel

ResultDataModel is a convenience implementation of DataModel that wraps a JSTL Result object, typically representing the results of executing an SQL query via JSTL tags.

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultDataModel()</td>
</tr>
<tr>
<td>Construct a new ResultDataModel with no specified wrapped data.</td>
</tr>
<tr>
<td>ResultDataModel(javax.servlet.jsp.jstl.sql.Result result)</td>
</tr>
<tr>
<td>Construct a new ResultDataModel wrapping the specified Result.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getRowCount()</td>
</tr>
<tr>
<td>If there is wrappedData available, return the length of the array returned by calling getRows() on the underlying Result.</td>
</tr>
<tr>
<td>SortedMap getRowData()</td>
</tr>
<tr>
<td>If row data is available, return the SortedMap array element</td>
</tr>
</tbody>
</table>
at the index specified by \texttt{rowIndex} of the array returned by calling \texttt{getRows()} on the underlying \texttt{Result}.

\begin{verbatim}

Object

getRowIndex()  
   Return the zero-relative index of the currently selected row.

getWrappedData()  
   Return the object representing the data wrapped by this \texttt{DataModel}, if any.

boolean isRowAvailable()  
   Return \texttt{true} if there is \texttt{wrappedData} available, and the current value of \texttt{rowIndex} is greater than or equal to zero, and less than the length of the array returned by calling \texttt{getRows()} on the underlying \texttt{Result}.

void setRowIndex(int rowIndex)  
   Set the zero-relative index of the currently selected row, or \texttt{-1} to indicate that we are not positioned on a row.

void setWrappedData(Object data)  
   Set the object representing the data collection wrapped by this \texttt{DataModel}.

Methods inherited from class \texttt{javax.faces.model.DataModel}
addDataModelListener, getDataModelListeners, removeDataModelListener

Methods inherited from class \texttt{java.lang.Object}
code, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

\end{verbatim}

Constructor Detail

public \texttt{ResultDataModel()}

\texttt{ResultDataModel}
ResultDataModel

public ResultDataModel()

    Construct a new ResultDataModel with no specified wrapped data.

public ResultDataModel(javax.servlet.jsp.jstl.sql.Result result)

    Result ResultDataModel

        result Result

ResultDataModel

public ResultDataModel(javax.servlet.jsp.jstl.sql.Result result)

    Construct a new ResultDataModel wrapping the specified Result.

Parameters:

    result - Result to be wrapped (if any)

Method Detail

public boolean isRowAvailable()

    wrappedData rowIndex 0 Result getRows()

    true false

    Throws FacesException:
isRowAvailable

public boolean isRowAvailable()

Return true if there is wrappedData available, and the current value of rowIndex is greater than or equal to zero, and less than the length of the array returned by calling getRows() on the underlying Result. Otherwise, return false.

Specified by: isRowAvailable in class DataModel
Throws: FacesException - if an error occurs getting the row availability

public int getRowCount()

wrappedData Result getRows() wrappedData -1

Throws FacesException:

getRowCount

public int getRowCount()

If there is wrappedData available, return the length of the array returned by calling getRows() on the underlying Result. If no wrappedData is available, return -1.

Specified by: getRowCount in class DataModel
Throws: FacesException - if an error occurs getting the row count

public Object getRowData()
```java
SortedMap getRowData() {
    if (row data is available, return the SortedMap array element at the
    index specified by rowIndex of the array returned by calling getRows()
    on the underlying Result. If no wrapped data is available, return
    null.

    Note that, if a non-null Map is returned by this method, it will contain
    the values of the columns for the current row, keyed by column
    name. Column name comparisons must be performed in a case-
    insensitive manner.

    Specified by:
        getRowData in class DataModel
    Throws:
        FacesException - if an error occurs getting the row data
        IllegalArgumentException - if now row data is available at the
        currently specified row index
}
```

```java
public int getRowIndex() {
    Throws FacesException: NullPointerException
}
```

```java
getRowIndex
```
Description copied from class: **DataModel**

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no wrappedData is available, return -1.

**Specified by:**

   getRowIndex in class **DataModel**

**Throws:**

   FacesException - if an error occurs getting the row index

---

**public void setRowIndex(int rowIndex)**

**Throws**

   FacesException: NullPointerException

   IllegalArgumentException: NullPointerException  

   IllegalArgumentException: rowIndex -1

**setRowIndex**

**public void setRowIndex(int rowIndex)**

Description copied from class: **DataModel**

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the isRowAvailable() method to detect whether row data will be available for use by the getRowData() method.

If there is no wrappedData available when this method is called, the specified rowIndex is stored (and may be retrieved by a subsequent call to getRowData()), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a DataModelEvent will be sent to the rowSelected() method of all registered DataModelListeners.

**Specified by:**
**setRowIndex** in class **DataModel**

**Parameters:**
- **rowIndex** - The new zero-relative index (must be non-negative)

**Throws:**
- **FacesException** - if an error occurs setting the row index
- **IllegalArgumentException** - if **rowIndex** is less than -1

---

**public Object getWrappedData()**

**getWrappedData**

**public Object getWrappedData()**

**Description copied from class: **DataModel**

Return the object representing the data wrapped by this **DataModel**, if any.

**Specified by:**
- **getWrappedData** in class **DataModel**

---

**public void setWrappedData(Object data)**

**Throws**
- ClassCastException: data null Result

**setWrappedData**

**public void setWrappedData(Object data)**

**Description copied from class: **DataModel**

Set the object representing the data collection wrapped by this **DataModel**. If the specified **data** is null, detach this **DataModel** from any previously wrapped data collection instead.

If **data** is non-null, the currently selected row index must be set to
zero, and a DataModelEvent must be sent to the rowSelected() method of all registered DataModelListener indicating that this row is now selected.

**Specified by:**

setWrappedData in class DataModel

**Parameters:**

data - Data collection to be wrapped, or null to detach from any previous data collection

**Throws:**

ClassCastException - if data is non-null and is not a Result
public interface ResultSet
extends Record, ResultSet

Implements: Record, java.sql.ResultSet

ResultSet Interaction  EIS CCI ResultSet  JDBC ResultSet

CCI ResultSet  Connectors 1.0  CCI
since 0.8
See also java.sql.ResultSet

A ResultSet represents tabular data that is retrieved from an EIS instance by the execution of an Interaction. The CCI ResultSet is based on the JDBC ResultSet.

Refer the CCI specification in Connectors 1.0 for detailed requirements on the implementation of a CCI ResultSet.

Since: 0.8
Author: Rahul Sharma
See Also: ResultSet
Fields inherited from interface java.sql.ResultSet
CLOSE_CURSORS_AT_COMMIT, CONCUR_READ_ONLY, CONCUR_UPDATABLE,
FETCH_FORWARD, FETCH_REVERSE, FETCH_UNKNOWN,
HOLD_CURSORS_OVER_COMMIT, TYPE_FORWARD_ONLY,
TYPE_SCROLL_INSENSITIVE, TYPE_SCROLL_SENSITIVE

Method Summary

Methods inherited from interface javax.resource.cci.Record
clone, equals, getRecordName, getRecordShortDescription, hashCode,
setRecordName, setRecordShortDescription

Methods inherited from interface java.sql.ResultSet
absolute, afterLast, beforeFirst, cancelRowUpdates, clearWarnings,
close, deleteRow, findColumn, first, getArray, getArray,
getAsciiStream, getAsciiStream, getBigDecimal, getBigDecimal,
getBigDecimal, getBigDecimal, getBinaryStream, getBinaryStream,
getBlob, getBlob, getBoolean, getBoolean, getByte, getByte,
getBytes, getBytes, getCharacterStream, getCharacterStream,
getClob, getClob, getConcurrent, getCursorName, getDate, getDate,
getDate, getDate, getDouble, getDouble, getFetchDirection,
getFetchSize, getFloat, getFloat, getInt, getInt, getLong,
getLong, getMetaData, getObject, getObject, getObject, getObject,
getRef, getRef, getRow, getShort, getShort, getStatement,
getString, getString, getTime, getTime, getTime,
getTime, getTimestamp, getTimestamp, getTimestamp, getType,
getUnicodeStream, getUnicodeStream, getURL, getURL, getWarnings,
insertRow, isAfterLast, isFirst, isBeforeFirst, isFirst, isLast, last,
moveToCurrentRow, moveToInsertRow, next, previous, refreshRow,
relative, rowDeleted, rowInserted, rowUpdated, setFetchDirection,
setFetchSize, updateArray, updateArray, updateAsciiStream,
updateAsciiStream, updateBigDecimal, updateBigDecimal,
updateBinaryStream, updateBinaryStream, updateBlob, updateBlob,
updateBoolean, updateBoolean, updateByte, updateByte, updateBytes,
updateBytes, updateCharacterStream, updateCharacterStream,
updateClob, updateClob, updateDate, updateDouble,
updateDouble, updateFloat, updateFloat, updateInt, updateInt,
updateLong, updateLong, updateNull, updateNull, updateObject,
updateObject, updateObject, updateObject, updateObject, updateRef,
updateRef, updateRef, updateRow, updateShort, updateShort, updateString,
updateString, updateString, updateTime, updateTime, updateTime, updateTimestamp, updateTimestamp, wasNull
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.model Class ResultSetDataModel

java.lang.Object
   ↓ javax.faces.model.DataModel
      ↓ javax.faces.model.ResultSetDataModel

public class ResultSetDataModel

extends DataModel

Extends: DataModel

ResultSetDataModel is a convenience implementation of DataModel that wraps a ResultSet of Java objects. Note that the specified ResultSet MUST be scrollable. In addition, if input components (that will be updating model values) reference this object in value binding expressions, the specified ResultSet MUST be updatable.

Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResultSetDataModel()</td>
<td>Construct a new ResultSetDataModel with no specified wrapped data.</td>
</tr>
<tr>
<td>ResultSetDataModel(ResultSet resultSet)</td>
<td>Construct a new ResultSetDataModel wrapping the specified ResultSet.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>getRowCount()</td>
</tr>
</tbody>
</table>
Return -1, since ResultSet does not provide a standard way to determine the number of available rows without scrolling through the entire ResultSet, and this can be very expensive if the number of rows is large.

Object
getRowData()
If row data is available, return a Map representing the values of the columns for the row specified by rowIndex, keyed by the corresponding column names.

int
getRowIndex()
Return the zero-relative index of the currently selected row.

Object
getWrappedData()
Return the object representing the data wrapped by this DataModel, if any.

boolean
isRowAvailable()
Return true if there is wrappedData available, and the result of calling absolute() on the underlying ResultSet, passing the current value of rowIndex plus one (to account for the fact that ResultSet uses one-relative indexing), returns true.

void
setRowIndex(int rowIndex)
Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row.

void
setWrappedData(Object data)
Set the object representing the data collection wrapped by this DataModel.

Methods inherited from class javax.faces.model.DataModel
addDataModelListener, getDataModelListeners, removeDataModelListener

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail
public ResultSetDataModel()

        ResultSetDataModel

ResultSetDataModel

public ResultSetDataModel()

        Construct a new ResultSetDataModel with no specified wrapped data.

public ResultSetDataModel(java.sql.ResultSet resultSet)

        ResultSet         ResultSetDataModel

        resultSet         ResultSet

ResultSetDataModel

public ResultSetDataModel(ResultSet resultSet)

        Construct a new ResultSetDataModel wrapping the specified ResultSet.

        Parameters:
        resultSet - ResultSet to be wrapped (if any)

Method Detail

public boolean isRowAvailable()
**isRowAvailable**

```java
public boolean isRowAvailable()
```

Return `true` if there is wrappedData available, and the result of calling `absolute()` on the underlying `ResultSet`, passing the current value of `rowIndex` plus one (to account for the fact that `ResultSet` uses one-relative indexing), returns `true`. Otherwise, return `false`.

**Specified by:**

`isRowAvailable` in class `DataModel`

**Throws:**

`FacesException` - if an error occurs getting the row availability

---

**public int getRowCount()**

```java
-1  ResultSet  ResultSet
```

**Throws**

`FacesException`:

---

**getRowCount**

```java
public int getRowCount()
```

Return `-1`, since `ResultSet` does not provide a standard way to determine the number of available rows without scrolling through the entire `ResultSet`, and this can be very expensive if the number of rows is large.

**Specified by:**

`getRowCount` in class `DataModel`
Throws:

* FacesException - if an error occurs getting the row count

public Object getRowData()

    Map rowIndex null

null Map AbstractMap JavaDoc Map

• Map Comparator String.CASE_INSENSITIVE_ORDER

• UnsupportedOperationException clear() remove()
• entrySet() Set
  ◦ Set UnsupportedOperationException Set
    Iterator
  ◦ set value ResultSet
• keySet() Set UnsupportedOperationException Set
    Iterator
• containsKey() false put()
  InvalidArgumentException Map ResultSet
• values() Collection UnsupportedOperationException
    Collection Iterator

Throws       FacesException:
Throws       InvalidArgumentException:

getRowData

public Object getRowData()

If row data is available, return a Map representing the values of the columns for the row specified by rowIndex, keyed by the corresponding column names. If no wrapped data is available, return null.
If a non-null Map is returned, its behavior must correspond to the contract for a mutable Map as described in the JavaDocs for AbstractMap, with the following exceptions and specialized behavior:

- The Map, and any supporting objects it returns, must perform all column name comparisons in a case-insensitive manner. This case-insensitivity must be implemented using a case-insensitive Comparator, such as String.CASE_INSENSITIVE_ORDER.
- The following methods must throw UnsupportedOperationException: clear(), remove().
- The entrySet() method must return a Set that has the following behavior:
  - Throw UnsupportedOperationException for any attempt to add or remove entries from the Set, either directly or indirectly through an Iterator returned by the Set.
  - Updates to the value of an entry in this set must write through to the corresponding column value in the underlying ResultSet.
- The keySet() method must return a Set that throws UnsupportedOperationException on any attempt to add or remove keys, either directly or through an Iterator returned by the Set.
- The put() method must throw IllegalArgumentException if a key value for which containsKey() returns false is specified. However, if a key already present in the Map is specified, the specified value must write through to the corresponding column value in the underlying ResultSet.
- The values() method must return a Collection that throws UnsupportedOperationException on any attempt to add or remove values, either directly or through an Iterator returned by the Collection.

Specified by:
getRowData in class DataModel

Throws:
FacesException - if an error occurs getting the row data
IllegalArgumentException - if now row data is available at the currently specified row index
public int getRowIndex()

Throws  FacesException: NullPointerException

gRowIndex

public int getRowIndex()

Description copied from class: DataModel

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no wrappedData is available, return -1.

Specified by:
getRowIndex in class DataModel

Throws:
FacesException - if an error occurs getting the row index

public void setRowIndex(int rowIndex)

Throws  FacesException: NullPointerException

Throws  IllegalArgumentException: NullPointerException  
rowIndex -1

setRowIndex

public void setRowIndex(int rowIndex)

Description copied from class: DataModel

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the isRowAvailable() method to detect whether row data will be available for use by the getRowData() method.
If there is no wrappedData available when this method is called, the specified rowIndex is stored (and may be retrieved by a subsequent call to getRowData()), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a DataModelEvent will be sent to the rowSelected() method of all registered DataModelListener.

Specified by:
    setRowIndex in class DataModel

Parameters:
    rowIndex - The new zero-relative index (must be non-negative)

Throws:
    FacesException - if an error occurs setting the row index
    IllegalArgumentException - if rowIndex is less than -1

public Object getWrappedData()

getWrappedData

public Object getWrappedData()

    Description copied from class: DataModel

    Return the object representing the data wrapped by this DataModel, if any.

    Specified by:
    getWrappedData in class DataModel

public void setWrappedData(Object data)

    Throws ClassCastException: NullPointerException  

setWrappedData
public void setWrappedData(Object data)

**Description copied from class: DataModel**

Set the object representing the data collection wrapped by this DataModel. If the specified data is null, detach this DataModel from any previously wrapped data collection instead.

If data is non-null, the currently selected row index must be set to zero, and a DataModelEvent must be sent to the rowSelected() method of all registered DataModelListener indicating that this row is now selected.

**Specified by:**

setWrappedData in class DataModel

**Parameters:**

data - Data collection to be wrapped, or null to detach from any previous data collection

**Throws:**

ClassCastException - if data is not of the appropriate type for this DataModel implementation
The interface `javax.resource.cci.ResultSetInfo` provides information on the support provided for ResultSet by a connected EIS instance. A component calls the method `Connection.getResultInfo` to get the `ResultSetInfo` instance.

A CCI implementation is not required to support `javax.resource.cci.ResultSetInfo` interface. The implementation of this interface is provided only if the CCI supports the ResultSet facility.

Version: 0.9
Author: Rahul Sharma

See Also:  
`Connection`, `ResultSet`, `ConnectionMetaData`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean deletesAreDetected(int type)</code></td>
<td></td>
</tr>
</tbody>
</table>
boolean insertsAreDetected(int type)
Indicates whether or not a visible row insert can be
detected by calling ResultSet.rowInserted.

boolean othersDeletesAreVisible(int type)
Indicates whether deletes made by others are visible.

boolean othersInsertsAreVisible(int type)
Indicates whether inserts made by others are visible.

boolean othersUpdatesAreVisible(int type)
Indicates whether updates made by others are visible.

boolean ownDeletesAreVisible(int type)

boolean ownInsertsAreVisible(int type)

boolean ownUpdatesAreVisible(int type)

boolean supportsResultSetType(int type)
Indicates whether or not a resource adapter supports a
type of ResultSet.

boolean supportsResultSetConcurrency(int type, int concurrency)
Indicates whether or not a resource adapter supports the
concurrency type in combination with the given ResultSet type/

boolean updatesAreDetected(int type)
Indicates whether or not a visible row update can be
detected by calling the method ResultSet.rowUpdated.

**Method Detail**

public boolean updatesAreDetected(int type) throws ResourceException

Resultset.rowUpdated

typtype Resultset.ResultSet.TYPE_XXX
return true false

Throws ResourceException:
updatesAreDetected

boolean updatesAreDetected(int type) throws ResourceException

Indicates whether or not a visible row update can be detected by calling the method ResultSet.rowUpdated.

Parameters:
- type - type of the ResultSet i.e. ResultSet.TYPE_XXX

Returns:
- true if changes are detected by the result set type; false otherwise

Throws:
- ResourceException - Failed to get the information

See Also:
- ResultSet.rowUpdated()
Parameters:
  type - type of the ResultSet i.e. ResultSet.TYPE_XXX

Returns:
  true if changes are detected by the result set type; false otherwise

Throws:
  ResourceException - Failed to get the information

See Also:
  ResultSet.rowInserted()

---

public boolean deletesAreDetected(int type) throws ResourceException
deltesAreDetected

boolean deletesAreDetected(int type)
throws ResourceException

Throws:
  ResourceException

---

public boolean supportsResultSetType(int type) throws ResourceException
ResultSet

ResultSet supportsResultSetType

boolean supportsResultSetType(int type)
throws ResourceException

supportsResultSetType

Indicates whether or not a resource adapter supports a type of
(resultSet)

**Parameters:**
- `type` - type of the ResultSet i.e. ResultSet.TYPE_XXX

**Returns:**
- true if ResultSet type supported; false otherwise

**Throws:**
- `ResourceException` - Failed to get the information

---

public boolean supportsResultTypeConcurrency(int type, int concurrency) throws ResourceException

**ResultSet**

<table>
<thead>
<tr>
<th>type</th>
<th>ResultSet ResultSet.TYPE_XXX</th>
</tr>
</thead>
<tbody>
<tr>
<td>concurrency</td>
<td>java.sql.ResultSet ResultSet</td>
</tr>
<tr>
<td>return</td>
<td>true false</td>
</tr>
<tr>
<td>Throws</td>
<td><code>ResourceException</code></td>
</tr>
</tbody>
</table>

**supportsResultTypeConcurrency**

boolean **supportsResultTypeConcurrency**(int type, int concurrency) throws ResourceException

Indicates whether or not a resource adapter supports the concurrency type in combination with the given ResultSet type/

**Parameters:**
- `type` - type of the ResultSet i.e. ResultSet.TYPE_XXX
- `concurrency` - ResultSet concurrency type defined in java.sql.ResultSet

**Returns:**
- true if the specified combination supported; false otherwise

**Throws:**
- `ResourceException` - Failed to get the information
public boolean othersUpdatesAreVisible(int type) throws ResourceException

    type                ResultSet  ResultSet.TYPE_XXX
    return              ResultSet  true  false
    Throws               ResourceException:

othersUpdatesAreVisible

boolean othersUpdatesAreVisible(int type)
    throws ResourceException

    Indicates whether updates made by others are visible.

Parameters:
    type - type of the ResultSet i.e. ResultSet.TYPE_XXX

Returns:
    true if updates by others are visible for the ResultSet type; false otherwise

Throws:
    ResourceException - Failed to get the information

public boolean othersDeletesAreVisible(int type) throws ResourceException

    type                ResultSet  ResultSet.TYPE_XXX
    return              ResultSet  true  false
    Throws               ResourceException:

othersDeletesAreVisible

boolean othersDeletesAreVisible(int type)
    throws ResourceException

    Indicates whether deletes made by others are visible.
Parameters:
  type - type of the ResultSet i.e. ResultSet.TYPE_XXX

Returns:
  true if deletes by others are visible for the ResultSet type; false otherwise

Throws:
  ResourceException - Failed to get the information

public boolean othersInsertsAreVisible(int type) throws ResourceException

  type  ResultSet  ResultSet.TYPE_XXX
  return ResultSet  true  false

Throws
  ResourceException:

othersInsertsAreVisible

boolean othersInsertsAreVisible(int type)
  throws ResourceException

Indicates whether inserts made by others are visible.

Parameters:
  type - type of the ResultSet i.e. ResultSet.TYPE_XXX

Returns:
  true if inserts by others are visible for the ResultSet type; false otherwise

Throws:
  ResourceException - Failed to get the information

public boolean ownUpdatesAreVisible(int type) throws ResourceException

OthersInsertsAreVisible
ownUpdatesAreVisible

boolean ownUpdatesAreVisible(int type)
throws ResourceException

Throws:
ResourceException

public boolean ownInsertsAreVisible(int type) throws
ResourceException

ownInsertsAreVisible

boolean ownInsertsAreVisible(int type)
throws ResourceException

Throws:
ResourceException

public boolean ownDeletesAreVisible(int type) throws
ResourceException

ownDeletesAreVisible

boolean ownDeletesAreVisible(int type)
throws ResourceException

Throws:
ResourceException
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.annotation.security Annotation Type RolesAllowed

@Documented
@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})
public @interface RolesAllowed

**Implements:** Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})

RolesAllowed

since 1.0

Specifies the list of roles permitted to access method(s) in an application. The value of the RolesAllowed annotation is a list of security role names. This annotation can be specified on a class or on method(s). Specifying it at a class level means that it applies to all the methods in the class. Specifying it on a method means that it is applicable to that method only. If applied at both the class and methods level, the method value overrides the class value if the two conflict.

Since:

1.0

---

## Required Element Summary

| String[] | value |

---

## Element Detail
abstract public String[] value()
javax.persistence Class RollbackException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.persistence.PersistenceException
                  └ javax.persistence.RollbackException

All Implemented Interfaces:
  Serializable

class RollbackException

  extends PersistenceException

  Extends: Throwable > Exception > RuntimeException > PersistenceException

EntityTransaction.commit() since Java Persistence 1.0

  See also commit()

Thrown by the persistence provider when the EntityTransaction.commit() fails.

Since:
  Java Persistence 1.0

See Also:
  EntityTransaction.commit(), Serialized Form

Constructor Summary

RollbackException()
  Constructs a new RollbackException exception with null as its detail message.
RollbackException(String message)
   Constructs a new RollbackException exception with the specified detail message.

RollbackException(String message, Throwable cause)
   Constructs a new RollbackException exception with the specified detail message and cause.

RollbackException(Throwable cause)
   Constructs a new RollbackException exception with the specified cause.

Method Summary

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public RollbackException()
   null RollbackException

RollbackException

public RollbackException()
   Constructs a new RollbackException exception with null as its detail message.
public RollbackException(String message)
RollbackException
message

RollbackException

public RollbackException(String message)

    Constructs a new RollbackException exception with the specified
detail message.

    Parameters:
    message - the detail message.

public RollbackException(String message, Throwable cause)
RollbackException
message
cause

RollbackException

public RollbackException(String message, Throwable cause)

    Constructs a new RollbackException exception with the specified
detail message and cause.

    Parameters:
    message - the detail message.
    cause - the cause.

public RollbackException(Throwables cause)
RollbackException
public RollbackException(Throwable cause)

Constructs a new RollbackException exception with the specified cause.

Parameters:
cause - the cause.
javax.transaction

Class RollbackException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.transaction.RollbackException

All Implemented Interfaces:
  Serializable

public class RollbackException
  extends Exception

Extends: Throwable > Exception

RollbackException
TransactionManager

RollbackException exception is thrown when the transaction has been marked for rollback only or the transaction has been rolled back instead of committed. This is a local exception thrown by methods in the UserTransaction, Transaction, and TransactionManager interfaces.

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RollbackException()</td>
<td></td>
</tr>
<tr>
<td>RollbackException(String msg)</td>
<td></td>
</tr>
</tbody>
</table>

Method Summary
Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public RollbackException()

RollbackException

public RollbackException()

public RollbackException(String msg)

RollbackException

public RollbackException(String msg)
PS:
javax.annotation.security  Annotation Type RunAs

@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface RunAs

**Implements**: Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)

J2EE /

class RunAs

Since: 1.0

Defines the identity of the application during execution in a J2EE container. This allows developers to execute under a particular role. The role must map to the user / group information in the containers security realm. It’s value is the name of a security role.

Since: 1.0

### Required Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>value</strong></td>
<td>String</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String value()
value

public abstract String value

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.soap  

**Class SAAJMetaFactory**

```java
java.lang.Object
  ^javax.xml.soap.SAAJMetaFactory
```

**public abstract class SAAJMetaFactory**

**extends Object**

SAAJ API SAAJ 1.3  
SAAJ 1.2  
SAAJMetaFactory

**SAAJMetaFactory  public**

```
    since SAAJ 1.3
```

The access point for the implementation classes of the factories defined in the SAAJ API. All of the `newInstance` methods defined on factories in SAAJ 1.3 defer to instances of this class to do the actual object creation. The implementations of `newInstance()` methods (in SOAPFactory and MessageFactory) that existed in SAAJ 1.2 have been updated to also delegate to the SAAJMetaFactory when the SAAJ 1.2 defined lookup fails to locate the Factory implementation class name.

SAAJMetaFactory is a service provider interface. There are no public methods on this class.

**Since:**

SAAJ 1.3

**Author:**

SAAJ RI Development Team

---

**Constructor Summary**
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected abstract MessageFactory newMessageFactory(String protocol)</td>
<td>Creates a MessageFactory object for the given String protocol.</td>
</tr>
<tr>
<td>protected abstract SOAPFactory newSOAPFactory(String protocol)</td>
<td>Creates a SOAPFactory object for the given String protocol.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

## Constructor Detail

protected SAAJMetaFactory()

SAAJMetaFactory

protected SAAJMetaFactory()

## Method Detail

abstract protected MessageFactory newMessageFactory(String protocol) throws SOAPException

String protocol  
MessageFactory protocol

Throws SOAPException: MessageFactory
newMessageFactory

protected abstract MessageFactory newMessageFactory(String protocol) throws SOAPException

Creates a MessageFactory object for the given String protocol.

Parameters:
  protocol - a String indicating the protocol

Throws:
  SOAPException - if there is an error in creating the MessageFactory

See Also:
  SOAPConstants.SOAP_1_1_PROTOCOL,
  SOAPConstants.SOAP_1_2_PROTOCOL,
  SOAPConstants.DYNAMIC_SOAP_PROTOCOL

abstract protected SOAPFactory newSOAPFactory(String protocol) throws SOAPException

String SOAPFactory

Throws SOAPException: SOAPFactory

See SOAP_1_1_PROTOCOL, SOAP_1_2_PROTOCOL,
also DYNAMIC_SOAP_PROTOCOL

newSOAPFactory

protected abstract SOAPFactory newSOAPFactory(String protocol) throws SOAPException

Creates a SOAPFactory object for the given String protocol.

Parameters:
protocol - a String indicating the protocol

Throws:

SOAPException - if there is an error in creating the SOAPFactory

See Also:

SOAPConstants.SOAP_1_1_PROTOCOL,
SOAPConstants.SOAP_1_2_PROTOCOL,
SOAPConstants.DYNAMIC_SOAP_PROTOCOL

PS:
javax.xml.soap Class SAAJResult

java.lang.Object
    ▼ javax.xml.transform.dom.DOMResult
       ▼ javax.xml.soap.SAAJResult

All Implemented Interfaces:
   Result

public class SAAJResult
  extends DOMResult

  Extends: javax.xml.transform.dom.DOMResult

  SAAJ JAXP JAXB

  getNode() since SAAJ 1.3

Acts as a holder for the results of a JAXP transformation or a JAXB
marshalling, in the form of a SAAJ tree. These results should be
accessed by using the getResult() method. The DOMResult.getNode() method
should be avoided in almost all cases.

Since:
   SAAJ 1.3

Author:
   XWS-Security Development Team

---

Field Summary

| Fields inherited from class javax.xml.transform.dom.DOMResult |
| FEATURE |


### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SAAJResult()</strong></td>
<td>Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the default (SOAP 1.1) protocol.</td>
</tr>
<tr>
<td><strong>SAAJResult(SOAPElement rootNode)</strong></td>
<td>Creates a SAAJResult that will write the results as a child node of the SOAPElement specified.</td>
</tr>
<tr>
<td><strong>SAAJResult(SOAPMessage message)</strong></td>
<td>Creates a SAAJResult that will write the results into the SOAPPart of the supplied SOAPMessage.</td>
</tr>
<tr>
<td><strong>SAAJResult(String protocol)</strong></td>
<td>Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the specified protocol.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Node</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>getResult()</strong></td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public SAAJResult() throws SOAPException
```
SAAJResult (SOAP 1.1) SAAJ

<table>
<thead>
<tr>
<th>SAAJResult</th>
<th>SOAPElement</th>
<th>SAAJ API</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPMessage</td>
<td>SOAPPart</td>
<td>getNode()</td>
</tr>
</tbody>
</table>

Throws

since

SAAJ 1.3

SAAJResult

public SAAJResult()
    throws SOAPException

    Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the default (SOAP 1.1) protocol.

This kind of SAAJResult is meant for use in situations where the results will be used as a parameter to a method that takes a parameter whose type, such as SOAPElement, is drawn from the SAAJ API. When used in a transformation, the results are populated into the SOAPPart of a SOAPMessage that is created internally. The SOAPPart returned by DOMResult.getNode() is not guaranteed to be well-formed.

Throws:

    SOAPException - if there is a problem creating a SOAPMessage

Since:

    SAAJ 1.3

public SAAJResult(String protocol) throws SOAPException
    SAAJResult SAAJ DYNAMIC_SOAP_PROTOCOL
UnsupportedException

SAAJResult | SOAPElement | SAAJ API |
------------|-------------|----------|
SOAPMessage | SOAPPart | getNode() |

protocol SAAJ SOAP
SAAJResult

public SAAJResult(String protocol)
throws SOAPException

Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the specified protocol. The DYNAMIC_SOAP_PROTOCOL is ambiguous in this context and will cause this constructor to throw an UnsupportedOperationException.

This kind of SAAJResult is meant for use in situations where the results will be used as a parameter to a method that takes a parameter whose type, such as SOAPElement, is drawn from the SAAJ API. When used in a transformation the results are populated into the SOAPPart of a SOAPMessage that is created internally. The SOAPPart returned by DOMResult.getNode() is not guaranteed to be well-formed.

Parameters:
protocol - the name of the SOAP protocol that the resulting SAAJ tree should support

Throws:
SOAPException - if a SOAPMessage supporting the specified protocol cannot be created

Since:
SAAJ 1.3
SAAJResult

public SAAJResult(SOAPMessage message)

Creates a SAAJResult that will write the results into the SOAPPart of the supplied SOAPMessage. In the normal case these results will be written using DOM APIs and, as a result, the finished SOAPPart will not be guaranteed to be well-formed unless the data used to create it is also well formed. When used in a transformation the validity of the SOAPMessage after the transformation can be guaranteed only by means outside SAAJ specification.

Parameters:
message -- the message whose SOAPPart will be populated as a result of some transformation or marshalling operation

Since:
SAAJ 1.3

public SAAJResult(SOAPElememt rootNode)

Creates a SAAJResult that will write the results as a child node of the SOAPElememt specified. In the normal case these results will be written using DOM APIs and as a result may invalidate the structure of the SAAJ tree. This kind of SAAJResult should only be used when the validity of the incoming data can be guaranteed by means outside of
the SAAJ specification.

**Parameters:**
- `rootNode` - the root to which the results will be appended

**Since:**
SAAJ 1.3

---

### Method Detail

**public Node getResult()**

```java
default Node Tree
since SAAJ 1.3
```

**getResult**

**public Node getResult()**

**Returns:**
the resulting Tree that was created under the specified root Node.

**Since:**
SAAJ 1.3

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Summary</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>FIELD</td>
<td>FIELD</td>
</tr>
<tr>
<td></td>
<td>CONSTR</td>
<td>CONSTR</td>
</tr>
<tr>
<td></td>
<td>METHOD</td>
<td>METHOD</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES
DETAIL: FIELD | CONSTR | METHOD
javax.xml.registry Class SaveException

java.lang.Object
   ▼ java.lang.Throwable
      ▼ java.lang.Exception
         ▼ javax.xml.registry.JAXRException
             ▼ javax.xml.registry.RegistryException
                ▼javax.xml.registry.SaveException

All Implemented Interfaces:
   Serializable, JAXRResponse

public class SaveException
extends RegistryException

Extends: Throwable > Exception > JAXRException > RegistryException

save  RegistryException

A RegistryException that occurs during a save action.

Author:
   Farrukh S. Najmi

See Also:
   Serialized Form

Field Summary

| Fields inherited from class javax.xml.registry.JAXRException |
| cause |

| Fields inherited from interface javax.xml.registry.JAXRResponse |
| STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING |
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SaveException()</strong></td>
<td>-</td>
<td>Constructs a JAXRException object with no reason or embedded Throwable.</td>
</tr>
<tr>
<td><strong>SaveException(String reason)</strong></td>
<td>-</td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><strong>SaveException(String reason, Throwable cause)</strong></td>
<td>-</td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.</td>
</tr>
<tr>
<td><strong>SaveException(Throwable cause)</strong></td>
<td>-</td>
<td>Constructs a JAXRException object initialized with the given Throwable object.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class javax.xml.registry.RegistryException</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>setErrorObjectKey, setErrorObjectKey</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.xml.registry.JAXRException</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>getRequestId, getStatus, initCause, isAvailable</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getLocalizedMessage, getStackTrace, printStackTrace, printStackTrace, setStackTrace, toString</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
public SaveException()
 Throwable JAXRException

SaveException
 public SaveException()

    Constructs a JAXRException object with no reason or embedded Throwable.

public SaveException(String reason)
 JAXRException String reason

SaveException
 public SaveException(String reason)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

    Parameters:
    reason - a description of what caused the exception

public SaveException(String reason, Throwable cause)
 JAXRException String Throwable

Throwables
   reason
   cause JAXRException Throwable
SaveException

public SaveException(String reason, Throwable cause)

Constructs a JAXRException object with the given string as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
reason - a description of what caused the exception
cause - a Throwable object that is to be embedded in this JAXRException object

public SaveException(Throwable cause)

(Throwable cause) Throwable JAXRException
cause Exception Throwable

SaveException

public SaveException(Throwable cause)

Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
cause - the Throwable that caused this Exception

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
public class ScalarDataModel
extends DataModel

**ScalarDataModel** is a convenience implementation of DataModel that wraps an individual Java object.

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ScalarDataModel()</strong></td>
<td>Construct a new ScalarDataModel with no specified wrapped data.</td>
</tr>
<tr>
<td><strong>ScalarDataModel(Object scalar)</strong></td>
<td>Construct a new ScalarDataModel wrapping the specified scalar object.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getRowCount()</strong></td>
<td>If there is wrapped data available, return 1.</td>
</tr>
<tr>
<td><strong>getRowData()</strong></td>
<td>If wrapped data is available, return the wrapped data instance.</td>
</tr>
<tr>
<td><strong>getIndex()</strong></td>
<td></td>
</tr>
</tbody>
</table>
Return the zero-relative index of the currently selected row.

**getWrappedData()**

Return the object representing the data wrapped by this DataModel, if any.

**isRowAvailable()**

Return true if there is wrappedData available, and the current value of rowIndex is zero.

**setRowIndex(int rowIndex)**

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row.

**setWrappedData(Object data)**

Set the object representing the data collection wrapped by this DataModel.

**Methods inherited from class javax.faces.model.DataModel**

- addDataModelListener
- getDataModelListeners
- removeDataModelListener

**Methods inherited from class java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

**public ScalarDataModel()**

ScalarDataModel

**public ScalarDataModel()**
Construct a new ScalarDataModel with no specified wrapped data.

```java
public ScalarDataModel(Object scalar)

ScalarDataModel

scalar
```

ScalarDataModel

```java
public ScalarDataModel(Object scalar)

Construct a new ScalarDataModel wrapping the specified scalar object.

Parameters:
scalar - Scalar to be wrapped (if any)
```

**Method Detail**

```java
public boolean isRowAvailable()

wrappedData  rowIndex  0   true  false

Throws  FacesException:
```

isRowAvailable

```java
public boolean isRowAvailable()

Return true if there is wrappedData available, and the current value of rowIndex is zero. Otherwise, return false.

Specified by:
```
**isRowAvailable** in class **DataModel**

**Throws:**
- **FacesException** - if an error occurs getting the row availability

---

**public int getRowCount()**

wrappedData 1       wrappedData -1

**Throws**
- **FacesException**

**getRowCount**

**public int getRowCount()**

If there is wrappedData available, return 1. If no wrappedData is available, return -1.

**Specified by:**
- **getRowCount** in class **DataModel**

**Throws:**
- **FacesException** - if an error occurs getting the row count

---

**public Object getRowData()**

null

**Throws**
- **FacesException**
- **IllegalArgumentException**

**getRowData**

**public Object getRowData()**

If wrapped data is available, return the wrapped data instance.
Otherwise, return null.

Specified by:

getRowData in class DataModel

Throws:

FacesException - if an error occurs getting the row data
IllegalArgumentException - if now row data is available at the currently specified row index

public int getRowIndex()

Throws FacesException: NullPointerException

getRowIndex

public int getRowIndex()

Description copied from class: DataModel

Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, or no wrappedData is available, return -1.

Specified by:

getRowIndex in class DataModel

Throws:

FacesException - if an error occurs getting the row index

public void setRowIndex(int rowIndex)

Throws FacesException: NullPointerException

Throws IllegalArgumentException: NullPointerException rowIndex -1

setRowIndex
public void setRowIndex(int rowIndex)

**Description copied from class:** [DataModel](#)

Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the `isRowAvailable()` method to detect whether row data will be available for use by the `getRowData()` method.

If there is no `wrappedData` available when this method is called, the specified `rowIndex` is stored (and may be retrieved by a subsequent call to `getRowData()`), but no event is sent. Otherwise, if the currently selected row index is changed by this call, a `DataModelEvent` will be sent to the `rowSelected()` method of all registered `DataModelListener`s.

**Specified by:**

`setRowIndex` in class [DataModel](#)

**Parameters:**

- `rowIndex` - The new zero-relative index (must be non-negative)

**Throws:**

- [FacesException](#) - if an error occurs setting the row index
- [IllegalArgumentException](#) - if `rowIndex` is less than -1

---

public Object getWrappedData()

**getWrappedData**

public Object getWrappedData()

**Description copied from class:** [DataModel](#)

Return the object representing the data wrapped by this [DataModel](#), if any.
public void setWrappedData(Object data)

Description copied from class: DataModel

Set the object representing the data collection wrapped by this DataModel. If the specified data is null, detach this DataModel from any previously wrapped data collection instead.

If data is non-null, the currently selected row index must be set to zero, and a DataModelEvent must be sent to the rowSelected() method of all registered DataModelListener indicating that this row is now selected.

Specified by:
setWrappedData in class DataModel

Parameters:
- data - Data collection to be wrapped, or null to detach from any previous data collection

Throws:
- ClassCastException - if data is not of the appropriate type for this DataModel implementation
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.xml.bind Class SchemaOutputResolver**

**java.lang.Object**

- javax.xml.bind.SchemaOutputResolver

```java
public abstract class SchemaOutputResolver
    extends Object```

JAXB

Controls where a JAXB implementation puts the generates schema files.

An implementation of this abstract class has to be provided by the calling application to generate schemas.

This is a class, not an interface so as to allow future versions to evolve without breaking the compatibility.

**Author:**
Kohsuke Kawaguchi (kohsuke.kawaguchi@sun.com)

---

### Constructor Summary

**SchemaOutputResolver()**

---

### Method Summary

**createOutput(String namespaceUri, String suggestedFileName)**
Decides where the schema file (of the given namespace URI) will be written, and return it as a Result object.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public SchemaOutputResolver()

SchemaOutputResolver

public SchemaOutputResolver()

Method Detail

abstract public javax.xml.transform.Result createOutput(String namespaceUri, String suggestedFileName) throws java.io.IOException

 URI

namespaceUri      URI null
suggestedFileName JAXB "schema1.xsd"

return  <xs:import>
<xs:import>

createOutput
public abstract Result createOutput(String namespaceUri, String suggestedFileName)
throws IOException

Decides where the schema file (of the given namespace URI) will be written, and return it as a Result object.

This method is called only once for any given namespace. IOW, all the components in one namespace is always written into the same schema document.

Parameters:
namespaceUri - The namespace URI that the schema declares. Can be the empty string, but never be null.
suggestedFileName - A JAXB implementation generates an unique file name (like "schema1.xsd") for the convenience of the callee. This name can be used for the file name of the schema, or the callee can just ignore this name and come up with its own name. This is just a hint.

Returns:
a Result object that encapsulates the actual destination of the schema. If the Result object has a system ID, it must be an absolute system ID. Those system IDs are relativized by the caller and used for <xs:import> statements. If the Result object does not have a system ID, a schema for the namespace URI is generated but it won’t be explicitly <xs:import>ed from other schemas. If null is returned, the schema generation for this namespace URI will be skipped.

Throws:
IOException
PS:
| PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTN | METHOD | FRAMES | NO FRAMES | DETAIL: FIELD | CONSTN | METHOD |
public class ScopedAttributeELResolver
extends ELResolver

Defines variable resolution behavior for scoped attributes.

This resolver handles all variable resolutions (where base is null. It searches PageContext.findAttribute() for a matching attribute. If not found, it will return null, or in the case of setValue it will create a new attribute in the page scope with the given name.

Since:
   JSP 2.1

See Also:
   ELResolver

Field Summary

Fields inherited from class javax.el.ELResolver
RESOLVABLE_AT_DESIGN_TIME, TYPE

Constructor Summary

ScopedAttributeELResolver()
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Class&lt;String&gt; Object base)</code></td>
<td>If the base object is <code>null</code>, returns <code>String.class</code>.</td>
</tr>
<tr>
<td><code>getFeatureDescriptors(ELContext context, Object base)</code></td>
<td>If the base object is <code>null</code>, returns an <code>Iterator</code> containing <code>FeatureDescriptor</code> objects with information about each scoped attribute resolved by this resolver.</td>
</tr>
<tr>
<td><code>Class&lt;Object&gt; getType(ELContext context, Object base, Object property)</code></td>
<td>If the base object is <code>null</code>, returns <code>Object.class</code> to indicate that any type is valid to set for a scoped attribute.</td>
</tr>
<tr>
<td><code>Iterator&lt;FeatureDescriptor&gt; getFeatureDescriptors(ELContext context, Object base)</code></td>
<td>If the base object is <code>null</code>, returns an <code>Iterator</code> containing <code>FeatureDescriptor</code> objects with information about each scoped attribute resolved by this resolver.</td>
</tr>
<tr>
<td><code>getValue(ELContext context, Object base, Object property)</code></td>
<td>If the base object is <code>null</code>, searches the page, request, session and application scopes for an attribute with the given name and returns it, or <code>null</code> if no attribute exists with the current name.</td>
</tr>
<tr>
<td><code>boolean isReadOnly(ELContext context, Object base, Object property)</code></td>
<td>If the base object is <code>null</code>, returns <code>false</code> to indicate that scoped attributes are never read-only.</td>
</tr>
<tr>
<td><code>void setValue(ELContext context, Object base, Object property, Object val)</code></td>
<td>If the base object is <code>null</code>, sets an existing scoped attribute to the new value, or creates a new scoped attribute if one does not exist by this name.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`
### Constructor Detail

**ScopedAttributeELResolver**

```java
public ScopedAttributeELResolver()
```

### Method Detail

**getValue**

```java
public Object getValue(ELContext context, Object base, Object property)
```

If the base object is `null`, searches the page, request, session and application scopes for an attribute with the given name and returns it, or `null` if no attribute exists with the current name.

The `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver before returning if base is `null`. If this property is not `true` after this method is called, the caller should ignore the return value.

**Specified by:**

`getValue` in class `ELResolver`

**Parameters:**

- `context` - The context of this evaluation.
- `base` - Only `null` is handled by this resolver. Other values will result in an immediate return.
- `property` - The name of the scoped attribute to resolve.

**Returns:**

- If the `propertyResolved` property of `ELContext` was set to `true`, then the scoped attribute; otherwise undefined.

**Throws:**

- `NullPointerException` - if context is `null`
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.
**getType**

```java
public Class<Object> getType(ELContext context, Object base, Object property)
```

If the base object is `null`, returns `Object.class` to indicate that any type is valid to set for a scoped attribute.

The `propertyResolved` property of the `ELContext` object must be set to `true` by this resolver before returning if base is `null`. If this property is not `true` after this method is called, the caller should ignore the return value.

Specified by:
`getType` in class `ELResolver`

Parameters:
- `context` - The context of this evaluation.
- `base` - Only `null` is handled by this resolver. Other values will result in an immediate return.
- `property` - The name of the scoped attribute to resolve.

Returns:
If the `propertyResolved` property of `ELContext` was set to `true`, then `Object.class`; otherwise undefined.

Throws:
- `NullPointerException` - if `context` is `null`
- `ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

**setValue**

```java
public void setValue(ELContext context, Object base, Object property, Object val)
```

If the base object is `null`, sets an existing scoped attribute to the
new value, or creates a new scoped attribute if one does not exist by this name.

If the provided attribute name matches the key of an attribute in page scope, request scope, session scope, or application scope, the corresponding attribute value will be replaced by the provided value. Otherwise, a new page scope attribute will be created with the given name and value.

The propertyResolved property of the ELContext object must be set to true by this resolver before returning if base is null. If this property is not true after this method is called, the caller should ignore the return value.

Specified by:
setValue in class ELResolver

Parameters:
context - The context of this evaluation.
base - Only null is handled by this resolver. Other values will result in an immediate return.
property - The name of the scoped attribute to set.
val - The value for the scoped attribute.

Throws:
NullPointerException - if context is null.
ELException - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

isReadOnly

public boolean isReadOnly(ELContext context, Object base, Object property)

If the base object is null, returns false to indicate that scoped attributes are never read-only.

The propertyResolved property of the ELContext object must be set to true by this resolver before returning if base is null. If this property
is not true after this method is called, the caller should ignore the return value.

**Specified by:**

`isReadOnly` in class `ELResolver`  

**Parameters:**

- `context` - The context of this evaluation.
- `base` - Only null is handled by this resolver. Other values will result in an immediate return.
- `property` - The name of the scoped attribute.

**Returns:**

If the `propertyResolved` property of `ELContext` was set to true, then false; otherwise undefined.

**Throws:**

`NullPointerException` - if `context` is null.
`ELException` - if an exception was thrown while performing the property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

---

### getFeatureDescriptors

```java
public Iterator<FeatureDescriptor> getFeatureDescriptors(ELContext context, Object base)
```

If the base object is `null`, returns an `Iterator` containing `FeatureDescriptor` objects with information about each scoped attribute resolved by this resolver. Otherwise, returns `null`.

The `Iterator` returned must contain one instance of `FeatureDescriptor` for each scoped attribute found in any scope. Each info object contains information about a single scoped attribute, and is initialized as follows:

- `displayName` - The name of the scoped attribute.
- `name` - Same as `displayName` property.
- `shortDescription` - A suitable description for the scoped attribute. Should include the attribute's current scope (page, request, session, application). Will vary by implementation.
In addition, the following named attributes must be set in the returned FeatureDescriptors:

- **ELResolver.TYPE** - The current runtime type of the scoped attribute.
- **ELResolver.RESOLVABLE_AT_DESIGN_TIME** - true.

**Specified by:**

*getFeatureDescriptors* in class *ELResolver*

**Parameters:**

- **context** - The context of this evaluation.
- **base** - Only null is handled by this resolver. Other values will result in a null return value.

**Returns:**

An Iterator containing one FeatureDescriptor object for each scoped attribute, or null if base is not null.

**See Also:**

*FeatureDescriptor*

---

**getCommonPropertyType**

```java
public Class<String> getCommonPropertyType(ELContext context, Object base)
```

If the base object is null, returns String.class. Otherwise, returns null.

**Specified by:**

*getCommonPropertyType* in class *ELResolver*

**Parameters:**

- **context** - The context of this evaluation.
- **base** - Only null is handled by this resolver. Other values will result in a null return value.

**Returns:**

null if base is not null; otherwise String.class.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.mail.search  Class SearchException

java.lang.Object  
  java.lang.Throwable  
    java.lang.Exception  
      javax.mail.MessagingException  
        javax.mail.search.SearchException

All Implemented Interfaces:
  Serializable

public class SearchException

extends MessagingException

Extends: Throwable > Exception > MessagingException

Search

The exception thrown when a Search expression could not be handled.

Author:
  John Mani

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchException()</td>
<td>Constructs a SearchException with no detail message.</td>
</tr>
<tr>
<td>SearchException(String s)</td>
<td>Constructs a SearchException with the specified detail message.</td>
</tr>
</tbody>
</table>

Method Summary
Constructor Detail

public SearchException()
SearchException

SearchException

public SearchException()

    Constructs a SearchException with no detail message.

public SearchException(String s)
SearchException

String s

SearchException

public SearchException(String s)
Constructs a SearchException with the specified detail message.

**Parameters:**

s - the detail message
javax.mail.search  Class SearchTerm

java.lang.Object  
   javax.mail.search.SearchTerm

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   AddressTerm, AndTerm, ComparisonTerm, FlagTerm, NotTerm, OrTerm, StringTerm

public abstract class SearchTerm
extends Object
implements Serializable

Implements: java.io.Serializable
Extended by: AddressTerm, AndTerm, ComparisonTerm, FlagTerm, NotTerm, OrTerm, StringTerm

JavaMail API

      JDK 1.1    Message.RecipientType
Message.RecipientType  readReplace  JDK 1.2

Search criteria are expressed as a tree of search-terms, forming a parse-tree for the search expression.

Search-terms are represented by this class. This is an abstract class; subclasses implement specific match methods.
Search terms are serializable, which allows storing a search term between sessions. **Warning:** Serialized objects of this class may not be compatible with future JavaMail API releases. The current serialization support is appropriate for short term storage.

**Warning:** Search terms that include references to objects of type `Message.RecipientType` will not be deserialized correctly on JDK 1.1 systems. While these objects will be deserialized without throwing any exceptions, the resulting objects violate the `type-safe enum` contract of the `Message.RecipientType` class. Proper deserialization of these objects depends on support for the `readReplace` method, added in JDK 1.2.

**Author:**
Bill Shannon, John Mani

**See Also:**
[Serialized Form](#)

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SearchTerm()</code></td>
<td></td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract boolean match(Message msg)</code></td>
<td>This method applies a specific match criterion to the given message and returns the result.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

---

### Constructor Detail

public `SearchTerm()`
abstract public boolean match(Message msg)

method applies a specific match criterion to the given message and returns the result.

Parameters:
msg - The match criterion is applied on this message

Returns:
true, it the match succeeds, false if the match fails
javax.persistence

Annotation Type SecondaryTable

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface SecondaryTable

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

SecondaryTable

```java
@Target
@Retention
public @interface SecondaryTable
```

1

```java
@Entity
@Table(name="CUSTOMER")
@SecondaryTable(name="CUST_DETAIL",
    pkJoinColumns=@PrimaryKeyJoinColumn(name="CUST_ID"))
public class Customer { ... }
```

2

```java
@Entity
@Table(name="CUSTOMER")
@SecondaryTable(name="CUST_DETAIL",
    pkJoinColumns={
        @PrimaryKeyJoinColumn(name="CUST_ID"),
        @PrimaryKeyJoinColumn(name="CUST_TYPE")})
public class Customer { ... }
```

since Java Persistence 1.0

This annotation is used to specify a secondary table for the annotated entity class. Specifying one or more secondary tables indicates that the data for the entity class is stored across multiple tables.

If no `SecondaryTable` annotation is specified, it is assumed that all
persistent fields or properties of the entity are mapped to the primary table. If no primary key join columns are specified, the join columns are assumed to reference the primary key columns of the primary table, and have the same names and types as the referenced primary key columns of the primary table.

Example 1: Single secondary table with a single primary key column.

```java
@Entity
@Table(name="CUSTOMER")
@SecondaryTable(name="CUST_DETAIL",
    pkJoinColumns=@PrimaryKeyJoinColumn(name="CUST_ID"))
public class Customer {
    ...
}
```

Example 2: Single secondary table with multiple primary key columns.

```java
@Entity
@Table(name="CUSTOMER")
@SecondaryTable(name="CUST_DETAIL",
    pkJoinColumns={
        @PrimaryKeyJoinColumn(name="CUST_ID"),
        @PrimaryKeyJoinColumn(name="CUST_TYPE"))
    public class Customer {
        ...
    }
```

Since:
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required) The name of the table.</td>
<td></td>
</tr>
</tbody>
</table>

### Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) The catalog of the table.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PrimaryKeyJoinColumn[]</th>
<th>pkJoinColumns</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Optional) The columns that are used to join with the primary table.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>schema</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>(Optional) The schema of the table.</td>
</tr>
<tr>
<td>--------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>UniqueConstraint[]</td>
<td>(Optional) Unique constraints that are to be placed on the table.</td>
</tr>
</tbody>
</table>

**Element Detail**

**abstract public String name()**

**name**

```java
public abstract String name
```

(Required) The name of the table.

**abstract public String catalog()**

**catalog**

```java
public abstract String catalog
```

(Optional) The catalog of the table.

Defaults to the default catalog.

**Default:**

"""
abstract public String schema()

schema

public abstract String schema

(Optional) The schema of the table.

Defaults to the default schema for user.

Default: ""

abstract public PrimaryJoinColumn[] pkJoinColumns()

pkJoinColumns

public abstract PrimaryJoinColumn[] pkJoinColumns

(Optional) The columns that are used to join with the primary table.

Defaults to the column(s) of the same name(s) as the primary key column(s) in the primary table

Default:

{}
abstract public UniqueConstraint[] uniqueConstraints()

uniqueConstraints

public abstract UniqueConstraint[] uniqueConstraints

(Optional) Unique constraints that are to be placed on the table. These are typically only used if table generation is in effect. These constraints apply in addition to any constraints specified by the Column andJoinColumn annotations and constraints entailed by primary key mappings.

Defaults to no additional constraints.

Default:
{}

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
This annotation is used to specify multiple secondary tables for an entity.

Example 1: Multiple secondary tables assuming primary key column

    @Entity
@Table(name="EMPLOYEE")
@SecondaryTables(
    @SecondaryTable(name="EMP_DETAIL"),
    @SecondaryTable(name="EMP_HIST")
)
public class Employee { ... }

Example 2: Multiple secondary tables with differently named primary key columns.

@Entity
@Table(name="EMPLOYEE")
@SecondaryTables(
    @SecondaryTable(name="EMP_DETAIL",
        pkJoinColumns=@PrimaryKeyJoinColumn(name="EMPL_ID")),
    @SecondaryTable(name="EMP_HIST",
        pkJoinColumns=@PrimaryKeyJoinColumn(name="EMPLOYEE_ID"))
)
public class Employee { ... }

Since:
Java Persistence 1.0

---

<table>
<thead>
<tr>
<th>Required Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SecondaryTable[]</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Element Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract public <strong>SecondaryTable[]</strong> <strong>value()</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>public abstract <strong>SecondaryTable[]</strong> <strong>value</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi Class SecurityException

java.lang.Object
  ↓ java.lang.Throwable
    ↓ java.lang.Exception
      ↓ javax.resource.ResourceException
        ↓ javax.resource.spi.SecurityException

All Implemented Interfaces:
  Serializable

public class SecurityException

extends ResourceException

Extends: Throwable > Exception > ResourceException

SecurityException

  • Subject
  • EIS
  • EIS
  • EIS EIS
  • EIS

version 1.0

A SecurityException indicates error conditions related to the security contract between an application server and resource adapter. The common error conditions represented by this exception are:

  • Invalid security information (represented as a Subject instance) passed across the security contract - for example, credentials have expired or have invalid format.
  • Lack of support for a specific security mechanism in an EIS or resource adapter.
  • Failure to create a connection to an EIS because of failed
authentication or authorization.

- Failure to authenticate a resource principal to an EIS instance or failure to establish a secure association with an underlying EIS instance.
- Access control exception to indicate that a requested access to an EIS resource or a request to create a new connection is denied.

Version:
1.0

Author:
Rahul Sharma, Ram Jeyaraman

See Also:
Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SecurityException()</code></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><code>SecurityException(String message)</code></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><code>SecurityException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and an error code.</td>
</tr>
<tr>
<td><code>SecurityException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>SecurityException(Throwables cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

## Method Summary

Methods inherited from class javax.resource.ResourceException:

- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`
### Constructor Detail

#### public SecurityException()

null

SecurityException

#### public SecurityException()

Constructs a new instance with null as its detail message.

#### public SecurityException(String message)

*message*

SecurityException

#### public SecurityException(String message)

Constructs a new instance with the specified detail message.

**Parameters:**

message - the detail message.
public SecurityException(Throwable cause)
    cause  throwable

SecurityException

public SecurityException(Throwable cause)
    Constructs a new throwable with the specified cause.
    Parameters:
    cause - a chained exception of type Throwable.

public SecurityException(String message, Throwable cause)
    cause  throwable

SecurityException

public SecurityException(String message, Throwable cause)
    Constructs a new throwable with the specified detail message and cause.
    Parameters:
    message - the detail message.
    cause - a chained exception of type Throwable.

public SecurityException(String message, String errorCode)
throwable

message

errorCode

SecurityException

public SecurityException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:

message - a description of the exception.
errorCode - a string specifying the vendor specific error code.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.model *Class SelectItem*

*java.lang.Object*

    javax.faces.model.SelectItem

All Implemented Interfaces:
    Serializable

Direct Known Subclasses:
    SelectItemGroup

public class **SelectItem**

extends **Object**

implements **Serializable**

**Implements:** java.io.Serializable

**Extended by:** SelectItemGroup

**SelectItem** represents a single *item* in the list of supported *items* associated with a **UISelectMany** or **UISelectOne** component.

**See Also:**
    Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SelectItem()</strong></td>
<td>Construct a SelectItem with no initialized property values.</td>
</tr>
<tr>
<td><strong>SelectItem(Object value)</strong></td>
<td>Construct a SelectItem with the specified value.</td>
</tr>
</tbody>
</table>
**SelectItem** *(Object value, String label)*  
Construct a SelectItem with the specified value and label.

**SelectItem** *(Object value, String label, String description)*  
Construct a SelectItem instance with the specified value, label and description.

**SelectItem** *(Object value, String label, String description, boolean disabled)*  
Construct a SelectItem instance with the specified property values.

**SelectItem** *(Object value, String label, String description, boolean disabled, boolean escape)*  
Construct a SelectItem instance with the specified property values.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| **String** | **getDescription()**  
Return a description of this item, for use in development tools. |
| **String** | **getLabel()**  
Return the label of this item, to be rendered visibly for the user. |
| **Object** | **getValue()**  
Return the value of this item, to be delivered to the model if this item is selected by the user. |
| **boolean** | **isDisabled()**  
Return the disabled flag for this item, which should modify the rendered output to make this item unavailable for selection by the user if set to true. |
| **boolean** | **isEscape()**  
Getter for property escape. |
| **void** | **setDescription(String description)**  
Set the description of this item, for use in development tools. |
| **void** | **setDisabled(boolean disabled)**  
Set the disabled flag for this item, which should modify the... |
rendered output to make this item unavailable for selection by the user if set to true.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void setEscape(boolean escape)</td>
<td>Setter for property escape.</td>
</tr>
<tr>
<td>void setLabel(String label)</td>
<td>Set the label of this item, to be rendered visibly for the user.</td>
</tr>
<tr>
<td>void setValue(Object value)</td>
<td>Set the value of this item, to be delivered to the model if this item is selected by this user.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait

**Constructor Detail**

**public SelectItem()**

Construct a SelectItem with no initialized property values.

**SelectItem**

**public SelectItem()**

Construct a SelectItem with no initialized property values.

**public SelectItem(Object value)**

SelectItem.label String description
null disabled false escape true
**SelectItem**

public `SelectItem(Object value)`

Construct a `SelectItem` with the specified value. The `label` property will be set to the value (converted to a String, if necessary), the `description` property will be set to null, the `disabled` property will be set to false, and the `escape` property will be set to true.

**Parameters:**

- `value` - Value to be delivered to the model if this item is selected by the user

---

public `SelectItem(Object value, String label)`

Construct a `SelectItem` with the specified value and label. The `description` property will be set to null, the `disabled` property will be set to false, and the `escape` property will be set to true.

**Parameters:**

- `value` - Value to be delivered to the model if this item is selected by the user
public SelectItem(Object value, String label, String description)

SelectItem disabled false escape true

value
label
description

SelectItem

public SelectItem(Object value, String label, String description)

Construct a SelectItem instance with the specified value, label and description. This disabled property will be set to false, and the escape property will be set to true.

Parameters:
value - Value to be delivered to the model if this item is selected by the user
label - Label to be rendered for this item in the response
description - Description of this item, for use in tools

public SelectItem(Object value, String label, String description, boolean disabled)

SelectItem escape true

value
label
description
SelectItem

public SelectItem(Object value, String label, String description, boolean disabled)

Construct a SelectItem instance with the specified property values. The escape property will be set to true.

Parameters:

value - Value to be delivered to the model if this item is selected by the user
label - Label to be rendered for this item in the response
description - Description of this item, for use in tools
disabled - Flag indicating that this option is disabled

class SelectItem

public SelectItem(Object value, String label, String description, boolean disabled, boolean escape)

SelectItem

 value
 label
 description
 disabled
 escape
 since 1.2

SelectItem

public SelectItem(Object value, String label, String description,
boolean disabled, boolean escape)

Construct a SelectItem instance with the specified property values.

**Parameters:**
- value - Value to be delivered to the model if this item is selected by the user
- label - Label to be rendered for this item in the response
- description - Description of this item, for use in tools
- disabled - Flag indicating that this option is disabled
- escape - Flag indicating that the text of this option should be escaped when rendered.

**Since:**
1.2

### Method Detail

**public String getDescription()**

**getDescription**

**public String getDescription()**

Return a description of this item, for use in development tools.

---

**public void setDescription(String description)**

*description*
**setDescription**

```java
public void setDescription(String description)
```

Set the description of this item, for use in development tools.

**Parameters:**
- `description` - The new description

**isDisabled**

```java
public boolean isDisabled()
```

true

**setDisabled**

```java
public void setDisabled(boolean disabled)
```

true

**disabled**

Set the disabled flag for this item, which should modify the rendered output to make this item unavailable for selection by the user if set to true.
true.

Parameters:
  disabled - The new disabled flag

public String getLabel()

getLabel

public String getLabel()

  Return the label of this item, to be rendered visibly for the user.

public void setLabel(String label)

setLabel

public void setLabel(String label)

  Set the label of this item, to be rendered visibly for the user.

  Parameters:
    label - The new label

public Object getValue()
**getValue**

public Object getValue()

    Return the value of this item, to be delivered to the model if this item is selected by the user.

---

**public void setValue(Object value)**

    value

**setValue**

public void setValue(Object value)

    Set the value of this item, to be delivered to the model if this item is selected by this user.

**Parameters:**

    value - The new value

---

**public boolean isEscape()**

    return escape

**isEscape**

public boolean isEscape()
Getter for property escape.

>Returns:

Value of property escape.

```
public void setEscape(boolean escape)
```

setEscape

public void setEscape(boolean escape)

Setter for property escape.

**Parameters:**

escape - New value of property escape.
javax.faces.model Class SelectItemGroup

java.lang.Object
   ↓ javax.faces.model.SelectItem
      ↓ javax.faces.model.SelectItemGroup

All Implemented Interfaces:
   Serializable

public class SelectItemGroup
extends SelectItem

Extends: SelectItem

SelectItemGroup SelectItem "mut" UISelectMany
UISelectOne Renderer value label

SelectItemGroup selectItems Renderer
<select> HTML Renderer <optgroup> HTML 4.01

SelectItemGroup is a subclass of SelectItem that identifies a set of options that will be made available as a subordinate "submenu" or "options list", depending upon the requirements of the UISelectMany or UISelectOne renderer that is actually used. In general, the value property of this instance will be ignored, and the label property of this instance will be used to label the submenu.

Although it is feasible to incorporate SelectItemGroup instances in the selectItems property of this instance (thereby creating a data structure suitable for cascading submenus), some renderers may place restrictions on the level of nesting they support. For example, HTML based renderers that create an <select> element will typically render this instance as an <optgroup> element, but the HTML 4.01 Specification disallows nested option groups.
Constructor Summary

**SelectItemGroup()**
Construct a SelectItemGroup with no initialized property values.

**SelectItemGroup(String label)**
Construct a SelectItemGroup with the specified label and no associated selectItems.

**SelectItemGroup(String label, String description, boolean disabled, SelectItem[] selectItems)**
Construct a SelectItemGroup with the specified properties.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getSelectItems()</strong></td>
<td>Return the set of subordinate SelectItem objects for this group.</td>
</tr>
<tr>
<td><strong>setSelectItems(SelectItem[] selectItems)</strong></td>
<td>Set the set of subordinate SelectItem objects for this group.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.model.SelectItem
getDescription, getLabel, getValue, isDisabled, isEscape, setDescription, setDisabled, setEscape, setLabel, setValue

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

```java
public SelectItemGroup()
```
public SelectItemGroup()

Construct a SelectItemGroup with no initialized property values.

public SelectItemGroup(String label)

selectItem SelectItemGroup value 0
String description null disabled false

label

Throws NullPointerException: label false

public SelectItemGroup(String label)

Construct a SelectItemGroup with the specified label and no associated selectItems. The value property will be set to a zero-length String, the description property will be set to null, and the disabled property will be set to false.

Parameters:
label - Label to be rendered for this group in the response

Throws:
NullPointerException - if label is false

public SelectItemGroup(String label, String description, boolean disabled, SelectItem[] selectItems)
Construct a `SelectItemGroup` with the specified properties. The value property will be set to a zero-length String.

**Parameters:**
- `label` - Label to be rendered for this group in the response
- `description` - Description of this group, for use in tools
- `disabled` - Flag indicating that this group is disabled
- `selectItems` - Array of `SelectItem` describing the items available in this group

**Throws:**
- `NullPointerException` - if `label` or `selectItems` is `false`
public SelectItem[] getSelectItems()

Return the set of subordinate SelectItem for this group.

public void setSelectItems(SelectItem[] selectItems)

Throws NullPointerException: selectItems null

setSelectItems

public void setSelectItems(SelectItem[] selectItems)

Set the set of subordinate SelectItem for this group.

Parameters:
selectItems - The new set of subordinate items

Throws:
NullPointerException - if selectItems is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail  **Class SendFailedException**

**java.lang.Object**
  ▼ **java.lang.Throwable**
    ▼ **java.lang.Exception**
      ▼ **javax.mail.MessagingException**
        ▼ **javax.mail.SendFailedException**

**All Implemented Interfaces:**
  **Serializable**

---

**public class SendFailedException**

extends **MessagingException**

**Extends:** Throwable > Exception > **MessagingException**

---

**See also**  **send, sendMessage, javax.mail.event.TransportEvent**

This exception is thrown when the message cannot be sent.

The exception includes those addresses to which the message could not be sent as well as the valid addresses to which the message was sent and valid addresses to which the message was not sent.

**Author:**
  John Mani, Max Spivak

**See Also:**
  **Transport.send(javax.mail.Message),**
  **Transport.sendMessage(javax.mail.Message, javax.mail.Address[]),**
  **TransportEvent, Serialized Form**
### Field Summary

<table>
<thead>
<tr>
<th>Protected Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>invalid</code> Address[]</td>
<td></td>
</tr>
<tr>
<td><code>validSent</code> Address[]</td>
<td></td>
</tr>
<tr>
<td><code>validUnsent</code> Address[]</td>
<td></td>
</tr>
</tbody>
</table>

### Constructor Summary

**SendFailedException()**

Constructs a SendFailedException with no detail message.

**SendFailedException(String s)**

Constructs a SendFailedException with the specified detail message.

**SendFailedException(String s, Exception e)**

Constructs a SendFailedException with the specified Exception and detail message.

**SendFailedException(String msg, Exception ex, Address[] validSent, Address[] validUnsent, Address[] invalid)**

Constructs a SendFailedException with the specified string and the specified address objects.

### Method Summary

<table>
<thead>
<tr>
<th>Address[] Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getInvalidAddresses()</code></td>
<td>Return the addresses to which this message could not be sent.</td>
</tr>
<tr>
<td><code>getValidSentAddresses()</code></td>
<td>Return the addresses to which this message was sent successfully.</td>
</tr>
<tr>
<td><code>getValidUnsentAddresses()</code></td>
<td>Return the addresses that are valid but to which this message was not sent.</td>
</tr>
</tbody>
</table>
### Methods inherited from class `javax.mail.MessagingException`

- `getCause`, `getNextException`, `setNextException`, `toString`

### Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Field Detail

#### invalid

protected transient `Address[]` **invalid**

---

#### validSent

protected transient `Address[]` **validSent**

---

#### validUnsent

protected transient `Address[]` **validUnsent**

---

### Constructor Detail
SendFailedException

public SendFailedException()

Constructs a SendFailedException with no detail message.

public SendFailedException(String s)

Constructs a SendFailedException with the specified detail message.

Parameters:

s - the detail message

public SendFailedException(String s, Exception e)

See also
getNextException, setNextException
Constructs a SendFailedException with the specified string and the specified address objects.

Parameters:
- msg - the detail message
- ex - the embedded exception

See also:
- MessagingException.getNextException()
- MessagingException.setNextException(java.lang.Exception)
validSent - valid addresses to which message was sent
validUnsent - valid addresses to which message was not sent
invalid - the invalid addresses

See Also:
MessagingException.getNextException(),
MessagingException.setNextException(java.lang.Exception)

### Method Detail

**public Address[] getValidSentAddresses()**

```
return null
```

**getValidSentAddresses**

**public Address[] getValidSentAddresses()**

> Return the addresses to which this message was sent successfully.

**Returns:**
Addresses to which the message was sent successfully or null

**public Address[] getValidUnsentAddresses()**

```
return null
```

**getValidUnsentAddresses**

**public Address[] getValidUnsentAddresses()**

> Return the addresses that are valid but to which this message was not sent.

**Returns:**
Addresses that are valid but to which the message was not sent successfully or null

public Address[] getInvalidAddresses()

    return null

getInvalidAddresses

public Address[] getInvalidAddresses()

    Return the addresses to which this message could not be sent.

    Returns:
    Addresses to which the message sending failed or null;

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature
javax.mail.search  Class SentDateTerm

java.lang.Object  ↓ javax.mail.search.SearchTerm  ↓
  ↓ javax.mail.search.ComparisonTerm  ↓
   ↓ javax.mail.search.DateTerm  ↓
    ↓ javax.mail.search.SentDateTerm

All Implemented Interfaces:
  Serializable

public final class SentDateTerm
extends DateTerm

Extends: SearchTerm > ComparisonTerm > DateTerm

Message SentDate

This class implements comparisons for the Message SentDate.

Author:
  Bill Shannon, John Mani

See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.DateTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>date</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.ComparisonTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison, EQ, GE, GT, LE, LT, NE</td>
</tr>
</tbody>
</table>
Constructor Summary

**SentDateTerm**(int comparison, *Date* date)
  Constructor.

Method Summary

<table>
<thead>
<tr>
<th>boolean equals(<em>Object</em> obj)</th>
</tr>
</thead>
</table>
| Equality comparison.

<table>
<thead>
<tr>
<th>boolean match(<em>Message</em> msg)</th>
</tr>
</thead>
</table>
| The match method.

Methods inherited from class *javax.mail.search.DateTerm*
getComparison, getDate, hashCode, match

Methods inherited from class *java.lang.Object*
class, finalize, getClass, notify, notifyAll, toString, wait, wait

Constructor Detail

```java
public SentDateTerm(int comparison, java.util.Date date)
```

- *comparison*
- *date*

**SentDateTerm**

```java
public SentDateTerm(int comparison, *Date* date)
```

Constructor.

**Parameters:**
comparison - the Comparison type
date - the date to be compared

## Method Detail

### public boolean match(Message msg)

```
msg
return
```

#### match

**public boolean match(Message msg)**

The match method.

**Specified by:**

`match` in class `SearchTerm`

**Parameters:**

- `msg`: the date comparator is applied to this Message's sent date

**Returns:**

true if the comparison succeeds, otherwise false

### public boolean equals(Object obj)

#### equals

**public boolean equals(Object obj)**

Equality comparison.

**Overrides:**

`equals` in class `DateTerm`
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.persistence Annotation Type SequenceGenerator

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface SequenceGenerator

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

GeneratedValue

@SequenceGenerator(name="EMP_SEQ", allocationSize=25)

since Java Persistence 1.0

This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the GeneratedValue annotation. A sequence generator may be specified on the entity class or on the primary key field or property. The scope of the generator name is global to the persistence unit (across all generator types).

Example:

@SequenceGenerator(name="EMP_SEQ", allocationSize=25)

Since: Java Persistence 1.0

### Required Element Summary

<table>
<thead>
<tr>
<th></th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Required</td>
</tr>
</tbody>
</table>
|   | A unique generator name that can be
referenced by one or more classes to be the generator for primary key values.

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>allocationSize</td>
<td>(Optional) The amount to increment by when allocating sequence numbers from the sequence.</td>
</tr>
<tr>
<td>int</td>
<td>initialValue</td>
<td>(Optional) The value from which the sequence object is to start generating.</td>
</tr>
<tr>
<td>String</td>
<td>sequenceName</td>
<td>(Optional) The name of the database sequence object from which to obtain primary key values.</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String name()

**name**

public abstract String name

(Required) A unique generator name that can be referenced by one or more classes to be the generator for primary key values.

abstract public String sequenceName()
sequenceName

public abstract String sequenceName

(Optional) The name of the database sequence object from which to obtain primary key values.

Defaults to a provider-chosen value.

Default: ""

abstract public int initialValue()

initialValue

public abstract int initialValue

(Optional) The value from which the sequence object is to start generating.

Default: 1

abstract public int allocationSize()

allocationSize

public abstract int allocationSize

(Optional) The amount to increment by when allocating sequence numbers from the sequence.
Default:
50

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The `javax.xml.rpc.encoding.SerializationContext` interface is implemented by the JAX-RPC runtime system in an XML processing mechanism specific manner. A serializer uses the `SerializationContext` interface during the serialization to get the context information related to the XML processing mechanism and to manage information specific to serialization.

**Version:**
1.0

**Author:**
Rahul Sharma

**See Also:**
`Serializer`
PS:
javax.xml.rpc.encoding Interface Serializer

All Superinterfaces:
Serializable

public interface Serializer
extends Serializable

Implements: java.io.Serializable

javax.xml.rpc.encoding.Serializer Serializer Serializer XML
Java XML

version 1.0

The javax.xml.rpc.encoding.Serializer interface defines the base interface for serializers. A Serializer converts a Java object to an XML representation using a specific XML processing mechanism and based on the specified type mapping and encoding style.

Version:
1.0
Author:
Rahul Sharma

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getMechanismType()</td>
</tr>
<tr>
<td>Gets the type of the XML processing mechanism and representation used by this Serializer.</td>
</tr>
</tbody>
</table>

Method Detail
public String getMechanismType()
Serializer  XML

return  XML

getMechanismType

String getMechanismType()

Gets the type of the XML processing mechanism and representation used by this Serializer.

Returns:
XML processing mechanism type

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.rpc.encoding Interface SerializerFactory

All Superinterfaces:
   Serializable

public interface SerializerFactory
   extends Serializable

Implements: java.io.Serializable

javax.xml.rpc.encoding.SerializerFactory Serializer
SerializerFactory TypeMapping TypeMappingRegistry
   version 1.0
See also javax.xml.rpc.encoding.Serializer

The javax.xml.rpc.encoding.SerializerFactory is a factory of the serializers. A SerializerFactory is registered with a TypeMapping object as part of the TypeMappingRegistry.

Version:
   1.0
Author:
   Rahul Sharma
See Also:
   Serializer

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Serializer getSerializerAs(String mechanismType)</td>
</tr>
<tr>
<td>Returns a Serializer for the specified XML processing mechanism type.</td>
</tr>
<tr>
<td>Iterator getSupportedMechanismTypes()</td>
</tr>
<tr>
<td>Returns a list of all XML processing mechanism types supported by this SerializerFactory.</td>
</tr>
</tbody>
</table>
public Serializer getSerializerAs(String mechanismType)
XML Serializer

getSerializerAs

Returns a Serializer for the specified XML processing mechanism type.

Parameters:
mechanismType - XML processing mechanism type [TBD: definition of valid constants]

Throws:
JAXRPCException - If SerializerFactory does not support the specified XML processing mechanism
IllegalArgumentException - If an invalid mechanism type is specified.

public java.util.Iterator<E> getSupportedMechanismTypes()
SerializerFactory XML

getSupportedMechanismTypes

return XML

Returns a list of all XML processing mechanism types supported by this SerializerFactory.

**Returns:**
List of unique identifiers for the supported XML processing mechanism types

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
**javax.jms Interface ServerSession**

public interface ServerSession

ServerSession JMS

ServerSession

- `getSession` - ServerSession JMS
- `start` - ServerSession JMS run

JMS ConnectionConsumer ServerSession
ConnectionConsumer ServerSessionPool ServerSession
ServerSession JMS JMS ServerSession

ServerSession ServerSession ServerSession start
start ServerSession run Session run
run ServerSession ServerSession ServerSessionPool

JMS API ConnectionConsumer Session
ConnectionConsumer Session JMS

**version** 1.0 - 9 March 1998

See [javax.jms.ServerSessionPool](#),
also [javax.jms.ConnectionConsumer](#)

A `ServerSession` object is an application server object that is used by a server to associate a thread with a JMS session (optional).

A `ServerSession` implements two methods:

- `getSession` - returns the `ServerSession`'s JMS session.
- `start` - starts the execution of the `ServerSession` thread and results in the execution of the JMS session's `run` method.
A `ConnectionConsumer` implemented by a JMS provider uses a `ServerSession` to process one or more messages that have arrived. It does this by getting a `ServerSession` from the `ConnectionConsumer`'s `ServerSessionPool`; getting the `ServerSession`'s JMS session; loading it with the messages; and then starting the `ServerSession`.

In most cases the `ServerSession` will register some object it provides as the `ServerSession`'s thread run object. The `ServerSession`'s start method will call the thread's `start` method, which will start the new thread, and from it, call the `run` method of the `ServerSession`'s run object. This object will do some housekeeping and then call the `Session`'s `run` method. When `run` returns, the `ServerSession`'s run object can return the `ServerSession` to the `ServerSessionPool`, and the cycle starts again.

Note that the JMS API does not architect how the `ConnectionConsumer` loads the `Session` with messages. Since both the `ConnectionConsumer` and `Session` are implemented by the same JMS provider, they can accomplish the load using a private mechanism.

**Version:**
1.0 - 9 March 1998

**Author:**
Mark Hapner, Rich Burridge

**See Also:**
[ServerSessionPool](#), [ConnectionConsumer](#)

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Session getSession()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the <code>ServerSession</code>'s Session.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Void start()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cause the <code>Session</code>'s <code>run</code> method to be called to process messages that were just assigned to it.</td>
</tr>
</tbody>
</table>

---

### Method Detail
public **Session** getSession() throws **JMSException**

ServerSession Session Connection Session Session

ServerSession start

return

Throws **JMSException**: JMS

---

**getSession**

```java
public **Session** getSession()
```

throws **JMSException**

Return the ServerSession's Session. This must be a Session created by the same Connection that will be dispatching messages to it. The provider will assign one or more messages to the Session and then call `start` on the ServerSession.

Returns:
the server session's session

Throws:
**JMSException** - if the JMS provider fails to get the associated session for this ServerSession due to some internal error.

---

public void start() throws **JMSException**

Session run

Throws **JMSException**: JMS

---

**start**

```java
public void start()
```

throws **JMSException**

Cause the Session's `run` method to be called to process messages that were just assigned to it.

Throws:
**JMSException** - if the JMS provider fails to start the server session to process messages due to some internal error.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
### Interface ServerSessionPool

```java
public interface ServerSessionPool
```

<table>
<thead>
<tr>
<th>ServerSessionPool</th>
<th>ConnectionConsumer</th>
<th>ServerSession</th>
</tr>
</thead>
<tbody>
<tr>
<td>getServerSession</td>
<td>JMS API</td>
<td>ServerSession</td>
</tr>
<tr>
<td></td>
<td>ServerSession</td>
<td>ServerSession</td>
</tr>
</tbody>
</table>

**Version:** 1.0 - 9 March 1998

**See also:** [javax.jms.ServerSession](en)

A `ServerSessionPool` object is an object implemented by an application server to provide a pool of `ServerSession` objects for processing the messages of a `ConnectionConsumer` (optional).

Its only method is `getServerSession`. The JMS API does not architect how the pool is implemented. It could be a static pool of `ServerSession` objects, or it could use a sophisticated algorithm to dynamically create `ServerSession` objects as needed.

If the `ServerSessionPool` is out of `ServerSession` objects, the `getServerSession` call may block. If a `ConnectionConsumer` is blocked, it cannot deliver new messages until a `ServerSession` is eventually returned.

**Version:** 1.0 - 9 March 1998

**Author:** Mark Hapner, Rich Burridge

**See Also:**
Method Summary

<table>
<thead>
<tr>
<th>ServerSession</th>
<th>getServerSession()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return a server session from the pool.</td>
</tr>
</tbody>
</table>

Method Detail

```java
class ServerSession {
    public ServerSession getServerSession() throws JMSException {
        return ServerSession.getServerSession();
    }
}
```

`getServerSession`

Returns:

- a server session from the pool

Throws:

- `JMSException` - if an application server fails to return a `ServerSession` out of its server session pool.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail Class Service

java.lang.Object
   javax.mail.Service

Direct Known Subclasses:
   Store, Transport

public abstract class Service
extends Object

Extended by: Store, Transport

store  transport

Session  URLName  Connection  version
         
1.33, 07/05/14

An abstract class that contains the functionality common to messaging services, such as stores and transports.

A messaging service is created from a Session and is named using a URLName. A service must be connected before it can be used. Connection events are sent to reflect its connection status.

Version:
   1.33, 07/05/14

Author:
   Christopher Cotton, Bill Shannon, Kanwar Oberoi

---

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected boolean</td>
<td>debug</td>
<td>Debug flag for this service.</td>
</tr>
</tbody>
</table>
# Constructor Summary

**protected** `Service(Session session, URLName urlname)`

Constructor.

---

# Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addConnectionListener(ConnectionListener l)</code></td>
<td>Add a listener for Connection events on this service.</td>
</tr>
<tr>
<td><code>void close()</code></td>
<td>Close this service and terminate its connection.</td>
</tr>
<tr>
<td><code>void connect()</code></td>
<td>A generic connect method that takes no parameters.</td>
</tr>
<tr>
<td><code>void connect(String host, int port, String user, String password)</code></td>
<td>Similar to connect(host, user, password) except a specific port can be specified.</td>
</tr>
<tr>
<td><code>void connect(String user, String password)</code></td>
<td>Connect to the current host using the specified username and password.</td>
</tr>
<tr>
<td><code>void connect(String host, String user, String password)</code></td>
<td>Connect to the specified address.</td>
</tr>
<tr>
<td><code>protected void finalize()</code></td>
<td>Stop the event dispatcher thread so the queue can be garbage collected.</td>
</tr>
<tr>
<td><code>URLName getURLName()</code></td>
<td>Return a URLName representing this service.</td>
</tr>
<tr>
<td><code>boolean isConnected()</code></td>
<td>Is this service currently connected?</td>
</tr>
<tr>
<td><code>protected void notifyConnectionListeners(int type)</code></td>
<td></td>
</tr>
</tbody>
</table>
void Notify all ConnectionListeners.

protected boolean protocolConnect(String host, int port, String user, String password)
    The service implementation should override this method to perform the actual protocol-specific connection attempt.

protected void queueEvent(MailEvent event, Vector vector)
    Add the event and vector of listeners to the queue to be delivered.

void removeConnectionListener(ConnectionListener l)
    Remove a Connection event listener.

protected void setConnected(boolean connected)
    Set the connection state of this service.

protected void setURLName(URLName url)
    Set the URLName representing this service.

String toString()
    Return getURLName.toString() if this service has a URLName, otherwise it will return the default toString.

---

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

---

Field Detail

protected Session session

The session from which this service was created.
protected URLName url

The URLName of this service.

protected boolean debug

Debug flag for this service. Set from the session's debug flag when this service is created.

## Constructor Detail

protected Service(Session session, URLName urlName)

- `session`: Session object for this service
- `urlName`: URLName object to be used for this service

Constructor.

## Method Detail
public void connect() throws MessagingException

A generic connect method that takes no parameters. Subclasses can implement the appropriate authentication schemes. Subclasses that need additional information might want to use some properties or might get it interactively using a popup window.

If the connection is successful, an "open" ConnectionEvent is delivered to any ConnectionListeners on this service.

Most clients should just call this method to connect to the service.

It is an error to connect to an already connected service.

The implementation provided here simply calls the following connect(String, String, String) method with nulls.

Throws:
public void connect(String host, String user, String password) throws MessagingException

ConnectionListener "open" Connec
Connect to the specified address. This method provides a simple authentication scheme that requires a username and password.

If the connection is successful, an "open" ConnectionEvent is delivered to any ConnectionListeners on this service.

It is an error to connect to an already connected service.

The implementation in the Service class will collect defaults for the host, user, and password from the session, from the URLName for this service, and from the supplied parameters and then call the protocolConnect method. If the protocolConnect method returns false, the user will be prompted for any missing information and the protocolConnect method will be called again. The subclass should override the protocolConnect method. The subclass should also implement the getURLName method, or use the implementation in this class.

On a successful connection, the setURLName method is called with a URLName that includes the information used to make the connection, including the password.

If the username passed in is null, a default value will be chosen as described above. If the password passed in is null and this is the first successful connection to this service, the user name and the password collected from the user will be saved as defaults for subsequent connection attempts to this same service when using other Service object instances (the connection information is typically always saved within a particular Service object instance). The password is saved using the Session method setPasswordAuthentication. If the password passed in is not null, it is not saved, on the assumption that the application is managing passwords explicitly.

**Parameters:**

host - the host to connect to
public void connect(String user, String password) throws MessagingException

null

user

password

Throws: AuthenticationFailedException - for authentication failures

Throws: MessagingException - for other failures

Throws: IllegalStateException - if the service is already connected

See Also: ConnectionEvent, Session.setPasswordAuthentication(javax.mail.URLName, javax.mail.PasswordAuthentication)

public void connect(String user, String password) throws MessagingException

null

user

password

Throws: AuthenticationFailedException:

Throws: MessagingException:

Throws: IllegalStateException:

since JavaMail 1.4

See also setPasswordAuthentication, connect(java.lang.String, java.lang.String, java.lang.String)

**connect**

public void connect(String user, String password) throws MessagingException

Connect to the current host using the specified username and password. This method is equivalent to calling the connect(host, user, password) method with null for the host name.

**Parameters:**

user - the user name
password - this user's password

**Throws:**
- `AuthenticationFailedException` - for authentication failures
- `MessagingException` - for other failures
- `IllegalStateException` - if the service is already connected

**Since:**
JavaMail 1.4

**See Also:**
- `ConnectionEvent`
- `Session.setPasswordAuthentication(javax.mail.URLName, javax.mail.PasswordAuthentication)`
- `connect(java.lang.String, java.lang.String, java.lang.String)`

---

```java
public void connect(String host, int port, String user, String password)
throws MessagingException
```

**connect(host, user, password)**

- `host`
- `port` -1
- `user`
- `password`

**Throws**
- `AuthenticationFailedException`
- `MessagingException`
- `IllegalStateException`

**See also**
- `javax.mail.event.ConnectionEvent`

---

```java
public void connect(String host, int port, String user, String password)
throws MessagingException
```

**connect**

- `String host`,
- `int port`,
- `String user`,
- `String password`

**throws** `MessagingException`

Similar to `connect(host, user, password)` except a specific port can be specified.
Parameters:
  host - the host to connect to
  port - the port to connect to (-1 means the default port)
  user - the user name
  password - this user's password

Throws:
  AuthenticationFailedException - for authentication failures
  MessagingException - for other failures
  IllegalStateException - if the service is already connected

See Also:
  connect(java.lang.String, java.lang.String,
           java.lang.String), ConnectionEvent

protected boolean protocolConnect(String host, int port,
String user, String password) throws MessagingException

null protocolConnect fals
false AuthenticationFailedException String

protocolConnect
  host
  port
  user
  password
  return

Throws AuthenticationFailedException:
Throws MessagingException:

protocolConnect

protected boolean protocolConnect(String host,
  int port,
  String user,
  String password)
The service implementation should override this method to perform the actual protocol-specific connection attempt. The default implementation of the `connect` method calls this method as needed.

The `protocolConnect` method should return `false` if a user name or password is required for authentication but the corresponding parameter is null; the `connect` method will prompt the user when needed to supply missing information. This method may also return `false` if authentication fails for the supplied user name or password. Alternatively, this method may throw an `AuthenticationFailedException` when authentication fails. This exception may include a String message with more detail about the failure.

The `protocolConnect` method should throw an exception to report failures not related to authentication, such as an invalid host name or port number, loss of a connection during the authentication process, unavailability of the server, etc.

**Parameters:**
- `host` - the name of the host to connect to
- `port` - the port to use (-1 means use default port)
- `user` - the name of the user to login as
- `password` - the user's password

**Returns:**
- true if connection successful, false if authentication failed

**Throws:**
- `AuthenticationFailedException` - for authentication failures
- `MessagingException` - for non-authentication failures

---

public boolean isConnected()
isConnected

public boolean isConnected()

Is this service currently connected?

This implementation uses a private boolean field to store the connection state. This method returns the value of that field.

Subclasses may want to override this method to verify that any connection to the message store is still alive.

**Returns:**
true if the service is connected, false if it is not connected

---

protected void setConnected(boolean connected)

connect  close

isConnected
connected  true false

setConnected

protected void setConnected(boolean connected)

Set the connection state of this service. The connection state will automatically be set by the service implementation during the connect and close methods. Subclasses will need to call this method to set the state if the service was automatically disconnected.

The implementation in this class merely sets the private field returned by the isConnected method.
Parameters:

- **connected**: true if the service is connected, false if it is not connected

---

public void close() throws **MessagingException**

ConnectionListener  close ConnectionEvent

Messaging FolderMessage

MessagingException

setConnected(false)  **connected**: false

ConnectionListener  close ConnectionEvent

finally  super.close()

Throws  **MessagingException**:  

See also  javax.mail.event.ConnectionEvent

---

**close**

public void **close()**

throws **MessagingException**

Close this service and terminate its connection. A close ConnectionEvent is delivered to any ConnectionListeners. Any Messaging components (Folders, Messages, etc.) belonging to this service are invalid after this service is closed. Note that the service is closed even if this method terminates abnormally by throwing a MessagingException.

This implementation uses setConnected(false) to set this service's connected state to false. It will then send a close ConnectionEvent to any registered ConnectionListeners. Subclasses overriding this method to do implementation specific cleanup should call this method as a last step to insure event notification, probably by including a call to super.close() in a finally clause.

Throws:
public URLName getURLName()

Return a URLName representing this service. The returned URLName does *not* include the password field.

Subclasses should only override this method if their URLName does not follow the standard format.

The implementation in the Service class returns (usually a copy of) the url field with the password and file information stripped out.

**Returns:**
the URLName representing this service

**See Also:**
URLName
### setURLName

```java
protected void setURLName(URLName url)
```

Set the URLName representing this service. Normally used to update the `url` field after a service has successfully connected.

Subclasses should only override this method if their URL does not follow the standard format. In particular, subclasses should override this method if their URL does not require all the possible fields supported by `URLName`; a new `URLName` should be constructed with any unneeded fields removed.

The implementation in the `Service` class simply sets the `url` field.

See Also:
- [URLName](javax.mail.URLName)

---

### public void addConnectionListener(ConnectionListener l)

#### Connection

```java
public void addConnectionListener(ConnectionListener l)
```

`ConnectionListener` is the interface that represents a connection listener. See also [javax.mail.event.ConnectionEvent](javax.mail.event.ConnectionEvent)

### addConnectionListener

```java
public void addConnectionListener(ConnectionListener l)
```
Add a listener for Connection events on this service.

The default implementation provided here adds this listener to an internal list of ConnectionListeners.

**Parameters:**
- `l` - the Listener for Connection events

**See Also:**
- [ConnectionEvent](#)

---

```java
public void removeConnectionListener(ConnectionListener l)
```

**Connection**

**ConnectionListener**

Remove a Connection event listener.

The default implementation provided here removes this listener from the internal list of ConnectionListeners.

**Parameters:**
- `l` - the listener

**See Also:**
- [addConnectionListener](#)

---

```java
protected void notifyConnectionListeners(int type)
```

**Connection**

**ConnectionListener**

Connection
ConnectionListener

**notifyConnectionListeners**

```java
protected void notifyConnectionListeners(int type)
```

Notify all ConnectionListeners. Service implementations are expected to use this method to broadcast connection events.

The provided default implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered ConnectionListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

**public String toString()**

```java
URLName toString() getURLName.toString() toString
```

**toString**

```java
public String toString()
```

Return getURLName.toString() if this service has a URLName, otherwise it will return the default toString.

**Overrides:**

`toString` in class `Object`

**protected void queueEvent(MailEvent event, java.util.Vector<E> vector)**
queueEvent

protected void queueEvent(MailEvent event, Vector vector)

Add the event and vector of listeners to the queue to be delivered.

protected void finalize() throws Throwable

finalize

protected void finalize() throws Throwable

Stop the event dispatcher thread so the queue can be garbage collected.

Overrides:
   finalize in class Object

Throws:
   Throwable
javax.xml.registry.infomodel Interface Service

All Superinterfaces:
   ExtensibleObject, RegistryEntry, RegistryObject, Versionable

public interface Service
extends RegistryEntry

Implements: RegistryEntry

Service  Organization  Web  RegistryObject  Service
ServiceBinding  UDDI  BusinessService

See also   javax.xml.registry.infomodel.ServiceBinding

Service instances are RegistryObjects that provide information on services (for example, web services) offered by an Organization. A Service may have a set of ServiceBinding instances. Maps to a BusinessService in UDDI.

Author:
   Farrukh S. Najmi
See Also:
   ServiceBinding

Field Summary

Fields inherited from interface
javax.xml.registry.infomodel.RegistryEntry
STABILITY_DYNAMIC, STABILITY_DYNAMIC_COMPATIBLE, STABILITY_STATIC,
STATUS_APPROVED, STATUS_DEPRECATED, STATUS_SUBMITTED,
STATUS_WITHDRAWN
<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addServiceBinding</code></td>
<td><code>addServiceBinding(ServiceBinding serviceBinding)</code></td>
<td>Adds a child ServiceBinding.</td>
</tr>
<tr>
<td><code>void addServiceBindings</code></td>
<td><code>addServiceBindings(Collection serviceBindings)</code></td>
<td>Adds a Collection of ServiceBinding children.</td>
</tr>
<tr>
<td><code>Organization getProvidingOrganization</code></td>
<td></td>
<td>Gets the Organization that provides this service.</td>
</tr>
<tr>
<td><code>Collection getServiceBindings</code></td>
<td></td>
<td>Gets all children ServiceBindings.</td>
</tr>
<tr>
<td><code>void removeServiceBinding</code></td>
<td><code>removeServiceBinding(ServiceBinding serviceBinding)</code></td>
<td>Removes a child ServiceBinding.</td>
</tr>
<tr>
<td><code>void removeServiceBindings</code></td>
<td><code>removeServiceBindings(Collection serviceBindings)</code></td>
<td>Removes a Collection of children ServiceBindings.</td>
</tr>
<tr>
<td><code>void setProvidingOrganization</code></td>
<td><code>setProvidingOrganization(Organization providingOrganization)</code></td>
<td>Sets the Organization that provides this service.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.registry.infomodel.RegistryEntry`
- getExpiration
- getStability
- getStatus
- setExpiration
- setStability

Methods inherited from interface `javax.xml.registry.infomodel.RegistryObject`
- addAssociation
- addAssociations
- addClassification
- addClassifications
- addExternalIdentifier
- addExternalIdentifiers
- addExternallink
- addExternalLinks
- getAssociatedObjects
- getAssociations
- getAuditTrail
- getClassifications
- getDescription
- getExternalIdentifiers
- getExternalLinks
- getKey
- getLifeCycleManager
- getName
- getObjectType
- getRegistryPackages
- getSubmittingOrganization
- removeAssociation
- removeAssociations
- removeClassification
- removeClassifications
- removeExternalIdentifier
- removeExternalIdentifiers
- removeExternallink
- removeExternalLinks
- setAssociations
- setClassifications
- setDescription
- setExternalIdentifiers
- setExternallinks
- setKey
- setName
- toXML

Methods inherited from interface `javax.xml.registry.infomodel.ExtensibleObject`
public Organization getProvidingOrganization() throws JAXRException

    Organization Organization  null Organization
    RegistryObject#getSubmittingOrganization
    Submitting Organization

0

    return Organization
    Throws JAXRException: JAXR
    See also getSubmittingOrganization()

getProvidingOrganization

    Organization getProvidingOrganization() throws JAXRException

    Gets the Organization that provides this service. Providing Organization may be null. The providing Organization may be different from the Submitting Organization as defined by RegistryObject#getSubmittingOrganization.

    Capability Level: 0

    Returns:
the Organization that provides this service

Throws:

JAXRException - If the JAXR provider encounters an internal error

See Also:

RegistryObject.getSubmittingOrganization()

public void setProvidingOrganization(Organization providingOrganization) throws JAXRException
Organization

0

providingOrganization

Throws

JAXRException: JAXR

setProvidingOrganization

void setProvidingOrganization(Organization providingOrganization)
throws JAXException

Sets the Organization that provides this service.

Capability Level: 0

Parameters:

providingOrganization - the Organization that provides this service

Throws:

JAXRException - If the JAXR provider encounters an internal error

public void addServiceBinding(ServiceBinding serviceBinding) throws JAXRException
ServiceBinding
addServiceBinding

void addServiceBinding(ServiceBinding serviceBinding)
    throws JAXRException

Adds a child ServiceBinding.

Capability Level: 0

Parameters:
    serviceBinding - the ServiceBinding being added

Throws:
    JAXRException - If the JAXR provider encounters an internal error

public void addServiceBindings(java.util.Collection<E> serviceBindings) throws JAXRException
ServiceBinding Collection

addServiceBindings

void addServiceBindings(Collection serviceBindings)
    throws JAXRException

Adds a Collection of ServiceBinding children.
public void removeServiceBinding(ServiceBinding serviceBinding) throws JAXRException

Removes a child ServiceBinding.

Capability Level: 0

Parameters:
    serviceBinding - the ServiceBinding being removed

Throws:
    JAXRException - If the JAXR provider encounters an internal error
**removeServiceBindings**

```java
void removeServiceBindings(Collection serviceBindings)
throws JAXRException
```

Removes a Collection of children ServiceBindings.

**Capability Level:** 0

**Parameters:**
- `serviceBindings` - the Collection of ServiceBindings being removed

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

**public java.util.Collection<E> getServiceBindings() throws JAXRException**

```java
ServiceBinding Collection
null
```

**Throws:**
- `JAXRException` - JAXR

**supplierCardinality**

1..*

**associates**

<javas.xml.registry.infomodel.ServiceBinding>

aggregationByValue

en

**See also**

`javas.xml.registry.infomodel.ServiceBinding`
Collection getServiceBindings() throws JAXRException

Gets all children ServiceBindings.

**Capability Level: 0**

**Returns:**
Collection of ServiceBinding instances. The Collection may be empty but not null.

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

**See Also:**
ServiceBinding

---

**Overview**  **Package**  **Tree**  **Deprecated**  **Index**  **Help**

PREV CLASS  NEXT CLASS  SUMMARY: NESTED | FIELD | CONSTR | METHOD  FRAMES  NO FRAMES  DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc Interface Service

public interface Service

Implements: RegistryEntry

Service  Organization Web  RegistryObject Service  ServiceBinding  UDDI  BusinessService

See also  javax.xml.registry.infomodel.ServiceBinding

Service class acts as a factory of the following:

- Dynamic proxy for the target service endpoint.
- Instance of the type javax.xml.rpc.Call for the dynamic invocation of a remote operation on the target service endpoint.
- Instance of a generated stub class

Version: 1.0
Author: Rahul Sharma
See Also: Call, Stub

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Call</strong></td>
</tr>
<tr>
<td>createCall()</td>
</tr>
<tr>
<td>Creates a call object not associated with specific operation or target service endpoint.</td>
</tr>
<tr>
<td>createCall(QName portName)</td>
</tr>
<tr>
<td>Creates a call instance.</td>
</tr>
<tr>
<td>createCall(QName portName, QName operationName)</td>
</tr>
<tr>
<td>Creates a call instance.</td>
</tr>
<tr>
<td>createCall(QName portName, String operationName)</td>
</tr>
<tr>
<td>Creates a call instance.</td>
</tr>
</tbody>
</table>
### Method Detail

**getPort**

```
Remote getPort(QName portName,
```

**Returns**

The `getPort()` method returns either an instance of a generated stub implementation class or a dynamic proxy.

---

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>create()</code></td>
<td>Creates a <code>Call</code> instance.</td>
</tr>
<tr>
<td><code>getCalls(QName portName)</code></td>
<td>Gets an array of preconfigured <code>Call</code> objects for invoking operations on the specified port.</td>
</tr>
<tr>
<td><code>getHandlerRegistry()</code></td>
<td>Returns the configured <code>HandlerRegistry</code> instance for this <code>Service</code> instance.</td>
</tr>
<tr>
<td><code>getPort(Class serviceEndpointInterface)</code></td>
<td>The <code>getPort()</code> method returns either an instance of a generated stub implementation class or a dynamic proxy.</td>
</tr>
<tr>
<td><code>getPort(QName portName, Class serviceEndpointInterface)</code></td>
<td>The <code>getPort()</code> method returns either an instance of a generated stub implementation class or a dynamic proxy.</td>
</tr>
<tr>
<td><code>getPorts()</code></td>
<td>Returns an <code>Iterator</code> for the list of <code>QName</code>s of service endpoints grouped by this service.</td>
</tr>
<tr>
<td><code>getServiceName()</code></td>
<td>Gets the name of this service.</td>
</tr>
<tr>
<td><code>getTypeMappingRegistry()</code></td>
<td>Gets the <code>TypeMappingRegistry</code> for this <code>Service</code> object.</td>
</tr>
<tr>
<td><code>getWSDLDocumentLocation()</code></td>
<td>Gets the location of the WSDL document for this <code>Service</code>.</td>
</tr>
</tbody>
</table>
The getPort method returns either an instance of a generated stub implementation class or a dynamic proxy. A service client uses this dynamic proxy to invoke operations on the target service endpoint. The `serviceEndpointInterface` specifies the service endpoint interface that is supported by the created dynamic proxy or stub instance.

**Parameters:**
- `portName` - Qualified name of the service endpoint in the WSDL service description
- `serviceEndpointInterface` - Service endpoint interface supported by the dynamic proxy or stub instance

**Returns:**
java.rmi.Remote Stub instance or dynamic proxy that supports the specified service endpoint interface

**Throws:**
- `ServiceException` - This exception is thrown in the following cases:
  - If there is an error in creation of the dynamic proxy or stub instance
  - If there is any missing WSDL metadata as required by this method
  - Optionally, if an illegal `serviceEndpointInterface` or `portName` is specified

**See Also:**
- Proxy, InvocationHandler

---

`getPort`

```java
Remote getPort(Class serviceEndpointInterface)
        throws ServiceException
```

The `getPort` method returns either an instance of a generated stub implementation class or a dynamic proxy. The parameter
serviceEndpointInterface specifies the service endpoint interface that is supported by the returned stub or proxy. In the implementation of this method, the JAX-RPC runtime system takes the responsibility of selecting a protocol binding (and a port) and configuring the stub accordingly. The returned Stub instance should not be reconfigured by the client.

**Parameters:**
- serviceEndpointInterface - Service endpoint interface

**Returns:**
- Stub instance or dynamic proxy that supports the specified service endpoint interface

**Throws:**
- ServiceException -
  - If there is an error during creation of stub instance or dynamic proxy
  - If there is any missing WSDL metadata as required by this method
  - Optionally, if an illegal serviceEndpointInterface is specified

---

getCalls

**Call**[] get Calls(QName portName)

throws ServiceException

Gets an array of preconfigured call objects for invoking operations on the specified port. There is one call object per operation that can be invoked on the specified port. Each call object is pre-configured and does not need to be configured using the setter methods on call interface.

Each invocation of the get Calls method returns a new array of preconfigured call objects

This method requires the Service implementation class to have access to the WSDL related metadata.
createCall

```java
Call createCall(QName portName)
```

Throws: `ServiceException` - If any error in the creation of the call object

Parameters:
- `portName` - Qualified name for the target service endpoint

Returns: Call instance

createCall

```java
Call createCall(QName portName, QName operationName)
```

Throws: `ServiceException` - If any error in the creation of the call object

Parameters:
- `portName` - Qualified name for the target service endpoint
- `operationName` - Qualified Name of the operation for which this
call object is to be created.

**Returns:**
Call instance

**Throws:**
ServiceException - If any error in the creation of the call object

---

createCall

```java
Call createCall(QName portName,
String operationName)
throws ServiceException
```

Creates a call instance.

**Parameters:**
- portName - Qualified name for the target service endpoint
- operationName - Name of the operation for which this call object is to be created.

**Returns:**
Call instance

**Throws:**
ServiceException - If any error in the creation of the call object

---

createCall

```java
Call createCall()
throws ServiceException
```

Creates a call object not associated with specific operation or target service endpoint. This call object needs to be configured using the setter methods on the call interface.

**Returns:**
Call object
Throws:

ServiceException - If any error in the creation of the call object

---

getServiceName

QName getServiceName()

Gets the name of this service.

**Returns:**

Qualified name of this service

---

getPorts

Iterator getPorts() throws ServiceException

Returns an Iterator for the list of QNames of service endpoints grouped by this service

**Returns:**

Returns java.util.Iterator with elements of type javax.xml.namespace.QName

**Throws:**

ServiceException - If this Service class does not have access to the required WSDL metadata

---

getWSDLDocumentLocation

URL getWSDLDocumentLocation()
Gets the location of the WSDL document for this Service.

**Returns:**
URL for the location of the WSDL document for this service

---

**get TypeMappingRegistry**

TypeMappingRegistry getTypeMappingRegistry()

Gets the TypeMappingRegistry for this Service object. The returned TypeMappingRegistry instance is pre-configured to support the standard type mapping between XML and Java types as required by the JAX-RPC specification.

**Returns:**
The TypeMappingRegistry for this Service object.

**Throws:**
java.lang.UnsupportedOperationException - if the Service class does not support the configuration of TypeMappingRegistry.

---

**getHandlerRegistry**

HandlerRegistry getHandlerRegistry()

Returns the configured HandlerRegistry instance for this Service instance.

**Returns:**
HandlerRegistry

**Throws:**
java.lang.UnsupportedOperationException - if the Service class does not support the configuration of a HandlerRegistry.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws  **Class Service**

**java.lang.Object**
  - **javax.xml.ws.Service**

```java
public class Service
extends Object

Implements: RegistryEntry
```

Service  Organization Web  RegistryObject Service  ServiceBinding  UDDI  BusinessService

**See also**  **javax.xml.registry.infomodel.ServiceBinding**

Service objects provide the client view of a Web service.

Service acts as a factory of the following:

- Proxies for a target service endpoint.
- Instances of `javax.xml.ws.Dispatch` for dynamic message-oriented invocation of a remote operation.

The ports available on a service can be enumerated using the `getPorts` method. Alternatively, you can pass a service endpoint interface to the `UnaryPort` method and let the runtime select a compatible port.

Handler chains for all the objects created by a `Service` can be set by means of a `HandlerResolver`.

An `Executor` may be set on the service in order to gain better control over the threads used to dispatch asynchronous callbacks. For instance, thread pooling with certain parameters can be enabled by creating a `ThreadPoolExecutor` and registering it with the service.

**Since:**
JAX-WS 2.0

See Also:
Provider, HandlerResolver, Executor

## Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>Service.Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The orientation of a dynamic client or service.</td>
</tr>
</tbody>
</table>

## Constructor Summary

| protected Service | (URL wsdlDocumentLocation, QName serviceName) |

## Method Summary

<table>
<thead>
<tr>
<th>void</th>
<th>addPort(QName portName, String bindingId, String endpointAddress)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a new port for the service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static Service</th>
<th>create(QName serviceName)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Create a Service instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static Service</th>
<th>create(URL wsdlDocumentLocation, QName serviceName)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Create a Service instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>&lt;T&gt; Dispatch&lt;T&gt;</th>
<th>createDispatch(QName portName, Class&lt;T&gt; type, Service.Mode mode)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a Dispatch instance for use with objects of the users choosing. Dispatch&lt;Object&gt; createDispatch(QName portName, JAXBContext context, Service.Mode mode)</td>
</tr>
<tr>
<td></td>
<td>Creates a Dispatch instance for use with JAXB generated objects. Executor getExecutor()</td>
</tr>
<tr>
<td></td>
<td>Returns the executor for this Service instance. HandlerResolver getHandlerResolver()</td>
</tr>
<tr>
<td></td>
<td>Returns the configured handler resolver.</td>
</tr>
</tbody>
</table>

| <T> | getPort(Class<T> serviceEndpointInterface) |
The `getPort` method returns a stub.

```java
public <T> T getPort(QName portName, Class<T> serviceEndpointInterface)
```

The `getPort` method returns a stub. A service client uses this stub to invoke operations on the target service endpoint. The `serviceEndpointInterface` specifies the service endpoint.
interface that is supported by the created dynamic proxy or stub instance.

**Parameters:**
- **portName** - Qualified name of the service endpoint in the WSDL service description
- **serviceEndpointInterface** - Service endpoint interface supported by the dynamic proxy or stub instance

**Returns:**
Object Proxy instance that supports the specified service endpoint interface

**Throws:**
- **WebServiceException** - This exception is thrown in the following cases:
  - If there is an error in creation of the proxy
  - If there is any missing WSDL metadata as required by this method
  - Optionally, if an illegal **serviceEndpointInterface** or **portName** is specified

**See Also:**
- [Proxy](#), [InvocationHandler](#)

---

getPort

public `<T>` T getPort(Class `<T>` serviceEndpointInterface)

The getPort method returns a stub. The parameter **serviceEndpointInterface** specifies the service endpoint interface that is supported by the returned proxy. In the implementation of this method, the JAX-WS runtime system takes the responsibility of selecting a protocol binding (and a port) and configuring the proxy accordingly. The returned proxy should not be reconfigured by the client.

**Parameters:**
- **serviceEndpointInterface** - Service endpoint interface

**Returns:**
Object instance that supports the specified service endpoint interface

**Throws:**

- `WebServiceException` -
  - If there is an error during creation of the proxy
  - If there is any missing WSDL metadata as required by this method
  - Optionally, if an illegal `serviceEndpointInterface` is specified

### addPort

```java
public void addPort(QName portName,
                     String bindingId,
                     String endpointAddress)
```

Creates a new port for the service. Ports created in this way contain no WSDL port type information and can only be used for creating `Dispatch` instances.

**Parameters:**
- `portName` - Qualified name for the target service endpoint
- `bindingId` - A `String` identifier of a binding.
- `endpointAddress` - Address of the target service endpoint as a `URI`

**Throws:**

- `WebServiceException` - If any error in the creation of the port

**See Also:**

- `SOAPBinding.SOAP11HTTP_BINDING`, `SOAPBinding.SOAP12HTTP_BINDING`, `HTTPBinding.HTTP_BINDING`

### createDispatch

```java
public <T> Dispatch<T> createDispatch(QName portName,
                                       Class<T> type,
                                       ...
```


Service.Mode (mode)

Creates a Dispatch instance for use with objects of the users choosing.

Parameters:
- portName - Qualified name for the target service endpoint
- type - The class of object used to messages or message payloads. Implementations are required to support javax.xml.transform.Source and javax.xml.soap.SOAPMessage.
- mode - Controls whether the created dispatch instance is message or payload oriented, i.e. whether the user will work with complete protocol messages or message payloads. E.g. when using the SOAP protocol, this parameter controls whether the user will work with SOAP messages or the contents of a SOAP body. Mode must be MESSAGE when type is SOAPMessage.

Returns:
- Dispatch instance

Throws:
- WebServiceException - If any error in the creation of the Dispatch object

See Also:
- Source, SOAPMessage

createDispatch

public Dispatch<Object> createDispatch(QName portName,
                                           JAXBContext context,
                                           Service.Mode mode)

Creates a Dispatch instance for use with JAXB generated objects.

Parameters:
- portName - Qualified name for the target service endpoint
- context - The JAXB context used to marshall and
unmarshall messages or message payloads.
mode - Controls whether the created dispatch instance is message or payload oriented, i.e. whether the user will work with complete protocol messages or message payloads. E.g. when using the SOAP protocol, this parameter controls whether the user will work with SOAP messages or the contents of a SOAP body.

**Returns:**
Dispatch instance

**Throws:**
ServiceException - If any error in the creation of the Dispatch object

**See Also:**
JAXBContext

---

**getServiceName**

public QName getServiceName()

Gets the name of this service.

**Returns:**
Qualified name of this service

---

**getPorts**

public Iterator<QName> getPorts()

Returns an Iterator for the list of QNames of service endpoints grouped by this service

**Returns:**
Returns java.util.Iterator with elements of type javax.xml.namespace.QName

**Throws:**
**WebServiceException** - If this Service class does not have access to the required WSDL metadata

---

**getWSDLDocumentLocation**

```java
public URL getWSDLDocumentLocation()
```

Gets the location of the WSDL document for this Service.

**Returns:**
URL for the location of the WSDL document for this service

---

**getHandlerResolver**

```java
public HandlerResolver getHandlerResolver()
```

Returns the configured handler resolver.

**Returns:**
HandlerResolver The HandlerResolver being used by this Service instance, or null if there isn't one.

---

**setHandlerResolver**

```java
public void setHandlerResolver(HandlerResolver handlerResolver)
```

Sets the HandlerResolver for this Service instance.

The handler resolver, if present, will be called once for each proxy or dispatch instance that is created, and the handler chain returned by the resolver will be set on the instance.

**Parameters:**
**HandlerResolver** - The HandlerResolver to use for all subsequently created proxy/dispatch objects.

**See Also:**
- [HandlerResolver](#)

---

**getExecutor**

```java
public Executor getExecutor()
```

Returns the executor for this service instance. The executor is used for all asynchronous invocations that require callbacks.

**Returns:**
- The `java.util.concurrent.Executor` to be used to invoke a callback.

**See Also:**
- [Executor](#)

---

**setExecutor**

```java
public void setExecutor(Executor executor)
```

Sets the executor for this service instance. The executor is used for all asynchronous invocations that require callbacks.

**Parameters:**
- `executor` - The `java.util.concurrent.Executor` to be used to invoke a callback.

**Throws:**
- [SecurityException](#) - If the instance does not support setting an executor for security reasons (e.g. the necessary permissions are missing).

**See Also:**
- [Executor](#)
Create a Service instance. The specified WSDL document location and service qualified name must uniquely identify a wsdl:service element.

**Parameters:**
- `wsdlDocumentLocation` - URL for the WSDL document location for the service
- `serviceName` - QName for the service

**Throws:**
- `WebServiceException` - If any error in creation of the specified service
subject to license terms.

PS:
javax.xml.ws  Enum Service.Mode

java.lang.Object
  \ java.lang.Enum<Service.Mode>
    \ javax.xml.ws.Service.Mode

All Implemented Interfaces:
  Serializable, Comparable<Service.Mode>

Enclosing class:
  Service

---

public static enum Service.Mode
extends Enum<Service.Mode>

Extends: Enum<E>
Contained within: Service

MESSAGE  PAYLOAD

The orientation of a dynamic client or service. MESSAGE provides access to entire protocol message, PAYLOAD to protocol message payload only.

---

<table>
<thead>
<tr>
<th>Enum Constant Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MESSAGE</td>
</tr>
<tr>
<td>PAYLOAD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueOf(String name)</td>
</tr>
<tr>
<td>static Service.Mode</td>
</tr>
<tr>
<td>---------------------</td>
</tr>
<tr>
<td>static Service.Mode[] values()</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

MESSAGE
public static final Service.Mode MESSAGE

PAYLOAD
public static final Service.Mode PAYLOAD

Method Detail

final public static Service.Mode[] values()
public static final Service.Mode[] values()

    Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

    for(Service.Mode c : Service.Mode.values())
        System.out.println(c);

    Returns:
    an array containing the constants of this enum type, in the order they're declared

public static Service.Mode valueOf(String name)

valueOf

public static Service.Mode valueOf(String name)

    Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

    Parameters:
    name - the name of the enum constant to be returned.

    Returns:
    the enum constant with the specified name

    Throws:
    IllegalArgumentException - if this enum type has no constant with the specified name

Submit a bug or feature
javax.xml.registry.infomodel Interface ServiceBinding

All Superinterfaces:
   ExtensibleObject, RegistryObject, URIValidator

public interface ServiceBinding
extends RegistryObject, URIValidator

Implements: RegistryObject, URIValidator

ServiceBinding RegistryObject Service ServiceBinding
SpecificationLink UDDI BindingTemplate

See also      javax.xml.registry.infomodel.Concept

ServiceBinding instances are RegistryObjects that represent technical
information on a specific way to access a specific interface offered by a
Service instance. A ServiceBinding may have a set of SpecificationLink
instances. Maps to a BindingTemplate in UDDI.

Author:
   Farrukh S. Najmi

See Also:
   Concept

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addSpecificationLink(SpecificationLink specificationLink)</td>
<td>Adds a child SpecificationLink.</td>
</tr>
<tr>
<td>String getAccessURI()</td>
<td>Gets the URI that gives access to the service via this binding.</td>
</tr>
</tbody>
</table>
### Service

**getService()**
- Gets the parent service for which this is a binding.

### Collection

**getSpecificationLinks()**
- Gets all children SpecificationLinks.

### ServiceBinding

**getTargetBinding()**
- Gets the next ServiceBinding in case there is a redirection from one service provider to another service provider.

**removeSpecificationLink**(*SpecificationLink* specificationLink)
- Removes a child SpecificationLink.

**removeSpecificationLinks**(*Collection* specificationLinks)

**setAccessURI**(*String* uri)
- Sets the URI that gives access to the service via this binding.

**setTargetBinding**(*ServiceBinding* binding)
- Sets the next ServiceBinding in case there is a redirection.

---

**Methods inherited from interface**

*javax.xml.registry.infomodel.RegistryObject*

- `addAssociation`, `addAssociations`, `addClassification`, `addClassifications`, `addExternalIdentifier`, `addExternalIdentifiers`, `addExternalLink`, `addExternalLinks`, `getAssociatedObjects`, `getAuditTrail`, `getClassifications`, `getDescription`, `getExternalIdentifiers`, `getExternalLinks`, `getKey`, `getLifeCycleManager`, `getName`, `getObjectType`, `getRegistryPackages`, `getSubmittingOrganization`, `removeAssociation`, `removeAssociations`, `removeClassification`, `removeClassifications`, `removeExternalIdentifier`, `removeExternalIdentifiers`, `removeExternalLink`, `removeExternalLinks`, `setAssociations`, `setClassifications`, `setDescription`, `setExternalIdentifiers`, `setExternalLinks`, `setKey`, `setName`, `toXML`  

---

**Methods inherited from interface**

*javax.xml.registry.infomodel.ExtensibleObject*

- `addSlot`, `addSlots`, `getSlot`, `getSlots`, `removeSlot`, `removeSlots`  

---

**Methods inherited from interface**

*javax.xml.registry.infomodel.URIValidator*
public String getAccessURI() throws JAXRException
URI NULL

0

    return URI

Throws JAXRException: JAXR

getAccessURI

String getAccessURI() throws JAXRException

    Gets the URI that gives access to the service via this binding. Default is a NULL String.

    Capability Level: 0

    Returns: the URI that gives access to the service via this binding

    Throws: JAXRException - If the JAXR provider encounters an internal error

public void setAccessURI(String uri) throws JAXRException
    URI accessURI targetBinding null
targetBinding accessURI JAXR
InvalidRequestException
setAccessURI

```java
void setAccessURI(String uri)
    throws JAXRException
```

Sets the URI that gives access to the service via this binding. The accessURI is mutually exclusive from targetBinding. JAXR Provider must throw an InvalidRequestException if an accessURI is set when there is already a non-null targetBinding defined.

**Capability Level: 0**

**Parameters:**
- **uri** - the URI that gives access to the service via this binding

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

---

```java
public ServiceBinding getTargetBinding() throws JAXRException
    ServiceBinding
```

```
return ServiceBinding
    throws JAXRException: JAXR
```

**getTargetBinding**

```java
ServiceBinding getTargetBinding()
```
Throws: 
   JAXRException - If the JAXR provider encounters an internal error

public void setTargetBinding(ServiceBinding binding) 
throws JAXRException
  ServiceBinding targetBinding accessURI null
  accessURI targetBinding JAXR
  InvalidRequestException

sets the next ServiceBinding in case there is a redirection. The targetBinding is mutually exclusive from the accessURI. JAXR Provider must throw an InvalidRequestException if a targetBinding is set when there is already a non-null accessURI defined.

Capability Level: 0
Parameters:
binding - the target ServiceBinding to which this object is redirected to

Throws:
JAXRException - If the JAXR provider encounters an internal error

public Service getService() throws JAXRException

0

return Service
Throws JAXRException: JAXR

getService

Service getService() throws JAXRException

Gets the parent service for which this is a binding.

Capability Level: 0

Returns:
the parent Service object

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void addSpecificationLink(SpecificationLink specificationLink) throws JAXRException
SpecificationLink

0
addSpecificationLink

void addSpecificationLink(SpecificationLink specificationLink)
throws JAXRException

Adds a child SpecificationLink.

Capability Level: 0

Parameters:
  specificationLink - the SpecificationLink being added

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void addSpecificationLinks(java.util.Collection&lt;E&gt; specificationLinks) throws JAXRException
SpecificationLink Collection

0

addSpecificationLinks

void addSpecificationLinks(Collection specificationLinks)
throws JAXRException

Adds a Collection of SpecificationLink children.

Capability Level: 0
Parameters:
   specificationLinks - the Collection of SpecificationLinks being added

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public void removeSpecificationLink(SpecificationLink specificationLink) throws JAXRException

removeSpecificationLink

void removeSpecificationLink(SpecificationLink specificationLink) throws JAXRException

Removes a child SpecificationLink.

Capability Level: 0

Parameters:
   specificationLink - the SpecificationLink being removed

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public void removeSpecificationLinks(java.util.Collection<E> specificationLinks) throws JAXRException

removeSpecificationLinks
removeSpecificationLinks

void removeSpecificationLinks(Collection specificationLinks)
throws JAXRException

Removes a Collection of children SpecificationLinks.

**Capability Level: 0**

**Parameters:**

- specificationLinks - the Collection of SpecificationLinks being removed

**Throws:**

- JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getSpecificationLinks()
throws JAXRException
SpecificationLink

return SpecificationLink Collection
null

Throws

JAXRException: JAXR

1..*

supplierCardinality associates

<jax.xml.registry.infomodel.SpecificationLink>
aggregationByValue en

See also

javax.xml.registry.infomodel.SpecificationLink
getSpecificationLinks

Collection getSpecificationLinks()
throws JAXRException

Gets all children SpecificationLinks.

Capability Level: 0

Returns:
Collection of SpecificationLink instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
SpecificationLink
javax.xml.ws.spi  Class ServiceDelegate

java.lang.Object
   javax.xml.ws.spi.ServiceDelegate

public abstract class ServiceDelegate
   extends Object

Service JAX-WS

Service javax.xml.ws.Provider#createServiceDelegate
Service since JAX-WS 2.0
See also javax.xml.ws.Service, javax.xml.ws.spi.Provider

Service delegates are used internally by Service objects to allow pluggability of JAX-WS implementations.

Every Service object has its own delegate, created using the javax.xml.ws.Provider#createServiceDelegate method. A Service object delegates all of its instance methods to its delegate.

Since:
   JAX-WS 2.0
See Also:
   Service, Provider

Constructor Summary

protected ServiceDelegate()
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addPort</code></td>
<td><code>void addPort(QName portName, String bindingId, String endpointAddress)</code></td>
<td>Creates a new port for the service.</td>
</tr>
<tr>
<td><code>createDispatch</code></td>
<td><code>&lt;T&gt;</code> <code>createDispatch(QName portName, Class&lt;T&gt; type, Service.Mode mode)</code></td>
<td>Creates a Dispatch instance for use with objects of the users choosing.</td>
</tr>
<tr>
<td><code>createDispatch</code></td>
<td><code>&lt;T&gt;</code> <code>createDispatch(QName portName, JAXBContext context, Service.Mode mode)</code></td>
<td>Creates a Dispatch instance for use with JAXB generated objects.</td>
</tr>
<tr>
<td><code>getExecutor</code></td>
<td><code>Executor getExecutor()</code></td>
<td>Returns the executor for this Service instance.</td>
</tr>
<tr>
<td><code>getHandlerResolver</code></td>
<td><code>HandlerResolver getHandlerResolver()</code></td>
<td>Returns the configured handler resolver.</td>
</tr>
<tr>
<td><code>getPort</code></td>
<td><code>&lt;T&gt;</code> <code>T getPort(Class&lt;T&gt; serviceEndpointInterface)</code></td>
<td>The getPort method returns a stub.</td>
</tr>
<tr>
<td><code>getPort</code></td>
<td><code>&lt;T&gt;</code> <code>T getPort(QName portName, Class&lt;T&gt; serviceEndpointInterface)</code></td>
<td>The getPort method returns a stub.</td>
</tr>
<tr>
<td><code>getPorts</code></td>
<td><code>Iterator&lt;QName&gt; getPorts()</code></td>
<td>Returns an Iterator for the list of QNames of service endpoints grouped by this service.</td>
</tr>
<tr>
<td><code>getServiceName</code></td>
<td><code>QName getServiceName()</code></td>
<td>Gets the name of this service.</td>
</tr>
<tr>
<td><code>getWSDLDocumentLocation</code></td>
<td><code>URL getWSDLDocumentLocation()</code></td>
<td>Gets the location of the WSDL document for this Service.</td>
</tr>
<tr>
<td><code>setExecutor</code></td>
<td><code>void setExecutor(Executor executor)</code></td>
<td>Sets the executor for this Service instance.</td>
</tr>
<tr>
<td><code>setHandlerResolver</code></td>
<td><code>void setHandlerResolver(HandlerResolver handlerResolver)</code></td>
<td>Sets the HandlerResolver for this Service instance.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object: `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait`
### Constructor Detail

**protected ServiceDelegate()**

**ServiceDelegate**

**protected ServiceDelegate()**

### Method Detail

**getPort**

```java
public abstract <T> T getPort(QName portName, Class<T> serviceEndpointInterface)
```

The `getPort` method returns a stub. A service client uses this stub to invoke operations on the target service endpoint. The `serviceEndpointInterface` specifies the service endpoint interface that is supported by the created dynamic proxy or stub instance.

**Parameters:**
- `portName` - Qualified name of the service endpoint in the WSDL service description
- `serviceEndpointInterface` - Service endpoint interface supported by the dynamic proxy or stub instance

**Returns:**
Object Proxy instance that supports the specified service endpoint interface

**Throws:**
- `WebServiceException` - This exception is thrown in the following cases:
  - If there is an error in creation of the proxy
  - If there is any missing WSDL metadata as required by this method
getPort

public abstract <T> T getPort(Class<T> serviceEndpointInterface)

The getPort method returns a stub. The parameter serviceEndpointInterface specifies the service endpoint interface that is supported by the returned proxy. In the implementation of this method, the JAX-WS runtime system takes the responsibility of selecting a protocol binding (and a port) and configuring the proxy accordingly. The returned proxy should not be reconfigured by the client.

Parameters:
   serviceEndpointInterface - Service endpoint interface

Returns:
   Object instance that supports the specified service endpoint interface

Throws:
   WebServiceException -
   • If there is an error during creation of the proxy
   • If there is any missing WSDL metadata as required by this method
   • Optionally, if an illegal serviceEndpointInterface is specified
addPort

public abstract void addPort(QName portName,
        String bindingId,
        String endpointAddress)

Creates a new port for the service. Ports created in this way contain no WSDL port type information and can only be used for creating Dispatch instances.

Parameters:
    portName - Qualified name for the target service endpoint
    bindingId - A URI identifier of a binding.
    endpointAddress - Address of the target service endpoint as a URI

Throws:
    WebServiceException - If any error in the creation of the port

See Also:
    SOAPBinding.SOAP11HTTP_BINDING,
    SOAPBinding.SOAP12HTTP_BINDING,
    HTTPBinding.HTTP_BINDING

createDispatch

public abstract <T> Dispatch<T> createDispatch(QName portName,
        Class<T> type,
        Service.Mode mode)

Creates a Dispatch instance for use with objects of the users choosing.
Parameters:
portName - Qualified name for the target service endpoint
type - The class of object used to messages or message payloads. Implementations are required to support javax.xml.transform.Source and javax.xml.soap.SOAPMessage.
mode - Controls whether the created dispatch instance is message or payload oriented, i.e. whether the user will work with complete protocol messages or message payloads. E.g. when using the SOAP protocol, this parameter controls whether the user will work with SOAP messages or the contents of a SOAP body. Mode must be MESSAGE when type is SOAPMessage.

Returns:
Dispatch instance

Throws:
WebServiceException - If any error in the creation of the Dispatch object

See Also:
Source, SOAPMessage

abstract public Dispatch createDispatch(javax.xml.namespace.QName portName, JAXBContext context, Service.Mode mode)

JAXB Dispatch
portName Dispatch
context JAXB
mode SOAP SOAP SOAP

return dispatch
Throws ServiceException: Dispatch
See also javax.xml.bind.JAXBContext

createDispatch

public abstract Dispatch<Object> createDispatch(QName portName, JAXBContext context,
Creates a Dispatch instance for use with JAXB generated objects.

**Parameters:**
- `portName` - Qualified name for the target service endpoint
- `context` - The JAXB context used to marshall and unmarshall messages or message payloads.
- `mode` - Controls whether the created dispatch instance is message or payload oriented, i.e. whether the user will work with complete protocol messages or message payloads. E.g. when using the SOAP protocol, this parameter controls whether the user will work with SOAP messages or the contents of a SOAP body.

**Returns:**
- Dispatch instance

**Throws:**
- `ServiceException` - If any error in the creation of the Dispatch object

**See Also:**
- `JAXBContext`

```java
abstract public javax.xml.namespace.QName getServiceName()
```

```java
return
```

**getServiceName**

```java
public abstract QName getServiceName()
```

Gets the name of this service.

**Returns:**
- Qualified name of this service
abstract public java.util.Iterator<E> getPorts()

    Iterator QName
    return javax.xml.namespace.QName java.util.Iterator
    Throws WebServiceException: Service WSDL

getPorts

public abstract Iterator<QName> getPorts()

    Returns an Iterator for the list of QName of service endpoints grouped by this service

Returns:
    Returns java.util.Iterator with elements of type javax.xml.namespace.QName

Throws:
    WebServiceException - If this Service class does not have access to the required WSDL metadata

abstract public java.net.URL getWSDLDocumentLocation()

WSDL

    return WSDL URL

getWSDLDocumentLocation

public abstract URL getWSDLDocumentLocation()

    Gets the location of the WSDL document for this Service.

Returns:
    URL for the location of the WSDL document for this service

abstract public HandlerResolver getHandlerResolver()
getHandlerResolver

public abstract HandlerResolver getHandlerResolver()

Returns the configured handler resolver.

Returns:
   HandlerResolver The HandlerResolver being used by this Service instance, or null if there isn’t one.

Note:
   HandlerResolvers are not thread safe. They should be thread safe or be created on a thread pool.

abstract public void setHandlerResolver(HandlerResolver handlerResolver)

Sets the HandlerResolver for this Service instance.

The handler resolver, if present, will be called once for each proxy or dispatch instance that is created, and the handler chain returned by the resolver will be set on the instance.

Parameters:
   handlerResolver - The HandlerResolver to use for all subsequently created proxy/dispatch objects.

See Also:
   javax.xml.ws.handler.HandlerResolver
abstract public java.util.concurrent.Executor getExecutor()

Service

return java.util.concurrent.Executor

See also java.util.concurrent.Executor

getExecutor

public abstract Executor getExecutor()

Returns the executor for this Service instance. The executor is used for all asynchronous invocations that require callbacks.

Returns:
The java.util.concurrent.Executor to be used to invoke a callback.

See Also:
Executor

abstract public void setExecutor(java.util.concurrent.Executor executor)

Service

executor java.util.concurrent.Executor

Throws SecurityException:

See also java.util.concurrent.Executor

setExecutor

public abstract void setExecutor(Executor executor)

Sets the executor for this Service instance. The executor is used for all asynchronous invocations that require callbacks.
Parameters:
executor - The java.util.concurrent.Executor to be used to invoke a callback.

Throws:
SecurityException - If the instance does not support setting an executor for security reasons (e.g. the necessary permissions are missing).

See Also:
Executor
javax.xml.rpc Class ServiceException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.xml.rpc.ServiceException

All Implemented Interfaces:
  Serializable

public class ServiceException
  extends Exception
Extends: Throwable > Exception

javax.xml.rpc.ServiceException javax.xml.rpc.Service
ServiceFactory

version 1.0

The javax.xml.rpc.ServiceException is thrown from the methods in the
javax.xml.rpc.Service interface and ServiceFactory class.

Version:
  1.0
Author:
  Rahul Sharma
See Also:
  Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServiceException()</td>
</tr>
<tr>
<td>Constructs a new exception with null as its detail message.</td>
</tr>
<tr>
<td>ServiceException(String message)</td>
</tr>
</tbody>
</table>

Constructs a new exception with the specified detail message.

```java
ServiceException(String message, Throwable cause)
```

Constructs a new exception with the specified detail message and cause.

```java
ServiceException(Throwable cause)
```

Constructs a new exception with the specified cause and a detail message of (cause==null ?

---

### Method Summary

| Throwable | getLinkedCause() | Gets the Linked cause |

---

### Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getcause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

---

### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

---

### Constructor Detail

```java
public ServiceException()

nullcause
```

ServiceException

```java
public ServiceException()

Constructs a new exception with null as its detail message. The cause is not initialized.
```
public ServiceException(String message)  
cause

message  
getMessage

ServiceException

public ServiceException(String message)  

Constructs a new exception with the specified detail message. The cause is not initialized.

Parameters:
message - The detail message which is later retrieved using the getMessage method

public ServiceException(String message, Throwable cause)  
cause

message  
getMessage
cause  
getCause

ServiceException

public ServiceException(String message, Throwable cause)  

Constructs a new exception with the specified detail message and cause.

Parameters:
message - The detail message which is later retrieved using the getMessage method
cause - The cause which is saved for the later retrieval throw by the getCause method
public ServiceException(Throwable cause)

JAXRPCException cause

(cause
:cause.toString() cause

cause cause getCause null cause

ServiceException

public ServiceException(Throwable cause)

    Constructs a new exception with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause).

    Parameters:
    cause - The cause which is saved for the later retrieval throw by the getCause method. (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

Method Detail

public Throwable getLinkedCause()

    Gets the Linked cause

    Returns:
    The cause of this Exception or null if the cause is nonexistent or unknown
javax.xml.rpc Class ServiceFactory

java.lang.Object
   \ javax.xml.rpc.ServiceFactory

public abstract class ServiceFactory
extends Object

javax.xml.rpc.ServiceFactory javax.xml.rpc.Service
J2SE Service Service

ServiceFactory SERVICEFACTORY_PROPERTY
   version 1.1
   See also javax.xml.rpc.Service

The javax.xml.rpc.ServiceFactory is an abstract class that provides a factory for the creation of instances of the type javax.xml.rpc.Service. This abstract class follows the abstract static factory design pattern. This enables a J2SE based client to create a Service instance in a portable manner without using the constructor of the Service implementation class.

The ServiceFactory implementation class is set using the system property SERVICEFACTORY_PROPERTY.

Version: 1.1
Author: Rahul Sharma, Roberto Chinnici
See Also: Service
### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>SERVICEFACTORY_PROPERTY</td>
<td>A constant representing the property used to lookup the name of a ServiceFactory implementation class.</td>
</tr>
</tbody>
</table>

### Constructor Summary

<table>
<thead>
<tr>
<th>Access</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected</td>
<td>ServiceFactory()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract</td>
<td>Service createService(QName serviceName)</td>
<td>Create a Service instance.</td>
</tr>
<tr>
<td>abstract</td>
<td>Service createService(URL wsdlDocumentLocation, QName serviceName)</td>
<td>Create a Service instance.</td>
</tr>
<tr>
<td>abstract</td>
<td>Service loadService(Class serviceInterface)</td>
<td>Create an instance of the generated service implementation class for a given service interface, if available.</td>
</tr>
<tr>
<td>abstract</td>
<td>Service loadService(URL wsdlDocumentLocation, Class serviceInterface, Properties properties)</td>
<td>Create an instance of the generated service implementation class for a given service interface, if available.</td>
</tr>
<tr>
<td>abstract</td>
<td>Service loadService(URL wsdlDocumentLocation, QName serviceName, Properties properties)</td>
<td>Create an instance of the generated service implementation class for a given service, if available.</td>
</tr>
<tr>
<td>static</td>
<td>ServiceFactory newInstance()</td>
<td>Gets an instance of the ServiceFactory Only one copy of a factory exists and is returned to the application each time this method is called.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

SERVICEFACTORYPROPERTY

public static final String SERVICEFACTORY_PROPERTY

A constant representing the property used to lookup the name of a ServiceFactory implementation class.

See Also:
Constant Field Values

Constructor Detail

protected ServiceFactory()

ServiceFactory

protected ServiceFactory()

Method Detail

public static ServiceFactory newInstance() throws ServiceException

ServiceFactory
javax.xml.rpc.ServiceFactory

newInstance

public static ServiceFactory newInstance() throws ServiceException

Gets an instance of the ServiceFactory

Only one copy of a factory exists and is returned to the application each time this method is called.

The implementation class to be used can be overridden by setting the javax.xml.rpc.ServiceFactory system property.

Throws: ServiceException

abstract public Service createService(java.net.URL wsdlDocumentLocation, javax.xml.namespace.QName serviceName) throws ServiceException

Create a Service instance.

Parameters:
wsdlDocumentLocation - URL for the WSDL document location for the service
serviceName - QName for the service

Throws:
ServiceException - If any error in creation of the specified service

abstract public Service createService(javax.xml.namespace.QName serviceName) throws ServiceException

Service

serviceName - QName

Throws
ServiceException:

createService

public abstract Service createService(QName serviceName) throws ServiceException

Create a Service instance.

Parameters:
serviceName - QName for the service

Throws:
ServiceException - If any error in creation of the specified service

abstract public Service loadService(Class<T> serviceInterface) throws ServiceException

serviceInterface

Throws
ServiceException:
**loadService**

```java
public abstract Service loadService(Class serviceInterface)
    throws ServiceException
```

Create an instance of the generated service implementation class for a given service interface, if available.

**Parameters:**
- `serviceInterface` - Service interface

**Throws:**
- `ServiceException` - If there is any error while creating the specified service, including the case where a generated service implementation class cannot be located

```java
abstract public Service loadService(java.net.URL wsdlDocumentLocation, Class<T> serviceInterface, java.util.Properties properties)
    throws ServiceException
```

**ServiceException**
- `wsdlDocumentLocation` - WSDL URL
- `serviceInterface`
- `properties`

**Throws**
- `ServiceException`:

**loadService**

```java
public abstract Service loadService(URL wsdlDocumentLocation, Class serviceInterface, Properties properties)
    throws ServiceException
```

Create an instance of the generated service implementation class for a given service interface, if available. An implementation may use the provided `wsdlDocumentLocation` and `properties` to help locate the generated implementation class. If no such class is present, a `ServiceException` will be thrown.
Parameters:

- `wsdlDocumentLocation` - URL for the WSDL document location for the service or null
- `serviceInterface` - Service interface
- `properties` - A set of implementation-specific properties to help locate the generated service implementation class

Throws:

- `ServiceException` - If there is any error while creating the specified service, including the case where a generated service implementation class cannot be located

abstract public `Service` `loadService`(java.net.URL `wsdlDocumentLocation`, javax.xml.namespace.QName `serviceName`, java.util.Properties `properties`) throws `ServiceException`

Create an instance of the generated service implementation class for a given service, if available. The service is uniquely identified by the `wsdlDocumentLocation` and `serviceName` arguments. An implementation may use the provided `properties` to help locate the generated implementation class. If no such class is present, a `ServiceException` will be thrown.
**Parameters:**

- `wsdlDocumentLocation` - URL for the WSDL document location for the service or null
- `serviceName` - Qualified name for the service
- `properties` - A set of implementation-specific properties to help locate the generated service implementation class

**Throws:**

- `ServiceException` - If there is any error while creating the specified service, including the case where a generated service implementation class cannot be located
public interface ServiceLifecycle

void destroy()
JAX-RPC runtime system ends the lifecycle of a service endpoint instance by invoking the destroy method.

void init(Object context)
Used for initialization of a service endpoint.

public void init(Object context) throws ServiceException
JAX-RPC
The init method implementation should typecast the context parameter to an appropriate Java type. For service endpoints deployed on a servlet container based JAX-RPC runtime system, the context parameter is of the Java type `javax.xml.rpc.server.ServletEndpointContext`. The `ServletEndpointContext` provides an endpoint context maintained by the underlying servlet container based JAX-RPC runtime system.

**Parameters:**
- `context` - Endpoint context for a JAX-RPC service endpoint

**Throws:**
- `ServiceException` - If any error in initialization of the service endpoint; or if any illegal context has been provided in the init
public void destroy()
JAX-RPC destroy destroy

destroy

void destroy()

JAX-RPC runtime system ends the lifecycle of a service endpoint instance by invoking the destroy method. The service endpoint releases its resources in the implementation of the destroy method.
javax.xml.ws Annotation Type ServiceMode

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Inherited  
@Documented

public @interface ServiceMode

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Inherited  
@Documented

Provider

**since** JAX-WS 2.0

Used to indicate whether a Provider implementation wishes to work with entire protocol messages or just with protocol message payloads.

**Since:**  
JAX-WS 2.0

---

## Optional Element Summary

<table>
<thead>
<tr>
<th>Service.Mode</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service mode.</td>
<td></td>
</tr>
</tbody>
</table>

abstract public **Service.Mode** value()  
PAYLOAD  Provider MESSAGE  Provider

value
public abstract Service.Mode value

Service mode. PAYLOAD indicates that the Provider implementation wishes to work with protocol message payloads only. MESSAGE indicates that the Provider implementation wishes to work with entire protocol messages.

Default:
PAYLOAD

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.servlet Interface Servlet**

**All Known Subinterfaces:**
- [HttpJspPage](#), [JspPage](#)

**All Known Implementing Classes:**
- [FacesServlet](#), [GenericServlet](#), [HttpServlet](#)

```java
public interface Servlet

Implemented by: FacesServlet, GenericServlet, JspPage
```

A servlet is a small Java program that runs within a Web server. Servlets receive and respond to requests from Web clients, usually across HTTP, the HyperText Transfer Protocol.
To implement this interface, you can write a generic servlet that extends javax.servlet.GenericServlet or an HTTP servlet that extends javax.servlet.http.HttpServlet.

This interface defines methods to initialize a servlet, to service requests, and to remove a servlet from the server. These are known as life-cycle methods and are called in the following sequence:

1. The servlet is constructed, then initialized with the init method.
2. Any calls from clients to the service method are handled.
3. The servlet is taken out of service, then destroyed with the destroy method, then garbage collected and finalized.

In addition to the life-cycle methods, this interface provides the `getServletConfig` method, which the servlet can use to get any startup information, and the `getServletInfo` method, which allows the servlet to return basic information about itself, such as author, version, and copyright.

**Author:** Various

**See Also:** GenericServlet, HttpServlet

---

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><code>destroy()</code></td>
<td>Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.</td>
</tr>
<tr>
<td></td>
<td><code>getServletConfig()</code></td>
<td>Returns a <code>ServletConfig</code> object, which contains initialization and startup parameters for this servlet.</td>
</tr>
<tr>
<td></td>
<td><code>getServletInfo()</code></td>
<td>Returns information about the servlet, such as author, version, and copyright.</td>
</tr>
<tr>
<td></td>
<td><code>init(ServletConfig config)</code></td>
<td>Called by the servlet container to indicate to a servlet</td>
</tr>
</tbody>
</table>
that the servlet is being placed into service.

```java
public void init(ServletConfig config) throws ServletException
```

Called by the servlet container to indicate to a servlet that the servlet is being placed into service.

The servlet container calls the `init` method exactly once after instantiating the servlet. The `init` method must complete successfully before the servlet can receive any requests.
The servlet container cannot place the servlet into service if the init method

1. Throws a ServletException
2. Does not return within a time period defined by the Web server

Parameters:
   config - a ServletConfig object containing the servlet's configuration and initialization parameters

Throws:
   ServletException - if an exception has occurred that interferes with the servlet's normal operation

See Also:
   UnavailableException, getServletConfig()

---

public <strong>ServletConfig</strong> getServletConfig()

Returns a ServletConfig object, which contains initialization and startup parameters for this servlet. The ServletConfig object returned is the one passed to the init method.

Implementations of this interface are responsible for storing the ServletConfig object so that this method can return it. The GenericServlet class, which implements this interface, already does this.
Returns:
the ServletConfig object that initializes this servlet

See Also:
init(javax.servlet.ServletConfig)

```java
public void service(ServletRequest req, ServletResponse res) throws ServletException, java.io.IOException

servlet  init()

servlet

servlet  servlet  Java
the Java tutorial on multi-threaded programming

req ServletRequest
res servlet

Throws ServletException: servlet

Throws java.io.IOException:

service

void service(ServletRequest req,
ServletResponse res)
throws ServletException,
IOException

Called by the servlet container to allow the servlet to respond to a request.

This method is only called after the servlet's init() method has completed successfully.

The status code of the response always should be set for a servlet that throws or sends an error.
Servlets typically run inside multithreaded servlet containers that can handle multiple requests concurrently. Developers must be aware to synchronize access to any shared resources such as files, network connections, and as well as the servlet's class and instance variables. More information on multithreaded programming in Java is available in the Java tutorial on multi-threaded programming.

**Parameters:**
- req - the ServletRequest object that contains the client's request
- res - the ServletResponse object that contains the servlet's response

**Throws:**
- ServletException - if an exception occurs that interferes with the servlet's normal operation
- IOException - if an input or output exception occurs

```java
public String getServletInfo()
{HtmlXML
    return servlet String
}
```

**getServletInfo**

```java
String getServletInfo()

    Returns information about the servlet, such as author, version, and copyright.

    The string that this method returns should be plain text and not markup of any kind (such as HTML, XML, etc.).

    **Returns:**
    a String containing servlet information
```
public void destroy()

Called by the servlet container to indicate to a servlet that the servlet is being taken out of service. This method is only called once all threads within the servlet's service method have exited or after a timeout period has passed. After the servlet container calls this method, it will not call the service method again on this servlet.

This method gives the servlet an opportunity to clean up any resources that are being held (for example, memory, file handles, threads) and make sure that any persistent state is synchronized with the servlet's current state in memory.
javax.servlet Interface ServletConfig

All Known Implementing Classes: GenericServlet, HttpServlet

public interface ServletConfig

Implemented by: GenericServlet

A servlet configuration object used by a servlet container to pass information to a servlet during initialization.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getInitParameter(String name)</td>
<td>Returns a string containing the value of the named initialization parameter, or null if the parameter does not exist.</td>
</tr>
<tr>
<td>Enumeration getInitParameterNames()</td>
<td>Returns the names of the servlet's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the servlet has no initialization parameters.</td>
</tr>
<tr>
<td>ServletContext getServletContext()</td>
<td>Returns a reference to the ServletContext in which the caller is executing.</td>
</tr>
<tr>
<td>String getServletName()</td>
<td>Returns the name of this servlet instance.</td>
</tr>
</tbody>
</table>

Method Detail
public String getServletName()

getServletName

String getServletName()

Returns the name of this servlet instance. The name may be provided via server administration, assigned in the web application deployment descriptor, or for an unregistered (and thus unnamed) servlet instance it will be the servlet's class name.

Returns: the name of the servlet instance

---

public ServletContext getServletContext()

getServletContext

ServletContext getServletContext()

Returns a reference to the ServletContext in which the caller is executing.

Returns: a ServletContext object, used by the caller to interact with its servlet container

See Also: ServletContext
public String getInitParameter(String name)

    String null

    name null
    return String

getInitParameter

String getInitParameter(String name)

    Returns a String containing the value of the named initialization
    parameter, or null if the parameter does not exist.

Parameters:
    name - a String specifying the name of the initialization
    parameter

Returns:
    a String containing the value of the initialization parameter

public java.util.Enumeration<E> getInitParameterNames()

    Enumeration servlet servlet
    Enumeration return servlet String Enumeration

getInitParameterNames

Enumeration getInitParameterNames()

    Returns the names of the servlet's initialization parameters as an
    Enumeration of String objects, or an empty Enumeration if the servlet
    has no initialization parameters.

Returns:
    an Enumeration of String objects containing the names of the
    servlet's initialization parameters
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.servlet Interface ServletContext

public interface ServletContext

servlet servlet MIME

Java "Web ""Web " servlet Collection servlet
URL /catalog .war

Web "distributed"

ServletContext ServletConfig ServletConfig servlet
Web servlet
See also getServletConfig, getServletContext

Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.

There is one context per "web application" per Java Virtual Machine. (A "web application" is a collection of servlets and content installed under a specific subset of the server's URL namespace such as /catalog and possibly installed via a .war file.)

In the case of a web application marked "distributed" in its deployment descriptor, there will be one context instance for each virtual machine. In this situation, the context cannot be used as a location to share global information (because the information won't be truly global). Use an external resource like a database instead.

The ServletContext object is contained within the ServletConfig object, which the Web server provides the servlet when the servlet is initialized.

Author:
### Method Summary

<table>
<thead>
<tr>
<th>Method Type</th>
<th>Get Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td><code>getAttribute(String name)</code></td>
<td>Returns the servlet container attribute with the given name, or <code>null</code> if there is no attribute by that name.</td>
</tr>
<tr>
<td>Enumeration</td>
<td><code>getAttributeNames()</code></td>
<td>Returns an <code>Enumeration</code> containing the attribute names available within this servlet context.</td>
</tr>
<tr>
<td>ServletContext</td>
<td><code>getContext(String uripath)</code></td>
<td>Returns a <code>ServletContext</code> object that corresponds to a specified URL on the server.</td>
</tr>
<tr>
<td>String</td>
<td><code>getContextPath()</code></td>
<td>Returns the context path of the web application.</td>
</tr>
<tr>
<td>String</td>
<td><code>getInitParameter(String name)</code></td>
<td>Returns a string containing the value of the named context-wide initialization parameter, or <code>null</code> if the parameter does not exist.</td>
</tr>
<tr>
<td>Enumeration</td>
<td><code>getInitParameterNames()</code></td>
<td>Returns the names of the context's initialization parameters as an <code>Enumeration</code> of <code>String</code> objects, or an empty <code>Enumeration</code> if the context has no initialization parameters.</td>
</tr>
<tr>
<td>int</td>
<td><code>getMajorVersion()</code></td>
<td>Returns the major version of the Java Servlet API that this servlet container supports.</td>
</tr>
<tr>
<td>String</td>
<td><code>getMimeType(String file)</code></td>
<td>Returns the MIME type of the specified file, or <code>null</code> if the MIME type is not known.</td>
</tr>
<tr>
<td>int</td>
<td><code>getMinorVersion()</code></td>
<td>Returns the minor version of the Servlet API that this servlet container supports.</td>
</tr>
</tbody>
</table>

See Also:  
`Servlet.getServletConfig()`, `ServletConfig.getServletContext()`
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getNamedDispatcher(String name)</code></td>
<td>Returns a <code>RequestDispatcher</code> object that acts as a wrapper for the named servlet.</td>
</tr>
<tr>
<td><code>getRealPath(String path)</code></td>
<td>Returns a <code>String</code> containing the real path for a given virtual path.</td>
</tr>
<tr>
<td><code>getRequestDispatcher(String path)</code></td>
<td>Returns a <code>RequestDispatcher</code> object that acts as a wrapper for the resource located at the given path.</td>
</tr>
<tr>
<td><code>getResource(String path)</code></td>
<td>Returns a URL to the resource that is mapped to a specified path.</td>
</tr>
<tr>
<td><code>getResourceAsStream(String path)</code></td>
<td>Returns the resource located at the named path as an <code>InputStream</code> object.</td>
</tr>
<tr>
<td><code>getResourcePaths(String path)</code></td>
<td>Returns a directory-like listing of all the paths to resources within the web application whose longest sub-path matches the supplied path argument.</td>
</tr>
<tr>
<td><code>getServerInfo()</code></td>
<td>Returns the name and version of the servlet container on which the servlet is running.</td>
</tr>
<tr>
<td><code>getServlet(String name)</code></td>
<td><strong>Deprecated.</strong> <em>As of Java Servlet API 2.1, with no direct replacement.</em></td>
</tr>
</tbody>
</table>

This method was originally defined to retrieve a servlet from a `ServletContext`. In this version, this method always returns `null` and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.

In lieu of this method, servlets can share information using the `ServletContext` class and can perform shared business logic by invoking methods on common non-servlet classes.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getServletContextName()</strong></td>
<td>Returns the name of this web application corresponding to this ServletContext as specified in the deployment descriptor for this web application by the display-name element.</td>
</tr>
<tr>
<td><strong>getServletNames()</strong></td>
<td><strong>Deprecated. As of Java Servlet API 2.1, with no replacement.</strong></td>
</tr>
<tr>
<td></td>
<td>This method was originally defined to return an Enumeration of all the servlet names known to this context. In this version, this method always returns an empty Enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</td>
</tr>
<tr>
<td><strong>getServlets()</strong></td>
<td><strong>Deprecated. As of Java Servlet API 2.0, with no replacement.</strong></td>
</tr>
<tr>
<td></td>
<td>This method was originally defined to return an Enumeration of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</td>
</tr>
<tr>
<td><strong>log(Exception exception, String msg)</strong></td>
<td><strong>Deprecated. As of Java Servlet API 2.1, use log(String message, Throwable throwable) instead.</strong></td>
</tr>
<tr>
<td></td>
<td>This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.</td>
</tr>
<tr>
<td><strong>log(String msg)</strong></td>
<td>Writes the specified message to a servlet log file, usually an event log.</td>
</tr>
</tbody>
</table>
**Method Detail**

**public String getContextPath()**

Web

URI "/" "/" servlet ""

```
getContextPath() since Servlet 2.5
See also  getApplicationContext()
```

### getApplicationContext

```
String getApplicationContext()
```

Returns the context path of the web application.

The context path is the portion of the request URI that is used to select the context of the request. The context path always comes first in a request URI. The path starts with a "/" character but does
not end with a "/" character. For servlets in the default (root) context, this method returns "."

It is possible that a servlet container may match a context by more than one context path. In such cases the 
HttpServletRequest.getContextPath() will return the actual context path used by the request and it may differ from the path returned by this method. The context path returned by this method should be considered as the prime or preferred context path of the application.

Returns:
The context path of the web application, or "" for the default (root) context

Since:
Servlet 2.5

See Also:
HttpServletRequest.getContextPath()

public ServletContext getContext(String uripath)

URL ServletContext

servlet RequestDispatcher "/" Web

servlet URL null
uripath Web String
return URL ServletContext null
See also javax.servlet.RequestDispatcher

gectContext

ServletContext getContext(String uripath)

Returns a ServletContext object that corresponds to a specified URL on the server.
This method allows servlets to gain access to the context for various parts of the server, and as needed obtain RequestDispatcher objects from the context. The given path must be begin with "/", is interpreted relative to the server's document root and is matched against the context roots of other web applications hosted on this container.

In a security conscious environment, the servlet container may return null for a given URL.

**Parameters:**

- `uripath` - a `String` specifying the context path of another web application in the container.

**Returns:**

the `ServletContext` object that corresponds to the named URL, or null if either none exists or the container wishes to restrict this access.

**See Also:**

RequestDispatcher

```java
public int getMajorVersion()

servlet Java Servlet API Version 2.5 2

return 2
```

**getMajorVersion**

```java
int getMajorVersion()

    Returns the major version of the Java Servlet API that this servlet container supports. All implementations that comply with Version 2.5 must have this method return the integer 2.

    Returns:
    2
```

```java
public int getMinorVersion()
```
getMinorVersion

int getMinorVersion()

Returns the minor version of the Servlet API that this servlet container supports. All implementations that comply with Version 2.5 must have this method return the integer 5.

Returns:
5

public String getMimeType(String file)

Returns the MIME type of the specified file, or null if the MIME type is not known. The MIME type is determined by the configuration of the servlet container, and may be specified in a web application deployment descriptor. Common MIME types are "text/html" and "image/gif".

Parameters:
file - a String specifying the name of a file

Returns:
a String specifying the file's MIME type
public java.util.Set<E> getResourcePaths(String path)
  Web path '/' Web '/' Web

/welcome.html
/catalog/index.html
/catalog/products.html
/catalog/offers/books.html
/catalog/offers/music.html
/customer/login.jsp
/WEB-INF/web.xml
/WEB-INF/classes/com.acme.OrderServlet.class,

g得到ResourcePaths("/") {"/welcome.html", "/catalog/", "/customer/", "/WEB-INF/"}
  path /
  return Web null
  since Servlet 2.3

得到ResourcePaths

Set  get得到ResourcePaths(String path)

  Returns a directory-like listing of all the paths to resources within the
  web application whose longest sub-path matches the supplied path
  argument. Paths indicating subdirectory paths end with a '/'. The
  returned paths are all relative to the root of the web application and
  have a leading '/'. For example, for a web application containing

  /welcome.html
  /catalog/index.html
  /catalog/products.html
  /catalog/offers/books.html
Parameters:
  path - the partial path used to match the resources, which must start with a /

Returns:
a Set containing the directory listing, or null if there are no resources in the web application whose path begins with the supplied path.

Since:
Servlet 2.3

public java.net.URL getResource(String path) throws java.net.MalformedURLException
URL "/"

servlet servlet .war

servlet URL URLConnection
null

URL URL
.jsp JSP RequestDispatcher

java.lang.Class.getResource
  path String
getResource

```java
URL getResource(String path)
throws MalformedURLException
```

Returns a URL to the resource that is mapped to a specified path. The path must begin with a "/" and is interpreted as relative to the current context root.

This method allows the servlet container to make a resource available to servlets from any source. Resources can be located on a local or remote file system, in a database, or in a .war file.

The servlet container must implement the URL handlers and URLConnection objects that are necessary to access the resource.

This method returns `null` if no resource is mapped to the pathname.

Some containers may allow writing to the URL returned by this method using the methods of the URL class.

The resource content is returned directly, so be aware that requesting a .jsp page returns the JSP source code. Use a RequestDispatcher instead to include results of an execution.

This method has a different purpose than java.lang.Class.getResource, which looks up resources based on a class loader. This method does not use class loaders.

**Parameters:**
- `path` - a String specifying the path to the resource

**Returns:**
the resource located at the named path, or `null` if there is no resource at that path

**Throws:**
**MalformedURLException** - if the pathname is not given in the correct form

```java
public java.io.InputStream getResourceAsStream(String path)
    throws MalformedURLException {
    InputStream resourceAsStream = null;
    URL servletURL = null;
    try {
        servletURL = new URL(path);
        resourceAsStream = servletURL.openStream();
    } catch (MalformedURLException e) {
        // e.printStackTrace();
        throw new MalformedURLException(e.getMessage());
    } catch (IOException e) {
        // e.printStackTrace();
        throw new MalformedURLException(e.getMessage());
    }
    return resourceAsStream;
}
```

getAsResourceAsStream

`java.lang.Class.getResourceAsStream(String path)`

Returns the resource located at the named path as an `InputStream` object.

The data in the `InputStream` can be of any type or length. The path must be specified according to the rules given in `getResource`. This method returns `null` if no resource exists at the specified path.

Meta-information such as content length and content type that is available via `getResource` method is lost when using this method.

The servlet container must implement the URL handlers and `URLConnection` objects necessary to access the resource.
This method is different from java.lang.Class.getResourceAsStream, which uses a class loader. This method allows servlet containers to make a resource available to a servlet from any location, without using a class loader.

**Parameters:**
- path - a String specifying the path to the resource

**Returns:**
- the InputStream returned to the servlet, or null if no resource exists at the specified path

```java
public RequestDispatcher getRequestDispatcher(String path)
```

- Returns a `RequestDispatcher` object that acts as a wrapper for the resource located at the given path. A `RequestDispatcher` object can be used to forward a request to the resource or to include the resource in a response. The resource can be dynamic or static.

The pathname must begin with a "/" and is interpreted as relative to the current context root. Use `getContext` to obtain a
RequestDispatcher for resources in foreign contexts. This method returns null if the ServletContext cannot return a RequestDispatcher.

Parameters:
- path - a String specifying the pathname to the resource

Returns:
a RequestDispatcher object that acts as a wrapper for the resource at the specified path, or null if the ServletContext cannot return a RequestDispatcher

See Also:
- RequestDispatcher, getContext(java.lang.String)

```java
public RequestDispatcher getNamedDispatcher(String name)
```

Servlet JSP  Web servlet

```java
ServletContext name
RequestDispatcher servlet null
return servlet RequestDispatcher ServletContext
See also javax.servlet.RequestDispatcher, getContext, getServletName
```

getNamedDispatcher

```java
RequestDispatcher getNamedDispatcher(String name)
```

Returns a RequestDispatcher object that acts as a wrapper for the named servlet.

Servlets (and JSP pages also) may be given names via server administration or via a web application deployment descriptor. A
servlet instance can determine its name using
`ServletConfig.getServletName()`.

This method returns `null` if the `ServletContext` cannot return a
RequestDispatcher for any reason.

**Parameters:**
- `name` - A `String` specifying the name of a servlet to wrap

**Returns:**
- A `RequestDispatcher` object that acts as a wrapper for the named
  servlet, or `null` if the `ServletContext` cannot return a
  RequestDispatcher

**See Also:**
- `RequestDispatcher`, `getContext(java.lang.String)`,
  `ServletConfig.getServletName()`

```java
public Servlet getServlet(String name) throws ServletException
```

Java Servlet API 2.1

deprecated

```
ServletContext servlet null Java Servlet API
```

servletServletContext servlet

**getServlet**

```java
Servlet getServlet(String name)
```

throws `ServletException`

**Deprecated. As of Java Servlet API 2.1, with no direct replacement.**

This method was originally defined to retrieve a servlet from a
`ServletContext`. In this version, this method always returns `null` and
remains only to preserve binary compatibility. This method will be
permanently removed in a future version of the Java Servlet API.
In lieu of this method, servlets can share information using the ServletContext class and can perform shared business logic by invoking methods on common non-servlet classes.

**Throws:**

`ServletException`

---

**public java.util Enumeration&lt;E&gt; getServlets()**

Java Servlet API 2.0

*deprecated*  
servlet  
Enumeration  
Java Servlet API

---

**getServlets**

`Enumeration getServlets()`

**Deprecated. As of Java Servlet API 2.0, with no replacement.**

This method was originally defined to return an Enumeration of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.

---

**public java.util Enumeration&lt;E&gt; getServletNames()**

Java Servlet API 2.1

*deprecated*  
servlet  
Enumeration  
Enumeration  
Java Servlet API

---

**getServletNames**

`Enumeration getServletNames()`
**Deprecated.** As of Java Servlet API 2.1, with no replacement.

This method was originally defined to return an `Enumeration` of all the servlet names known to this context. In this version, this method always returns an empty `Enumeration` and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.

---

```java
public void log(String msg)
servelet  servlet  servlet

msg  String
```

### log

```java
void log(String msg)
```

Writes the specified message to a servlet log file, usually an event log. The name and type of the servlet log file is specific to the servlet container.

**Parameters:**

- `msg` - a `String` specifying the message to be written to the log file.

---

```java
public void log(Exception exception, String msg)
```

Java Servlet API 2.1  
#log(String message, Throwable throwable)

```java
deprecated
```

```java
servelet
```

### log

```java
void log(Exception exception, String msg)
```
**Deprecated.** As of Java Servlet API 2.1, use `log(String message, Throwable throwable)` instead.

This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.

```java
public void log(String message, Throwable throwable)

message
throwable

log

void log(String message,
 Throwable throwable)

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file. The name and type of the servlet log file is specific to the servlet container, usually an event log.

**Parameters:**
message - a String that describes the error or exception
throwable - the Throwable error or exception
```

```java
public String getRealPath(String path)

"http://host/contextPath/index.html" contextPath

getRealPath

public String getRealPath(String path) String "http://host/contextPath/index.html" contextPath ServletContext

path return .war null

String null
```

```java
public String getRealPath(String path)

String "http://host/contextPath/index.html" contextPath ServletContext

getRealPath

public String getRealPath(String path) String "http://host/contextPath/index.html" contextPath ServletContext

path return .war null

String null
```
**getRealPath**

`String getRealPath(String path)`

Returns a `String` containing the real path for a given virtual path. For example, the path "/index.html" returns the absolute file path on the server's filesystem would be served by a request for "http://host/contextPath/index.html", where contextPath is the context path of this ServletContext..

The real path returned will be in a form appropriate to the computer and operating system on which the servlet container is running, including the proper path separators. This method returns `null` if the servlet container cannot translate the virtual path to a real path for any reason (such as when the content is being made available from a .war archive).

**Parameters:**
- `path` - a `String` specifying a virtual path

**Returns:**
- a `String` specifying the real path, or `null` if the translation cannot be performed

---

**public String getServerInfo()**

`servlet` servlet

`servername/versionnumber`JavaServer Web Development Kit

`JavaServer Web Dev Kit/1.0`

`servlet` JavaServer Web Dev Kit/1.0 (JDK 1.1.6; Windows NT 4.0 x86)

return servlet String

**getServerInfo**
**String getServerInfo()**

Returns the name and version of the servlet container on which the servlet is running.

The form of the returned string is `servername/versionnumber`. For example, the JavaServer Web Development Kit may return the string `JavaServer Web Dev Kit/1.0`.

The servlet container may return other optional information after the primary string in parentheses, for example, `JavaServer Web Dev Kit/1.0 (JDK 1.1.6; Windows NT 4.0 x86)`.

**Returns:**

- a String containing at least the servlet container name and version number

---

**public String getInitParameter(String name)**

```
String null
```

**“Web ” webmaster**

```
name servlet
return String
```

See also **getInitParameter**

---

**getInitParameter**

```
String getInitParameter(String name)
```

Returns a String containing the value of the named context-wide initialization parameter, or null if the parameter does not exist.

This method can make available configuration information useful to an entire "web application". For example, it can provide a webmaster's email address or the name of a system that holds critical data.
Parameters:
name - a String containing the name of the parameter whose value is requested

Returns:
a String containing at least the servlet container name and version number

See Also:
ServletConfig.getInitParameter(java.lang.String)

---

public java.util.Enumeration&lt;E&gt; getInitParameterNames()

String Enumeration Enumeration
return String Enumeration
See also getInitParameter

getInitParameterNames

Enumeration getInitParameterNames()

Returns the names of the context's initialization parameters as an Enumeration of String objects, or an empty Enumeration if the context has no initialization parameters.

Returns:
an Enumeration of String objects containing the names of the context's initialization parameters

See Also:
ServletConfig.getInitParameter(java.lang.String)

---

public Object getAttribute(String name)
servlet
getAttributeNames

java.lang.object Java Servlet API
java.*javax.* sun.*

name String
**getAttribute**

```java
Object getAttribute(String name)
```

Returns the servlet container attribute with the given name, or `null` if there is no attribute by that name. An attribute allows a servlet container to give the servlet additional information not already provided by this interface. See your server documentation for information about its attributes. A list of supported attributes can be retrieved using `getAttributeNames`.

The attribute is returned as a `java.lang.Object` or some subclass. Attribute names should follow the same convention as package names. The Java Servlet API specification reserves names matching `java.*`, `javax.*`, and `sun.*`.

**Parameters:**
- `name` - a `String` specifying the name of the attribute

**Returns:**
- an `Object` containing the value of the attribute, or `null` if no attribute exists matching the given name

**See Also:**
- `getAttributeNames()`

---

**public java.util.Enumeration<E> getAttributeNames()**

```java
Enumeration Enumeration #getAttributes
```

Returns an `Enumeration` containing the names of all attributes that are recognized by the container. The enumeration can be cast to `java.util.Enumeration` and used to retrieve the corresponding values using `getAttribute`. The order of the enumation is unspecified.

**See Also:**
- `getAttribute`
Enumeration getAttributeNames()

Returns an Enumeration containing the attribute names available within this servlet context. Use the getAttribute(java.lang.String) method with an attribute name to get the value of an attribute.

Returns:
an Enumeration of attribute names
See Also:
getAttribute(java.lang.String)

public void setAttribute(String name, Object object)

Binds an object to a given attribute name in this servlet context. If the name specified is already used for an attribute, this method will replace the attribute with the new to the new attribute.

If listeners are configured on the ServletContext the container notifies them accordingly.

If a null value is passed, the effect is the same as calling removeAttribute().
Attribute names should follow the same convention as package names. The Java Servlet API specification reserves names matching java.*, javax.*, and sun.*.

Parameters:
- name - a String specifying the name of the attribute
- object - an Object representing the attribute to be bound

```java
public void removeAttribute(String name)
```

Removes the attribute with the given name from the servlet context. After removal, subsequent calls to `getAttribute(java.lang.String)` to retrieve the attribute's value will return `null`.

If listeners are configured on the `ServletContext` the container notifies them accordingly.

Parameters:
- name - a String specifying the name of the attribute to be removed

```java
public String getServletContextName()
```

Returns the name of the Web display-name. Since Servlet 2.3
getServletContextName

String getServletContextName()

Returns the name of this web application corresponding to this ServletContext as specified in the deployment descriptor for this web application by the display-name element.

Returns:
The name of the web application or null if no name has been declared in the deployment descriptor.

Since:
Servlet 2.3
public class ServletContextAttributeEvent

extends ServletContextEvent

Extends: java.util.EventObject > ServletContextEvent

Web servlet since v 2.3

See also javax.servlet.ServletContextAttributeListener

This is the event class for notifications about changes to the attributes of the servlet context of a web application.

Since:

v 2.3

See Also:

ServletContextAttributeListener, Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
source
Constructor Summary

**ServletContextAttributeEvent** *(ServletContext source, String name, Object value)*

Construct a ServletContextAttributeEvent from the given context for the given attribute name and attribute value.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getName()</strong></td>
<td>Return the name of the attribute that changed on the ServletContext.</td>
</tr>
<tr>
<td><strong>getValue()</strong></td>
<td>Returns the value of the attribute that has been added, removed, or replaced.</td>
</tr>
</tbody>
</table>

Methods inherited from class [*javax.servlet.ServletContextEvent*](#)

**getServletContext**

Methods inherited from class [*java.util.EventObject*](#)

**getSource**, **toString**

Methods inherited from class [*java.lang.Object*](#)


Constructor Detail

public ServletContextAttributeEvent(*ServletContext source, String name, Object value*)

ServletContextAttributeEvent

ServletContextAttributeEvent
Construct a ServletContextAttributeEvent from the given context for the given attribute name and attribute value.

**Method Detail**

**public String getName()**

**getVersion**

public String getName()

Return the name of the attribute that changed on the ServletContext.

**public Object getValue()**

**getValue**

public Object getValue()

Returns the value of the attribute that has been added, removed, or replaced. If the attribute was added, this is the value of the attribute. If the attribute was removed, this is the value of the removed attribute. If the attribute was replaced, this is the old value of the attribute.
javax.servlet  Interface ServletContextAttributeListener

All Superinterfaces:
EventListener

public interface ServletContextAttributeListener
extends EventListener

Implements: java.util.EventListener

Web servlet Web
since v 2.3
See also javax.servlet.ServletContextAttributeEvent

Implementations of this interface receive notifications of changes to the attribute list on the servlet context of a web application. To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

Since:
v 2.3
See Also:
ServletContextAttributeEvent

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void attributeAdded(ServletContextAttributeEvent scab)</td>
</tr>
<tr>
<td>Notification that a new attribute was added to the servlet context.</td>
</tr>
<tr>
<td>void attributeRemoved(ServletContextAttributeEvent scab)</td>
</tr>
<tr>
<td>Notification that an existing attribute has been removed from the servlet context.</td>
</tr>
<tr>
<td>void attributeReplaced(ServletContextAttributeEvent scab)</td>
</tr>
</tbody>
</table>
void Notification that an attribute on the servlet context has been replaced.

**Method Detail**

public void attributeAdded(ServletContextAttributeEvent scab)  
servlet

attributeAdded

void attributeAdded(ServletContextAttributeEvent scab)

Notification that a new attribute was added to the servlet context. Called after the attribute is added.

---

public void attributeRemoved(ServletContextAttributeEvent scab)  
servlet

attributeRemoved

void attributeRemoved(ServletContextAttributeEvent scab)

Notification that an existing attribute has been removed from the servlet context. Called after the attribute is removed.

---

public void attributeReplaced(ServletContextAttributeEvent scab)  
servlet
attributeReplaced

void attributeReplaced(ServletContextAttributeEvent scab)

Notification that an attribute on the servlet context has been replaced. Called after the attribute is replaced.
javax.servlet Class ServletContextEvent

java.lang.Object
   ↓ java.util.EventObject
      ↓ javax.servlet.ServletContextEvent

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   ServletContextAttributeEvent

public class ServletContextEvent
   extends EventObject

Extends: java.util.EventObject
Extended by: ServletContextAttributeEvent

Web servlet
   since v 2.3
See also javax.servlet.ServletContextListener

This is the event class for notifications about changes to the servlet context of a web application.

Since:
   v 2.3
See Also:
   ServletContextListener, Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
### Constructor Summary

| `ServletContextEvent(ServletContext source)` | Construct a `ServletContextEvent` from the given context. |

### Method Summary

| `getServletContext()` | Return the `ServletContext` that changed. |

Methods inherited from class `java.util.EventObject`
- `getSource`, `toString`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public ServletContext getServletContext()

ServletContext return ServletContext

getServletContext

public ServletContext getServletContext()

Return the ServletContext that changed.

Returns:
the ServletContext that sent the event.
javax.servlet Interface ServletContextListener

All Superinterfaces:
   EventListener

public interface ServletContextListener
   extends EventListener

Implements: java.util.EventListener

Implements of this interface receive notifications about changes to the servlet context of the web application they are part of. To receive notification events, the implementation class must be configured in the deployment descriptor for the web application.

Since:
   v 2.3

See Also:
   ServletContextEvent

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void  contextDestroyed(ServletContextEvent sce)</td>
</tr>
<tr>
<td>Notification that the servlet context is about to be shut down.</td>
</tr>
<tr>
<td>void  contextInitialized(ServletContextEvent sce)</td>
</tr>
<tr>
<td>Notification that the web application initialization process is starting.</td>
</tr>
</tbody>
</table>
public void contextInitialized(ServletContextEvent sce)
Web Web servlet
ServletContextListener

calendarInitialized

void contextInitialized(ServletContextEvent sce)

Notification that the web application initialization process is starting. All ServletContextListeners are notified of context initialization before any filter or servlet in the web application is initialized.

public void contextDestroyed(ServletContextEvent sce)
servlet ServletContextListener servlet

calendarDestroyed

void contextDestroyed(ServletContextEvent sce)

Notification that the servlet context is about to be shut down. All servlets and filters have been destroy()ed before any ServletContextListeners are notified of context destruction.
PS:
javax.xml.rpc.server Interface ServletEndpointContext

public interface ServletEndpointContext

ServletEndpointContext servlet JAX-RPC servlet
JAX-RPC ServiceLifecycle.init context Java
javax.xml.rpc.server.ServletEndpointContext
	servlet
	servlet JAX-RPC ServletEndpointContext JAX-RPC

version 1.1

The ServletEndpointContext provides an endpoint context maintained by
the underlying servlet container based JAX-RPC runtime system. For
service endpoints deployed on a servlet container based JAX-RPC
runtime system, the context parameter in the ServiceLifecycle.init
method is required to be of the Java type
javax.xml.rpc.server.ServletEndpointContext.

A servlet container based JAX-RPC runtime system implements the
ServletEndpointContext interface. The JAX-RPC runtime system is
required to provide appropriate session, message context, servlet context
and user principal information per method invocation on the endpoint
class.

Version: 1.1
Author: Rahul Sharma, Roberto Chinnici

Method Summary

`getHttpSession()`
<table>
<thead>
<tr>
<th>HttpSession</th>
<th>The <code>getHttpSession</code> method returns the current HTTP session (as a <code>javax.servlet.http.HttpSession</code>).</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageContext</td>
<td>The method <code>getMessageContext</code> returns the MessageContext targeted for this endpoint instance.</td>
</tr>
<tr>
<td>ServletContext</td>
<td>The method <code>getServletContext</code> returns the ServletContext associated with the web application that contain this endpoint.</td>
</tr>
<tr>
<td>Principal</td>
<td>Returns a <code>java.security.Principal</code> instance that contains the name of the authenticated user for the current method invocation on the endpoint instance.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>isUserInRole(String role)</code> Returns a boolean indicating whether the authenticated user for the current method invocation on the endpoint instance is included in the specified logical &quot;role&quot;.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public MessageContext getMessageContext()
```

**Handler**

- `getMessageContext`  
- `MessageContext`  
- `HandlerChain`

**HandlerChain**

- `SOAP`

**return**

- `MessageContext`

**Throws**

- `NoSuchElementException`

**See also**

- `javax.xml.rpc.handler.MessageContext`,  
- `javax.xml.rpc.handler.HandlerChain`,  
- `javax.xml.rpc.handler.Handler`

**getMessageContext**

- `MessageContext`  
- `getMessageContext()`
The method `getMessageContext` returns the `MessageContext` targeted for this endpoint instance. This enables the service endpoint instance to access the `MessageContext` propagated by request `HandlerChain` (and its contained `Handler` instances) to the target endpoint instance and to share any SOAP message processing related context. The endpoint instance can access and manipulate the `MessageContext` and share the SOAP message processing related context with the response `HandlerChain`.

**Returns:**

- `MessageContext`; If there is no associated `MessageContext`, this method returns `null`.

**Throws:**

- `IllegalStateException` - if this method is invoked outside a remote method implementation by a service endpoint instance.

**See Also:**

- `MessageContext`, `HandlerChain`, `Handler`

```java
public java.security.Principal getUserPrincipal()
```

Returns a `java.security.Principal` instance that contains the name of the authenticated user for the current method invocation on the endpoint instance. This method returns `null` if there is no associated principal yet. The underlying JAX-RPC runtime system takes the responsibility of providing the appropriate authenticated principal for a remote method invocation on the service endpoint instance.

**Returns:**

- A `java.security.Principal` for the authenticated principal
associated with the current invocation on the servlet endpoint instance; Returns null if there no authenticated user associated with a method invocation.

See Also:
Principal

```java
public HttpSession getHttpSession()
```

The `getHttpSession` method returns the current HTTP session (as a `javax.servlet.http.HttpSession`). When invoked by the service endpoint within a remote method implementation, the `getHttpSession` returns the HTTP session associated currently with this method invocation. This method returns `null` if there is no HTTP session currently active and associated with this service endpoint. An endpoint class should not rely on an active HTTP session being always there; the underlying JAX-RPC runtime system is responsible for managing whether or not there is an active HTTP session.

The `getHttpSession` method throws `JAXRPCException` if invoked by an non HTTP bound endpoint.

**Returns:**
The HTTP session associated with the current invocation or
null if there is no active session.

**Throws:**
JAXRPCException - If this method invoked by any non-HTTP bound endpoint

**See Also:**
 HttpSession

```java
public ServletContext getServletContext()
```

The method `getServletContext` returns the `ServletContext` associated with the web application that contain this endpoint. According to the Servlet specification, There is one context per web application (installed as a WAR) per JVM. A servlet based service endpoint is deployed as part of a web application.

**Returns:**
ServletContext

**See Also:**
ServletContext

```java
public boolean isUserInRole(String role)
```

`role` String

`return` boolean

false
boolean isUserInRole(String role)

Returns a boolean indicating whether the authenticated user for the current method invocation on the endpoint instance is included in the specified logical "role".

Parameters:
role - a String specifying the name of the role

Returns:
a boolean indicating whether the authenticated user associated with the current method invocation belongs to a given role; false if the user has not been authenticated.
**javax.servlet**  
**Class ServletException**

java.lang.Object  
  ▼ java.lang.Throwable  
    ▼ java.lang.Exception  
      ▼ javax.servlet.ServletException

**All Implemented Interfaces:**  
Serializable

**Direct Known Subclasses:**  
UnavailableException

```
public class ServletException
    extends Exception

Extends: Throwable > Exception
Extended by: UnavailableException

servlet
```

Defines a general exception a servlet can throw when it encounters difficulty.

**Author:**  
Various

**See Also:**  
Serialized Form

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServletException()</td>
<td>Constructs a new servlet exception.</td>
</tr>
<tr>
<td>ServletException(String message)</td>
<td>Constructs a new servlet exception with the specified message.</td>
</tr>
</tbody>
</table>
**ServletException** *(String message, Throwable rootCause)*

Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation, including a description message.

**ServletException** *(Throwable rootCause)*

Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getRootCause()</td>
<td>Returns the exception that caused this servlet exception.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getRootCause
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

**public ServletException()**

Constructs a new servlet exception.
public ServletException(String message)

message String

ServletException

Constructs a new servlet exception with the specified message. The message can be written to the server log and/or displayed for the user.

Parameters:
message - a String specifying the text of the exception message

public ServletException(String message, Throwable rootCause)

message String
rootCause servlet Throwable Servlet

ServletException

Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation, including a description message.

Parameters:
message - a String containing the text of the exception message
rootCause - the Throwable exception that interfered with the
public ServletException(Throwable rootCause)

server servlet “” servlet

Thrownables getLocalizedMessage ServletException

rootCause servlet Throwable servlet

ServletException

public ServletException(Throwable rootCause)

Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation. The exception's message is based on the localized message of the underlying exception.

This method calls the getLocalizedMessage method on the Throwable exception to get a localized exception message. When subclassing ServletException, this method can be overridden to create an exception message designed for a specific locale.

Parameters:

rootCause - the Throwable exception that interfered with the servlet's normal operation, making the servlet exception necessary

Method Detail

public Throwable getRootCause()
getRootCause

public Throwable getRootCause()

Returns the exception that caused this servlet exception.

**Returns:**
the Throwable that caused this servlet exception
javax.servlet  **Class ServletInputStream**

**java.lang.Object**
  └ java.io.InputStream
      └ javax.servlet.ServletInputStream

**All Implemented Interfaces:**
  Closeable

---

```java
public abstract class ServletInputStream extends InputStream

Extends: java.io.InputStream

readLine  HTTP POST
```

**ServletInputStream**  ServletRequest#getInputStream

-servlet  java.io.InputStream.read()

**See also**  javax.servlet.ServletRequest

---

Provides an input stream for reading binary data from a client request, including an efficient `readLine` method for reading data one line at a time. With some protocols, such as HTTP POST and PUT, a ServletInputStream object can be used to read data sent from the client.

A ServletInputStream object is normally retrieved via the ServletRequest.getInputStream() method.

This is an abstract class that a servlet container implements. Subclasses of this class must implement the `java.io.InputStream.read()` method.

**Author:**
  Various
See Also:
 ServletRequest

## Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected ServletInputStream()</td>
<td>Does nothing, because this is an abstract class.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int readLine(byte[] b, int off, int len)</td>
<td>Reads the input stream, one line at a time.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.io.InputStream

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>available, close, mark, markSupported, read, read, read, reset, skip</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</td>
</tr>
</tbody>
</table>

## Constructor Detail

protected ServletInputStream()

ServletInputStream

protected ServletInputStream()

    Does nothing, because this is an abstract class.

## Method Detail
public int readLine(byte[] b, int off, int len) throws java.io.IOException

-1

$b$ byte
$off$ int
$len$ int
$return$ -1

Throws java.io.IOException:

readLine

public int readLine(byte[] b,
                     int off,
                     int len)
        throws IOException

Reads the input stream, one line at a time. Starting at an offset, reads bytes into an array, until it reads a certain number of bytes or reaches a newline character, which it reads into the array as well.

This method returns -1 if it reaches the end of the input stream before reading the maximum number of bytes.

Parameters:
   $b$ - an array of bytes into which data is read
   $off$ - an integer specifying the character at which this method begins reading
   $len$ - an integer specifying the maximum number of bytes to read

Returns:
   an integer specifying the actual number of bytes read, or -1 if the end of the stream is reached

Throws:
IOException - if an input or output exception has occurred

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet Class ServletOutputStream

java.lang.Object
   └ java.io.OutputStream
      └ javax.servlet.ServletOutputStream

All Implemented Interfaces: Closeable, Flushable

public abstract class ServletOutputStream

extends OutputStream

Extends: java.io.OutputStream

ServletOutputStream ServletResponse: servlet

java.io.OutputStream.write(int)

See also javax.servlet.ServletResponse

Provides an output stream for sending binary data to the client. A ServletOutputStream object is normally retrieved via the ServletResponse.getOutputStream() method.

This is an abstract class that the servlet container implements. Subclasses of this class must implement the java.io.OutputStream.write(int) method.

Author: Various

See Also: ServletResponse

Constructor Summary
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>print(boolean b)</code></td>
<td>Writes a boolean value to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(char c)</code></td>
<td>Writes a character to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(double d)</code></td>
<td>Writes a double value to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(float f)</code></td>
<td>Writes a float value to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(int i)</code></td>
<td>Writes an int to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(long l)</code></td>
<td>Writes a long value to the client, with no carriage return-line feed (CRLF) at the end.</td>
</tr>
<tr>
<td><code>print(String s)</code></td>
<td>Writes a string to the client, without a carriage return-line feed (CRLF) character at the end.</td>
</tr>
<tr>
<td><code>println()</code></td>
<td>Writes a carriage return-line feed (CRLF) to the client.</td>
</tr>
<tr>
<td><code>println(boolean b)</code></td>
<td>Writes a boolean value to the client, followed by a carriage return-line feed (CRLF).</td>
</tr>
<tr>
<td><code>println(char c)</code></td>
<td>Writes a character to the client, followed by a carriage return-line feed (CRLF).</td>
</tr>
<tr>
<td><code>println(double d)</code></td>
<td>Writes a double value to the client, followed by a carriage return-line feed (CRLF).</td>
</tr>
</tbody>
</table>
void println(float f)
  Writes a float value to the client, followed by a carriage return-line feed (CRLF).

void println(int i)
  Writes an int to the client, followed by a carriage return-line feed (CRLF).

void println(long l)
  Writes a long value to the client, followed by a carriage return-line feed (CRLF).

void println(String s)
  Writes a String to the client, followed by a carriage return-line feed (CRLF).

Methods inherited from class java.io.OutputStream
close, flush, write, write, write

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

protected ServletOutputStream()

ServletOutputStream

protected ServletOutputStream()
  Does nothing, because this is an abstract class.

Method Detail
public void print(String s) throws java.io.IOException

String - (CRLF)

    s

Throws java.io.IOException:

print

public void print(String s)
throws IOException

    Prints a String to the client, without a carriage return-line feed (CRLF) character at the end.

Parameters:
    s - the String to send to the client

    IOException - if an input or output exception occurred

public void print(boolean b) throws java.io.IOException

boolean - (CRLF)

    b

Throws java.io.IOException:

print

public void print(boolean b)
throws IOException

    Prints a boolean value to the client, with no carriage return-line feed (CRLF) character at the end.

Parameters:
    b - the boolean value to send to the client

    IOException - if an input or output exception occurred
public void print(char c) throws java.io.IOException
- (CRLF)
  c
  Throws java.io.IOException:

print

public void print(char c)  
throws IOException

  Writes a character to the client, with no carriage return-line feed (CRLF) at the end.

  Parameters:
  c - the character to send to the client

  Throws:
  IOException - if an input or output exception occurred

public void print(int i) throws java.io.IOException
int - (CRLF)
  i
  int
  Throws java.io.IOException:

print

public void print(int i)  
throws IOException

  Writes an int to the client, with no carriage return-line feed (CRLF) at the end.

  Parameters:
i - the int to send to the client

Throws:

IOException - if an input or output exception occurred

public void print(long l) throws java.io.IOException

long - (CRLF)

/                     long

Throws         java.io.IOException:

print

public void print(long l) throws IOException

Writes a long value to the client, with no carriage return-line feed (CRLF) at the end.

Parameters:

l - the long value to send to the client

Throws:

IOException - if an input or output exception occurred

public void print(float f) throws java.io.IOException

float - (CRLF)

/                     float

Throws         java.io.IOException:

print

public void print(float f) throws IOException

Writes a float value to the client, with no carriage return-line feed (CRLF) at the end.
Parameters:
f - the float value to send to the client

Throws:
IOException - if an input or output exception occurred

---

public void print(double d) throws java.io.IOException

double - (CRLF)

\[ d \]

Throws
java.io.IOException:

---

print

public void print(double d)
throws IOException

Writes a double value to the client, with no carriage return-line feed (CRLF) at the end.

Parameters:
d - the double value to send to the client

Throws:
IOException - if an input or output exception occurred

---

println

public void println() throws java.io.IOException
- (CRLF)

Throws
java.io.IOException:

---

println

public void println()
throws IOException

Writes a carriage return-line feed (CRLF) to the client.
Throws:  
[IOException] - if an input or output exception occurred

```java
public void println(String s) throws java.io.IOException
String - (CRLF)
    s                  String
    StringBuffer
Throws  java.io.IOException:
```

println

```java
public void println(String s) throws IOException
    String s
    String
    StringBuffer
```

Writes a String to the client, followed by a carriage return-line feed (CRLF).

**Parameters:**
- s - the String to write to the client

**Throws:**
- [IOException] - if an input or output exception occurred

```java
public void println(boolean b) throws java.io.IOException
boolean - (CRLF)
    b                  boolean
    boolean
```

println

```java
public void println(boolean b) throws IOException
    boolean b
    boolean
```

Writes a boolean value to the client, followed by a carriage return-line feed (CRLF).
public void println(char c) throws java.io.IOException - (CRLF)
    c
    Throws java.io.IOException:

println

public void println(char c)
    throws IOException

    Writes a character to the client, followed by a carriage return-line feed (CRLF).

Parameters:
    c - the character to write to the client

Throws:
    IOException - if an input or output exception occurred

public void println(int i) throws java.io.IOException
    int - (CRLF)
    i
    int
    Throws java.io.IOException:

println

public void println(int i)
    throws IOException

    Writes an int to the client, followed by a carriage return-line feed
(CRLF) character.

**Parameters:**
- `i` - the int to write to the client

**Throws:**
- [IOException](https://docs.oracle.com/javase/8/docs/api/java/io/IOException.html) - if an input or output exception occurred

```java
public void println(long l) throws java.io.IOException
long - (CRLF)

/          long
Throws     java.io.IOException:
```

**println**

```java
public void println(long l)
throws IOException

Writes a long value to the client, followed by a carriage return-line feed (CRLF).

**Parameters:**
- `l` - the long value to write to the client

**Throws:**
- [IOException](https://docs.oracle.com/javase/8/docs/api/java/io/IOException.html) - if an input or output exception occurred

```java
public void println(float f) throws java.io.IOException
float - (CRLF)

/          float
f
Throws     java.io.IOException:
```

**println**

```java
public void println(float f)
throws IOException
```
Writes a float value to the client, followed by a carriage return-line feed (CRLF).

**Parameters:**
- f - the float value to write to the client

**Throws:**
- [IOException](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/IOException.html) - if an input or output exception occurred

```java
public void println(double d) throws java.io.IOException
```

**double** - (CRLF)

```java
double d
```

**Throws**
- java.io.IOException:

---

**println**

```java
public void println(double d)
```

Throws [IOException](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/IOException.html)

Writes a double value to the client, followed by a carriage return-line feed (CRLF).

**Parameters:**
- d - the double value to write to the client

**Throws:**
- [IOException](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/IOException.html) - if an input or output exception occurred

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/io/IOException.html).
PS:
javax.servlet  Interface ServletRequest

All Known Subinterfaces:  
HttpServletRequest

All Known Implementing Classes:  
HttpServletRequestWrapper, ServletRequestWrapper

---

public interface ServletRequest

Implemented by: HttpServletRequest, ServletRequestWrapper

.servlet.servlet ServletRequest ServletRequest

javax.servlet.http.HttpServletRequest HTTP

See also  javax.servlet.http.HttpServletRequest

---

Defines an object to provide client request information to a servlet. The servlet container creates a ServletRequest object and passes it as an argument to the servlet's service method.

A ServletRequest object provides data including parameter name and values, attributes, and an input stream. Interfaces that extend ServletRequest can provide additional protocol-specific data (for example, HTTP data is provided by HttpServletRequest).

Author:

Various

See Also:

HttpServletRequest

---

Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAttribute(String name)</code></td>
<td>Returns the value of the named attribute as an Object, or null if no attribute of the given name exists.</td>
</tr>
<tr>
<td><code>getAttributeNames()</code></td>
<td>Returns an Enumeration containing the names of the attributes available to this request.</td>
</tr>
<tr>
<td><code>getCharacterEncoding()</code></td>
<td>Returns the name of the character encoding used in the body of this request.</td>
</tr>
<tr>
<td><code>getContentLength()</code></td>
<td>Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known.</td>
</tr>
<tr>
<td><code>getContentType()</code></td>
<td>Returns the MIME type of the body of the request, or null if the type is not known.</td>
</tr>
<tr>
<td><code>getInputStream()</code></td>
<td>Retrieves the body of the request as binary data using a <code>ServletInputStream</code>.</td>
</tr>
<tr>
<td><code>getLocalAddr()</code></td>
<td>Returns the Internet Protocol (IP) address of the interface on which the request was received.</td>
</tr>
<tr>
<td><code>getLocale()</code></td>
<td>Returns the preferred <code>Locale</code> that the client will accept content in, based on the Accept-Language header.</td>
</tr>
<tr>
<td><code>getLocales()</code></td>
<td>Returns an Enumeration of <code>Locale</code> objects indicating, in decreasing order starting with the preferred locale, the locales that are acceptable to the client based on the Accept-Language header.</td>
</tr>
<tr>
<td><code>getLocalName()</code></td>
<td>Returns the host name of the Internet Protocol (IP) interface on which the request was received.</td>
</tr>
<tr>
<td><code>getLocalPort()</code></td>
<td>Returns the Internet Protocol (IP) port number of</td>
</tr>
</tbody>
</table>
the interface on which the request was received.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getParameter(String name)</code></td>
<td>Returns the value of a request parameter as a String, or null if the parameter does not exist.</td>
</tr>
<tr>
<td><code>getParameterMap()</code></td>
<td>Returns a java.util.Map of the parameters of this request.</td>
</tr>
<tr>
<td><code>getParameterNames()</code></td>
<td>Returns an Enumeration of String objects containing the names of the parameters contained in this request.</td>
</tr>
<tr>
<td><code>getParameterValues(String name)</code></td>
<td>Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.</td>
</tr>
<tr>
<td><code>getProtocol()</code></td>
<td>Returns the name and version of the protocol the request uses in the form <code>protocol/majorVersion.minorVersion</code>, for example, HTTP/1.1.</td>
</tr>
<tr>
<td><code>getReader()</code></td>
<td>Retrieves the body of the request as character data using a BufferedReader.</td>
</tr>
<tr>
<td><code>getRealPath(String path)</code></td>
<td>Deprecated. As of Version 2.1 of the Java Servlet API, use <code>ServletContext.getRealPath(java.lang.String)</code> instead.</td>
</tr>
<tr>
<td><code>getRemoteAddr()</code></td>
<td>Returns the Internet Protocol (IP) address of the client or last proxy that sent the request.</td>
</tr>
<tr>
<td><code>getRemoteHost()</code></td>
<td>Returns the fully qualified name of the client or the last proxy that sent the request.</td>
</tr>
<tr>
<td><code>getRemotePort()</code></td>
<td>Returns the Internet Protocol (IP) source port of</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>RequestDispatcher</td>
<td>The <code>RequestDispatcher</code> object that acts as a wrapper for the resource located at the given path.</td>
</tr>
<tr>
<td>getScheme()</td>
<td>Returns the name of the scheme used to make this request, for example, http, https, or ftp.</td>
</tr>
<tr>
<td>getServerName()</td>
<td>Returns the host name of the server to which the request was sent.</td>
</tr>
<tr>
<td>getServerPort()</td>
<td>Returns the port number to which the request was sent.</td>
</tr>
<tr>
<td>isSecure()</td>
<td>Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.</td>
</tr>
<tr>
<td>removeAttribute(String name)</td>
<td>Removes an attribute from this request.</td>
</tr>
<tr>
<td>setAttribute(String name, Object o)</td>
<td>Stores an attribute in this request.</td>
</tr>
<tr>
<td>setCharacterEncoding(String env)</td>
<td>Overrides the name of the character encoding used in the body of this request.</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public Object getAttribute(String name) {
    Object null
}
```

**servle HTTPS**

`javax.servlet.request.X509Certificate`  
`ServletRequest#setAttribute`  
`RequestDispatcher`
getAttribute

```
public java.util.Enumeration<E> getAttributeNames()
```

```java
Object getAttribute(String name)
```

Returns the value of the named attribute as an `Object`, or `null` if no attribute of the given name exists.

Attributes can be set two ways. The servlet container may set attributes to make available custom information about a request. For example, for requests made using HTTPS, the attribute `javax.servlet.request.X509Certificate` can be used to retrieve information on the certificate of the client. Attributes can also be set programatically using `setAttribute(java.lang.String, java.lang.Object)`. This allows information to be embedded into a request before a `RequestDispatcher` call.

Attribute names should follow the same conventions as package names. This specification reserves names matching `java.*`, `javax.*`, and `sun.*`.

**Parameters:**
- `name` - A `String` specifying the name of the attribute

**Returns:**
- An `Object` containing the value of the attribute, or `null` if the attribute does not exist
getAttributeNames

Enumeration getAttributeNames()

Returns an Enumeration containing the names of the attributes available to this request. This method returns an empty Enumeration if the request has no attributes available to it.

Returns:
   an Enumeration of strings containing the names of the request's attributes

public String getCharacterEncoding()

return String null

getCharacterEncoding

String getCharacterEncoding()

Returns the name of the character encoding used in the body of this request. This method returns null if the request does not specify a character encoding

Returns:
   a String containing the name of the character encoding, or null if the request does not specify a character encoding

public void setCharacterEncoding(String env) throws java.io.UnsupportedEncodingException
getReader()

env String

Throws java.io.UnsupportedEncodingException: ServletRequest
setCharacterEncoding

```java
void setCharacterEncoding(String env)
    throws UnsupportedEncodingException
```

Overrides the name of the character encoding used in the body of this request. This method must be called prior to reading request parameters or reading input using getReader(). Otherwise, it has no effect.

**Parameters:**
- `env` - String containing the name of the character encoding.

**Throws:**
- `UnsupportedEncodingException` - if this ServletRequest is still in a state where a character encoding may be set, but the specified encoding is invalid

---

public int getContentTypeLength()

```java
    -1 HTTP servlet CGI
    CONTENT_LENGTH
    return -1
```

**getContentLength**

```java
int getContentLength()
```

Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known. For HTTP servlets, same as the value of the CGI variable CONTENT_LENGTH.

**Returns:**
- an integer containing the length of the request body or -1 if the length is not known
public String getContentType()

MIME CONTENT_TYPE
return MIME String null

getContentType

String getContentType()

Returns the MIME type of the body of the request, or null if the type is not known. For HTTP servlets, same as the value of the CGI variable CONTENT_TYPE.

Returns:
- a String containing the name of the MIME type of the request,
- or null if the type is not known

public ServletInputStream getInputStream() throws java.io.IOException

ServletInputStream getInputStream()

throws IOException

Retrieves the body of the request as binary data using a ServletInputStream. Either this method or getReader() may be called to read the body, not both.

Returns:
- a ServletInputStream object containing the body of the request
### getParameter(String name)

**public String getParameter(String name)**

*String* null HTTP servlet

- **Throws:**
  - `IllegalStateException` - if the `getReader()` method has already been called for this request
  - `IOException` - if an input or output exception occurred

<table>
<thead>
<tr>
<th>Method Call</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>#getParameterValues()</code></td>
<td>Get all parameters for a request.</td>
</tr>
</tbody>
</table>

**Http POST**

1. `#getInputStream`
2. `#getReader`

- `String name`
- `String return`
- `See also` `getParameterValues`

---

### getParameter

**String getParameter(String name)**

Returns the value of a request parameter as a `String`, or `null` if the parameter does not exist. Request parameters are extra information sent with the request. For HTTP servlets, parameters are contained in the query string or posted form data.

You should only use this method when you are sure the parameter has only one value. If the parameter might have more than one value, use `getParameterValues(java.lang.String)`.

If you use this method with a multivalued parameter, the value returned is equal to the first value in the array returned by `getParameterValues`.

If the parameter data was sent in the request body, such as occurs
with an HTTP POST request, then reading the body directly via `getInputStream()` or `getReader()` can interfere with the execution of this method.

**Parameters:**
- `name` - a `String` specifying the name of the parameter

**Returns:**
- a `String` representing the single value of the parameter

**See Also:**
- `getParameterValues(java.lang.String)`

```java
public java.util.Enumeration<String> getParameterNames()
```

**getParameterNames**

Returns an `Enumeration` of `String` objects containing the names of the parameters contained in this request. If the request has no parameters, the method returns an empty `Enumeration`.

**Returns:**
- an `Enumeration` of `String` objects, each `String` containing the name of a request parameter; or an empty `Enumeration` if the request has no parameters

```java
public String[] getParameterValues(String name)
```

**getParameterValues(String name)**

1

- `name` - a `String`

**See also:**
- `getParameter`
getParameterValues

String[] getParameterValues(String name)

Returns an array of String objects containing all of the values the given request parameter has, or null if the parameter does not exist.

If the parameter has a single value, the array has a length of 1.

Parameters:
    name - a String containing the name of the parameter whose value is requested

Returns:
    an array of String objects containing the parameter's values

See Also:
    getParameter(java.lang.String)

getParameterMap

public java.util.Map<K, V> getParameterMap()

java.util.Map HTTP servlet
    return java.util.Map String String

getParameterMap

Map getParameterMap()

Returns a java.util.Map of the parameters of this request. Request parameters are extra information sent with the request. For HTTP servlets, parameters are contained in the query string or posted form data.

Returns:
    an immutable java.util.Map containing parameter names as keys and parameter values as map values. The keys in the parameter map are of type String. The values in the parameter map are of type String array.
public String getProtocol()

String getProtocol()

Returns the name and version of the protocol the request uses in the form protocol/majorVersion.minorVersion, for example, HTTP/1.1. For HTTP servlets, the value returned is the same as the value of the CGI variable SERVER_PROTOCOL.

Returns:

a String containing the protocol name and version number

public String getScheme()

String getScheme()

Returns the name of the scheme used to make this request, for example, http, https, or ftp. Different schemes have different rules for constructing URLs, as noted in RFC 1738.

Returns:

a String containing the name of the scheme used to make this request
public String getServerName()

getServerName

String getServerName()

Returns the host name of the server to which the request was sent. It is the value of the part before ":" in the Host header value, if any, or the resolved server name, or the server IP address.

Returns:
- a String containing the name of the server

public int getServerPort()

getServerPort

int getServerPort()

Returns the port number to which the request was sent. It is the value of the part after ":" in the Host header value, if any, or the server port where the client connection was accepted on.

Returns:
- an integer specifying the port number

public java.io.BufferedReader getReader() throws java.io.IOException

BufferedReader getReader

#getInputStream
getReader

**BufferedReader getReader()**

Throws: `IOException`

Retrieves the body of the request as character data using a `BufferedReader`. The reader translates the character data according to the character encoding used on the body. Either this method or `getInputStream()` may be called to read the body, not both.

Returns:

- A `BufferedReader` containing the body of the request

Throws:

- `UnsupportedEncodingException` - if the character set encoding used is not supported and the text cannot be decoded
- `IllegalStateException` - if `getInputStream()` method has been called on this request
- `IOException` - if an input or output exception occurred

See Also:

- `getInputStream()`

---

```
public String getRemoteAddr()
```

Internet Protocol (IP)  HTTP servlet CGI

**REMOTE_ADDR**

```
return IP String
```

**getRemoteAddr**

```
String getRemoteAddr()
```
Returns the Internet Protocol (IP) address of the client or last proxy that sent the request. For HTTP servlets, same as the value of the CGI variable REMOTE_ADDR.

**Returns:**

a String containing the IP address of the client that sent the request

```java
public String getRemoteHost()
```

**IP HTTP servlet CGI**

return String

**getRemoteHost**

```java
String getRemoteHost()
```

Returns the fully qualified name of the client or the last proxy that sent the request. If the engine cannot or chooses not to resolve the hostname (to improve performance), this method returns the dotted-string form of the IP address. For HTTP servlets, same as the value of the CGI variable REMOTE_HOST.

**Returns:**

a String containing the fully qualified name of the client

```java
public void setAttribute(String name, Object o)
```

**RequestDispatcher**

```java
java.*javax.* com.sun.* Sun Microsystems
null #removeAttribute
RequestDispatcher Web servlet servlet
```

name String
setAttribute

void setAttribute(String name, Object o)

Stores an attribute in this request. Attributes are reset between requests. This method is most often used in conjunction with RequestDispatcher.

Attribute names should follow the same conventions as package names. Names beginning with java.*, javax.*, and com.sun.*, are reserved for use by Sun Microsystems.
If the object passed in is null, the effect is the same as calling removeAttribute(java.lang.String).
It is warned that when the request is dispatched from the servlet resides in a different web application by RequestDispatcher, the object set by this method may not be correctly retrieved in the caller servlet.

Parameters:

   name - a String specifying the name of the attribute
   o - the Object to be stored

public void removeAttribute(String name)

   java.*javax.*  com.sun.*  Sun Microsystems
   name  String

removeAttribute
void **removeAttribute**(*String* name)

Removes an attribute from this request. This method is not generally needed as attributes only persist as long as the request is being handled.

Attribute names should follow the same conventions as package names. Names beginning with java.*, javax.*, and com.sun.*, are reserved for use by Sun Microsystems.

**Parameters:**

name - a String specifying the name of the attribute to remove

---

**public java.util.Locale getLocale()**

Accept-Language

return

**getLocale**

**Locale getLocale()**

Returns the preferred **Locale** that the client will accept content in, based on the Accept-Language header. If the client request doesn't provide an Accept-Language header, this method returns the default locale for the server.

**Returns:**

the preferred **Locale** for the client

---

**public java.utilEnumeration**<E> **getLocales()**

**Locale Enumeration** Accept-Language

Accept-Language

return

**getLocales**

**Locale Enumeration**

return
getLocales

Enumeration getLocales()

Returns an Enumeration of Locale objects indicating, in decreasing order starting with the preferred locale, the locales that are acceptable to the client based on the Accept-Language header. If the client request doesn't provide an Accept-Language header, this method returns an Enumeration containing one Locale, the default locale for the server.

Returns:
    an Enumeration of preferred Locale objects for the client

public boolean isSecure()

boolean HTTPS

    return boolean

isSecure

boolean isSecure()

Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.

Returns:
    a boolean indicating if the request was made using a secure channel

google public RequestDispatcher getRequestDispatcher(String path)

RequestDispatcher RequestDispatcher

    RequestDispatcher

servlet "/" servlet
getRequestDispatcher

getRequestDispatcher(String path)

Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path. A RequestDispatcher object can be used to forward a request to the resource or to include the resource in a response. The resource can be dynamic or static.

The pathname specified may be relative, although it cannot extend outside the current servlet context. If the path begins with a "/" it is interpreted as relative to the current context root. This method returns null if the servlet container cannot return a RequestDispatcher.

The difference between this method and ServletContext.getRequestDispatcher(String) is that this method can take a relative path.

Parameters:
- path - a String specifying the pathname to the resource. If it is relative, it must be relative against the current servlet.

Returns:
- a RequestDispatcher object that acts as a wrapper for the resource at the specified path, or null if the servlet container cannot return a RequestDispatcher

See Also:
- RequestDispatcher
- ServletContext.getRequestDispatcher(String)
public String getRealPath(String path)

getRealPath

String getRealPath(String path)

Deprecated. As of Version 2.1 of the Java Servlet API, use ServletContext.getRealPath(java.lang.String) instead.

public int getRemotePort()

getRemotePort

int getRemotePort()

Returns the Internet Protocol (IP) source port of the client or last proxy that sent the request.

Returns:
an integer specifying the port number

Since:
2.4

public String getLocalName()

getLocalName

String getLocalName()

Internet Protocol (IP)

return IP String

since 2.4
getLocalName

public String getLocalName()

    Returns the host name of the Internet Protocol (IP) interface on which the request was received.

    Returns:  
        a String containing the host name of the IP on which the request was received.

    Since:  
        2.4

public String getLocalAddr()

    Returns the Internet Protocol (IP) address of the interface on which the request was received.

    Returns:  
        a String containing the IP address on which the request was received.

    Since:  
        2.4

public int getLocalPort()
void getLocalPort()

Returns the Internet Protocol (IP) port number of the interface on which the request was received.

Returns:

an integer specifying the port number

Since:

2.4
Class ServletRequestAttributeEvent

public class ServletRequestAttributeEvent
extends ServletRequestEvent

Extends: java.util.EventObject > ServletRequestEvent

Since: Servlet 2.4
See also: javax.servlet.ServletRequestAttributeListener

This is the event class for notifications of changes to the attributes of the servlet request in an application.

See Also:
ServletRequestAttributeListener, Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
source
Constructor Summary

ServletResponseAttributeEvent (ServletContext sc, ServletRequest request, String name, Object value)

Construct a ServletRequestAttributeEvent giving the servlet context of this web application, the ServletRequest whose attributes are changing and the name and value of the attribute.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getName()</td>
<td>Return the name of the attribute that changed on the ServletRequest.</td>
</tr>
<tr>
<td>getValue()</td>
<td>Returns the value of the attribute that has been added, removed or replaced.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.servlet.ServletRequestEvent

getServletContext, getServletRequest

Methods inherited from class java.util.EventObject

getSource, toString

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

public ServletRequestAttributeEvent (ServletContext sc, ServletRequest request, String name, Object value)

Web servlet ServletRequest
ServletResponseAttributeEvent

sc ServletContext
ServletRequestAttributeEvent

public ServletRequestAttributeEvent(ServletContext sc, ServletRequest request, String name, Object value)

Construct a ServletRequestAttributeEvent giving the servlet context of this web application, the ServletRequest whose attributes are changing and the name and value of the attribute.

Parameters:
- `sc` - the ServletContext that is sending the event.
- `request` - the ServletRequest that is sending the event.
- `name` - the name of the request attribute.
- `value` - the value of the request attribute.

Method Detail

public String getName()

ServletRequest return

getName

public String getName()

Return the name of the attribute that changed on the ServletRequest.

Returns:
- the name of the changed request attribute
public Object getValue()

    return

getValue

public Object getValue()

    Returns the value of the attribute that has been added, removed or replaced. If the attribute was added, this is the value of the attribute. If the attribute was removed, this is the value of the removed attribute. If the attribute was replaced, this is the old value of the attribute.

    Returns:
    the value of the changed request attribute
javax.servlet Interface
ServletRequestAttributeListener

All Superinterfaces:
EventListener

public interface ServletRequestAttributeListener
extends EventListener

Implements: java.util.EventListener

ServletRequestAttributeListener Web Web servlet	servlet

since Servlet 2.4

A ServletRequestAttributeListener can be implemented by the developer interested in being notified of request attribute changes. Notifications will be generated while the request is within the scope of the web application in which the listener is registered. A request is defined as coming into scope when it is about to enter the first servlet or filter in each web application, as going out of scope when it exits the last servlet or the first filter in the chain.

Since:
Servlet 2.4

<table>
<thead>
<tr>
<th>Method Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>void attributeAdded(ServletRequestAttributeEvent srae)</td>
<td>Notification that a new attribute was added to the servlet request.</td>
</tr>
<tr>
<td>void attributeRemoved(ServletRequestAttributeEvent srae)</td>
<td>Notification that an existing attribute has been removed from</td>
</tr>
</tbody>
</table>
Method Detail

public void attributeAdded(ServletRequestAttributeEvent srae)
  servlet

attributeAdded

void attributeAdded(ServletRequestAttributeEvent srae)
  Notification that a new attribute was added to the servlet request. Called after the attribute is added.

public void attributeRemoved(ServletRequestAttributeEvent srae)
  servlet

attributeRemoved

void attributeRemoved(ServletRequestAttributeEvent srae)
  Notification that an existing attribute has been removed from the servlet request. Called after the attribute is removed.

public void attributeReplaced(ServletRequestAttributeEvent srae)
servlet

attributeReplaced

void attributeReplaced(ServletRequestAttributeEvent srae)

Notification that an attribute was replaced on the servlet request. Called after the attribute is replaced.
javax.servlet Class ServletRequestEvent

java.lang.Object
   java.util.EventObject
      javax.servlet.ServletRequestEvent

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   ServletRequestAttributeEvent

public class ServletRequestEvent
   extends EventObject

Extends: java.util.EventObject
Extended by: ServletRequestAttributeEvent

ServletRequest Web ServletContext
   since Servlet 2.4
See also javax.servlet.ServletRequestListener

Events of this kind indicate lifecycle events for a ServletRequest. The source of the event is the ServletContext of this web application.

Since:
   Servlet 2.4
See Also:
   ServletRequestListener, Serialized Form

Field Summary

Fields inherited from class java.util.EventObject
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServletRequestEvent( ServletContext <code>sc</code>, ServletRequest <code>request</code>)</td>
<td>Construct a ServletRequestEvent for the given ServletContext and ServletRequest.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServletContext <code>getServletContext()</code></td>
<td>Returns the ServletContext of this web application.</td>
</tr>
<tr>
<td>ServletRequest <code>getServletRequest()</code></td>
<td>Returns the ServletRequest that is changing.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.util.EventObject
getSource, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

### Constructor Detail

```java
class ServletRequestEvent {
    public ServletRequestEvent(ServletContext `sc`, ServletRequest `request`)
        ServletContext `sc` Web ServletContext
        ServletRequest `request` ServletRequest
}
```

ServletRequestEvent
public ServletRequestEvent(ServletContext sc, ServletRequest request)

Construct a ServletRequestEvent for the given ServletContext and ServletRequest.

Parameters:
  sc - the ServletContext of the web application.
  request - theServletRequest that is sending the event.

Method Detail

public ServletRequest getServletRequest()
ServletRequest

getServletRequest

public ServletRequest getServletRequest()

Returns the ServletRequest that is changing.

public ServletContext getServletContext()
Web  ServletContext

getServletContext

public ServletContext getServletContext()

Returns the ServletContext of this web application.

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet Interface ServletRequestListener

All Superinterfaces: 
EventListener

public interface ServletRequestListener 

extendsEventListener

Implements: java.util.EventListener

ServletRequestListener Web Web servlet servlet

since Servlet 2.4

A ServletRequestListener can be implemented by the developer interested in being notified of requests coming in and out of scope in a web component. A request is defined as coming into scope when it is about to enter the first servlet or filter in each web application, as going out of scope when it exits the last servlet or the first filter in the chain.

Since: Servlet 2.4

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void requestDestroyed(ServletRequestEvent sre)</td>
<td>The request is about to go out of scope of the web application.</td>
</tr>
<tr>
<td>void requestInitialized(ServletRequestEvent sre)</td>
<td>The request is about to come into scope of the web application.</td>
</tr>
</tbody>
</table>
public void requestDestroyed(ServletRequestEvent sre)  
Web

void requestDestroyed(ServletRequestEvent sre)

The request is about to go out of scope of the web application.

public void requestInitialized(ServletRequestEvent sre)  
Web

void requestInitialized(ServletRequestEvent sre)

The request is about to come into scope of the web application.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.servlet

**Class ServletRequestWrapper**

java.lang.Object

└ javax.servlet.ServletRequestWrapper

### All Implemented Interfaces:

- ServletRequest

### Direct Known Subclasses:

- HttpServletRequestWrapper

---

```java
public class ServletRequestWrapper
extends Object
implements ServletRequest
```

**Implements:** ServletRequest  
**Extended by:** HttpServletRequestWrapper

ServletRequest  Servlet  Wrapper  Decorator

**since**  v 2.3

**See also**  javax.servlet.ServletRequest

Provides a convenient implementation of the ServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet. This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped request object.

**Since:**

v 2.3

**See Also:**

ServletRequest
### Constructor Summary

**ServletRequestWrapper**(ServletRequest request)  
Creates a ServletRequest adaptor wrapping the given request object.

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object</td>
<td>getAttribute(String name)</td>
<td>The default behavior of this method is to call getAttribute(String name) on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration</td>
<td>getAttributeNames()</td>
<td>The default behavior of this method is to return getAttributeNames() on the wrapped request object.</td>
</tr>
<tr>
<td>String</td>
<td>getCharacterEncoding()</td>
<td>The default behavior of this method is to return getCharacterEncoding() on the wrapped request object.</td>
</tr>
<tr>
<td>int</td>
<td>getContentLength()</td>
<td>The default behavior of this method is to return getContentLength() on the wrapped request object.</td>
</tr>
<tr>
<td>String</td>
<td>getContentType()</td>
<td>The default behavior of this method is to return getContentType() on the wrapped request object.</td>
</tr>
<tr>
<td>ServletInputStream</td>
<td>getInputStream()</td>
<td>The default behavior of this method is to return getInputStream() on the wrapped request object.</td>
</tr>
<tr>
<td>String</td>
<td>getLocalAddr()</td>
<td>The default behavior of this method is to return getLocalAddr() on the wrapped request object.</td>
</tr>
<tr>
<td>Locale</td>
<td>getLocale()</td>
<td>The default behavior of this method is to return getLocale() on the wrapped request object.</td>
</tr>
<tr>
<td>Enumeration</td>
<td>getLocales()</td>
<td>The default behavior of this method is to return getLocales() on the wrapped request object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><code>getLocalName()</code></td>
<td>The default behavior of this method is to return <code>getLocalName()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getLocalPort()</code></td>
<td>The default behavior of this method is to return <code>getLocalPort()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getParameter(String name)</code></td>
<td>The default behavior of this method is to return <code>getParameter(String name)</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getParameterMap()</code></td>
<td>The default behavior of this method is to return <code>getParameterMap()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getParameterNames()</code></td>
<td>The default behavior of this method is to return <code>getParameterNames()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getParameterValues(String name)</code></td>
<td>The default behavior of this method is to return <code>getParameterValues(String name)</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getProtocol()</code></td>
<td>The default behavior of this method is to return <code>getProtocol()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getReader()</code></td>
<td>The default behavior of this method is to return <code>getReader()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getRealPath(String path)</code></td>
<td>The default behavior of this method is to return <code>getRealPath(String path)</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getRemoteAddr()</code></td>
<td>The default behavior of this method is to return <code>getRemoteAddr()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td><code>getRemoteHost()</code></td>
<td>The default behavior of this method is to return <code>getRemoteHost()</code> on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td>Default Behavior</td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HttpServletRequest.getRemotePort()</td>
<td>The default behavior of this method is to return getRemotePort() on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>HttpServletRequest.getRequest()</td>
<td>Return the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>RequestDispatcher.getRequestDispatcher(String path)</td>
<td>The default behavior of this method is to return getRequestDispatcher(String path) on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>String.getScheme()</td>
<td>The default behavior of this method is to return getScheme() on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>String.getServerName()</td>
<td>The default behavior of this method is to return getServerName() on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>int.getServerPort()</td>
<td>The default behavior of this method is to return getServerPort() on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>boolean.isSecure()</td>
<td>The default behavior of this method is to return isSecure() on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>void.removeAttribute(String name)</td>
<td>The default behavior of this method is to call removeAttribute(String name) on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>void.setAttribute(String name, Object o)</td>
<td>The default behavior of this method is to return setAttribute(String name, Object o) on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>void.setCharacterEncoding(String enc)</td>
<td>The default behavior of this method is to set the character encoding on the wrapped request object.</td>
<td></td>
</tr>
<tr>
<td>void.setRequest(HttpServletRequest request)</td>
<td>Sets the request object being wrapped.</td>
<td></td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang/Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public ServletRequestWrapper(ServletRequest request)
ServletRequest

Throws
IllegalArgumentException: request null

ServletRequestWrapper

public ServletRequestWrapper(ServletRequest request)

Creates a ServletRequest adaptor wrapping the given request object.

Throws:
IllegalArgumentException - if the request is null

Method Detail

public ServletRequest getRequest()

getRequest

public ServletRequest getRequest()

getRequest

Return the wrapped request object.

public void setRequest(ServletRequest request)
**setRequest**

public void setRequest(ServletRequest request)

Sets the request object being wrapped.

**Throws:**

IllegalArgumentException - if the request is null.

---

**public Object getAttribute(String name)**

getAttribute(String name)

**getAttribute**

public Object getAttribute(String name)

The default behavior of this method is to call getAttribute(String name) on the wrapped request object.

**Specified by:**

getAttribute in interface ServletRequest

**Parameters:**

name - a String specifying the name of the attribute

**Returns:**

an Object containing the value of the attribute, or null if the attribute does not exist

---

**public java.util.Enumeration<E> getAttributeNames()**

getAttributeNames()
**getAttributeNames**

```java
public Enumeration getAttributeNames()
```

The default behavior of this method is to return `getAttributeNames()` on the wrapped request object.

**Specified by:**
`getAttributeNames` in interface `ServletRequest`

**Returns:**
an `Enumeration` of strings containing the names of the request's attributes

---

**public String getCharacterEncoding()**

```java
getCharacterEncoding()
```

**getCharacterEncoding**

```java
public String getCharacterEncoding()
```

The default behavior of this method is to return `getCharacterEncoding()` on the wrapped request object.

**Specified by:**
`getCharacterEncoding` in interface `ServletRequest`

**Returns:**
a `String` containing the name of the character encoding, or `null` if the request does not specify a character encoding

---

**public void setCharacterEncoding(String enc) throws java.io.UnsupportedEncodingException**

```java
setCharacterEncoding
```
public void setCharacterEncoding(String enc)
    throws UnsupportedEncodingException

    The default behavior of this method is to set the character encoding
    on the wrapped request object.

    Specified by:
        setCharacterEncoding in interface ServletRequest

    Parameters:
        enc - String containing the name of the character encoding.

    Throws:
        UnsupportedEncodingException - if this ServletRequest is still in a
        state where a character encoding may be set, but the specified
        encoding is invalid

public int getContentType()
getContentType()

gContentType

public int getContentType()

    The default behavior of this method is to return getContentType()
    on the wrapped request object.

    Specified by:
        getContentType in interface ServletRequest

    Returns:
        an integer containing the length of the request body or -1 if the
        length is not known

-------------------

public String getContentType()
getContentType()
getContentType

public String getContentType()

The default behavior of this method is to return getContentType() on the wrapped request object.

Specified by:
getContentType in interface ServletRequest

Returns:
a String containing the name of the MIME type of the request, or null if the type is not known

---

public ServletInputStream getInputStream() throws java.io.IOException

getInputStream()

getInputStream

public ServletInputStream getInputStream() throws IOException

The default behavior of this method is to return getInputStream() on the wrapped request object.

Specified by:
getInputStream in interface ServletRequest

Returns:
a ServletInputStream object containing the body of the request

Throws:
IOException - if an input or output exception occurred

---

public String getParameter(String name)

getParameter(String name)
getParameter

public String getParameter(String name)

The default behavior of this method is to return getParameter(String name) on the wrapped request object.

Specified by:

generateParameter in interface ServletRequest

Parameters:

name - a String specifying the name of the parameter

Returns:

a String representing the single value of the parameter

See Also:

ServletRequest.getParameterValues(java.lang.String)

generateParameterMap

public java.util.Map<K, V> getParameterMap()

getParameterMap()

generateParameterMap

public Map getParameterMap()

The default behavior of this method is to return getParameterMap() on the wrapped request object.

Specified by:

generateParameterMap in interface ServletRequest

Returns:

an immutable java.util.Map containing parameter names as keys and parameter values as map values. The keys in the parameter map are of type String. The values in the parameter map are of type String array.

generateParameterNames

public java.util Enumeration<E> getParameterNames()

getParameterNames()
**getParameterNames**

public Enumeration getParameterNames()

The default behavior of this method is to return getParameterNames() on the wrapped request object.

**Specified by:**
getParameterNames in interface ServletRequest

**Returns:**
- an Enumeration of String objects, each String containing the name of a request parameter; or an empty Enumeration if the request has no parameters

---

**public String[] getParameterValues(String name)**

**getParameterValues(String name)**

**getParameterValues**

public String[] getParameterValues(String name)

The default behavior of this method is to return getParameterValues(String name) on the wrapped request object.

**Specified by:**
getParameterValues in interface ServletRequest

**Parameters:**
- name - a String containing the name of the parameter whose value is requested

**Returns:**
- an array of String objects containing the parameter's values

**See Also:**
ServletRequest.getParameter(java.lang.String)
public String getProtocol()
getProtocol()

getProtocol

public String getProtocol()

The default behavior of this method is to return getProtocol() on the
wrapped request object.

Specified by:

getProtocol in interface ServletRequest

Returns:

a String containing the protocol name and version number

public String getScheme()
getScheme()

getScheme

public String getScheme()

The default behavior of this method is to return getScheme() on the
wrapped request object.

Specified by:

getScheme in interface ServletRequest

Returns:

a String containing the name of the scheme used to make this
request

public String getServerName()
getServerName()
getServerName

public String getServerName()

    The default behavior of this method is to return getServerName() on the wrapped request object.

Specified by:
    getServerName in interface ServletRequest

Returns:
    a String containing the name of the server

getServerPort

public int getServerPort()

getServerPort()

getServerPort

public int getServerPort()

    The default behavior of this method is to return getServerPort() on the wrapped request object.

Specified by:
    getServerPort in interface ServletRequest

Returns:
    an integer specifying the port number

getReader

public java.io.BufferedReader getReader() throws java.io.IOException

getReader()

getReader

public BufferedReader getReader()
The default behavior of this method is to return getReader() on the wrapped request object.

Specified by:  
getReader in interface ServletRequest

Returns:  
a BufferedReader containing the body of the request

Throws:  
IOException - if an input or output exception occurred

See Also:  
ServletRequest.getInputStream()
public String getRemoteHost()

The default behavior of this method is to return getRemoteHost() on the wrapped request object.

Specified by:
getRemoteHost in interface ServletRequest

Returns:
a String containing the fully qualified name of the client

------

public void setAttribute(String name, Object o)
setAttribute(String name, Object o)

setAttribute

public void setAttribute(String name, Object o)

The default behavior of this method is to return setAttribute(String name, Object o) on the wrapped request object.

Specified by:
setAttribute in interface ServletRequest

Parameters:
name - a String specifying the name of the attribute
o - the Object to be stored

------

public void removeAttribute(String name)
removeAttribute(String name)

removeAttribute

public void removeAttribute(String name)

The default behavior of this method is to call removeAttribute(String
name) on the wrapped request object.

**Specified by:**

removeAttribute in interface ServletRequest

**Parameters:**

name - a String specifying the name of the attribute to remove

---

```java
public java.util.Locale getLocale()
getLocale()

getLocale
```

**getLocale**

**getLocal**

```java
public Locale getLocale()
```

The default behavior of this method is to return getLocale() on the wrapped request object.

**Specified by:**

getLocale in interface ServletRequest

**Returns:**

the preferred Locale for the client

---

```java
public java.util Enumeration<E> getLocales()
getLocales()

getLocales
```

**getLocales**

**getLocales**

```java
public Enumeration getLocales()
```

The default behavior of this method is to return getLocales() on the wrapped request object.

**Specified by:**

getLocales in interface ServletRequest
Returns:
an Enumeration of preferred Locale objects for the client

public boolean isSecure()
isSecure()  

isSecure  

The default behavior of this method is to return isSecure() on the wrapped request object.

Specified by:
   isSecure in interface ServletRequest
Returns:
a boolean indicating if the request was made using a secure channel

public RequestDispatcher getRequestDispatcher(String path)
getRequestDispatcher(String path)  

getRequestDispatcher  

The default behavior of this method is to return getRequestDispatcher(String path) on the wrapped request object.

Specified by:
   getRequestDispatcher in interface ServletRequest
Parameters:
   path - a String specifying the pathname to the resource. If it is
relative, it must be relative against the current servlet.  

**Returns:**

A `RequestDispatcher` object that acts as a wrapper for the resource at the specified path, or `null` if the servlet container cannot return a `RequestDispatcher`.

See Also:

- `RequestDispatcher`
- `ServletContext.getRequestDispatcher(java.lang.String)`

---

```java
public String getRealPath(String path)
getRealPath(String path)
```

**getRealPath**

```java
public String getRealPath(String path)
```

The default behavior of this method is to return `getRealPath(String path)` on the wrapped request object.

Specified by:

- `getRealPath` in interface `ServletRequest`

---

```java
public int getRemotePort()
getRemotePort()
```

**getRemotePort**

```java
public int getRemotePort()
```

The default behavior of this method is to return `getRemotePort()` on the wrapped request object.

Specified by:
getRemotePort in interface ServletRequest

Returns:
an integer specifying the port number

Since:
2.4

public String getLocalName()
getLocalName()
since 2.4

getLocalName

public String getLocalName()

The default behavior of this method is to return getLocalName() on the wrapped request object.

Specified by:
getLocalName in interface ServletRequest

Returns:
a String containing the host name of the IP on which the request was received.

Since:
2.4

public String getLocalAddr()
getLocalAddr()
since 2.4

getLocalAddr

public String getLocalAddr()

The default behavior of this method is to return getLocalAddr() on
the wrapped request object.

**Specified by:**
getLocalAddr in interface ServletRequest

**Returns:**
a String containing the IP address on which the request was received.

**Since:**
2.4

---

```java
public int getLocalPort()
getLocalPort()
since 2.4
```

**getLocalPort**

public int getLocalPort()

The default behavior of this method is to return getLocalPort() on the wrapped request object.

**Specified by:**
getLocalPort in interface ServletRequest

**Returns:**
an integer specifying the port number

**Since:**
2.4

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.servlet  Interface ServletResponse

All Known Subinterfaces:
  HttpServletResponse

All Known Implementing Classes:
  HttpServletResponseWrapper, ServletResponseWrapper

public interface ServletResponse

Implemented by: HttpServletResponse, ServletResponseWrapper

Servlet servlet ServletResponse servlet

MIME  #getOutputStream  ServletOutputStream
#getWriter  PrintWriter  multipart
ServletOutputStream

  #setCharacterEncoding  #setContentType  MIME  charset
#setLocale  charset  ISO-8859-1  setCharacterEncoding setContentType  setLocale  getWriter

MIME  Internet  RFC  RFC 2045  SMTP  HTTP  MIME

See also  javax.servlet.ServletOutputStream

Defines an object to assist a servlet in sending a response to the client. The servlet container creates a ServletResponse object and passes it as an argument to the servlet's service method.

To send binary data in a MIME body response, use the ServletOutputStream returned by getOutputStream(). To send character data, use the PrintWriter object returned by getWriter(). To mix binary and text data, for example, to create a multipart response, use a
ServletOutputStream and manage the character sections manually.

The charset for the MIME body response can be specified explicitly using the `setCharacterEncoding(java.lang.String)` and `setContentType(java.lang.String)` methods, or implicitly using the `setLocale(java.util.Locale)` method. Explicit specifications take precedence over implicit specifications. If no charset is specified, ISO-8859-1 will be used. The `setCharacterEncoding`, `setContentType`, or `setLocale` method must be called before `getWriter` and before committing the response for the character encoding to be used.

See the Internet RFCs such as RFC 2045 for more information on MIME. Protocols such as SMTP and HTTP define profiles of MIME, and those standards are still evolving.

**Author:**
Various

**See Also:**
ServletOutputStream

---

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>flushBuffer()</code></td>
<td>Forces any content in the buffer to be written to the client.</td>
</tr>
<tr>
<td><code>getBufferSize()</code></td>
<td>Returns the actual buffer size used for the response.</td>
</tr>
<tr>
<td><code>getCharacterEncoding()</code></td>
<td>Returns the name of the character encoding (MIME charset) used for the body sent in this response.</td>
</tr>
<tr>
<td><code>getContentType()</code></td>
<td>Returns the content type used for the MIME body sent in this response.</td>
</tr>
<tr>
<td><code>getLocale()</code></td>
<td>Returns the locale specified for this response.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getOutputStream()</code></td>
<td>Returns a <code>ServletOutputStream</code> suitable for writing binary data in the response.</td>
</tr>
<tr>
<td><code>getWriter()</code></td>
<td>Returns a <code>PrintWriter</code> object that can send character text to the client.</td>
</tr>
<tr>
<td><code>isCommitted()</code></td>
<td>Returns a boolean indicating if the response has been committed.</td>
</tr>
<tr>
<td><code>reset()</code></td>
<td>Clears any data that exists in the buffer as well as the status code and headers.</td>
</tr>
<tr>
<td><code>resetBuffer()</code></td>
<td>Clears the content of the underlying buffer in the response without clearing headers or status code.</td>
</tr>
<tr>
<td><code>setBufferSize(int size)</code></td>
<td>Sets the preferred buffer size for the body of the response.</td>
</tr>
<tr>
<td><code>setCharacterEncoding(String charset)</code></td>
<td>Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8.</td>
</tr>
<tr>
<td><code>setContentLength(int len)</code></td>
<td>Sets the length of the content body in the response In HTTP servlets, this method sets the HTTP Content-Length header.</td>
</tr>
<tr>
<td><code>setContentType(String type)</code></td>
<td>Sets the content type of the response being sent to the client, if the response has not been committed yet.</td>
</tr>
<tr>
<td><code>setLocale(Locale loc)</code></td>
<td>Sets the locale of the response, if the response has not been committed yet.</td>
</tr>
</tbody>
</table>
public String getCharacterEncoding()
(MIME charset)

Returns the name of the character encoding (MIME charset) used for the body sent in this response. The character encoding may have been specified explicitly using the setCharacterEncoding(java.lang.String) or setContentType(java.lang.String) methods, or implicitly using the setLocale(java.util.Locale) method. Explicit specifications take precedence over implicit specifications. Calls made to these methods after getWriter has been called or after the response has been committed have no effect on the character encoding. If no character encoding has been specified, ISO-8859-1 is returned.

See RFC 2047 (http://www.ietf.org/rfc/rfc2047.txt) for more information about character encoding and MIME.

Returns:

a String specifying the name of the character encoding, for example, UTF-8

public String getContentType()
MIME

getCharacterEncoding()  getWriter()  charset
charset
return
since
String text/html; charset=UTF-8 null
2.4

getContentType

String getContentType()

Returns the content type used for the MIME body sent in this response. The content type proper must have been specified using setContentType(java.lang.String) before the response is committed. If no content type has been specified, this method returns null. If a content type has been specified, and a character encoding has been explicitly or implicitly specified as described in getCharacterEncoding() or getWriter() has been called, the charset parameter is included in the string returned. If no character encoding has been specified, the charset parameter is omitted.

Returns:

a String specifying the content type, for example, text/html; charset=UTF-8, or null

Since:

2.4

public ServletOutputStream getOutputStream() throws java.io.IOException

ServletOutputStream flush()

return ServletOutputStream

Throws

IllegalStateException: getWriter

Throws

java.io.IOException:
getOutputStream

`ServletOutputStream getOutputStream()`

Throws `IOException`

Returns a `ServletOutputStream` suitable for writing binary data in the response. The servlet container does not encode the binary data.

Calling `flush()` on the `ServletOutputStream` commits the response. Either this method or `getWriter()` may be called to write the body, not both.

Returns:
a `ServletOutputStream` for writing binary data

Throws:

- `IllegalStateException` - if the `getWriter` method has been called on this response
- `IOException` - if an input or output exception occurred

See Also:
- `getWriter()`

---

public java.io.PrintWriter `getWriter()` throws java.io.IOException

```
#get
getCharacterEncoding
ISO-8859-1
getOutputStream
ISO-8859-1

PrintWriter flush()
```

```
#getOutputStream
return PrintWriter
Throws UnsupportedEncodingException: getCharacterEncoding
Throws IllegalStateException: getOutputStream
```
Throws: java.io.IOException:
See also: getOutputStream, setCharacterEncoding

getWriter

`getWriter()`

Throws: IOException

`getWriter()`

Returns a PrintWriter object that can send character text to the client. The PrintWriter uses the character encoding returned by `getCharacterEncoding()`. If the response's character encoding has not been specified as described in `getCharacterEncoding` (i.e., the method just returns the default value ISO-8859-1), `getWriter` updates it to ISO-8859-1.

Calling flush() on the PrintWriter commits the response.

Either this method or `getOutputStream()` may be called to write the body, not both.

Returns:
a PrintWriter object that can return character data to the client

Throws:
UnsupportedEncodingException - if the character encoding returned by `getCharacterEncoding` cannot be used
`IllegalStateException` - if the `getOutputStream` method has already been called for this response object
`IOException` - if an input or output exception occurred

See Also:
`getOutputStream()`, `setCharacterEncoding(java.lang.String)`

---

public void setCharacterEncoding(String charset)
(MIME charset) UTF-8

`#setLocale` text/html String `#setContentType`

UTF-8 String text/html; charset=UTF-8 String
-setContentType

-getWriter

-servlet writer HTTP Content-Type
-servlet HTTP servlet writer

-setCharacterEncoding

-void setCharacterEncoding(String charset)

Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8. If the character encoding has already been set by setContentType(java.lang.String) or setLocale(java.util.Locale), this method overrides it. Calling setContentType(java.lang.String) with the String of text/html and calling this method with the String of UTF-8 is equivalent with calling setContentType with the String of text/html; charset=UTF-8.

This method can be called repeatedly to change the character encoding. This method has no effect if it is called after getWriter has been called or after the response has been committed.

Containers must communicate the character encoding used for the servlet response’s writer to the client if the protocol provides a way for doing so. In the case of HTTP, the character encoding is communicated as part of the Content-Type header for text media types. Note that the character encoding cannot be communicated via HTTP headers if the servlet does not specify a content type; however, it is still used to encode text written via the servlet
response's writer.

**Parameters:**

- **charset** - a String specifying only the character set defined by IANA Character Sets
  
  (http://www.iana.org/assignments/character-sets)

**Since:**

2.4

**See Also:**

`#setLocale`

```java
public void setContentLength(int len)
```

**HTTP servlet**

**HTTP Content-Length**

| len | Content-Length |

**setContentLength**

```java
void setContentLength(int len)
```

Sets the length of the content body in the response. In HTTP servlets, this method sets the HTTP Content-Length header.

**Parameters:**

- **len** - an integer specifying the length of the content being returned to the client; sets the Content-Length header

---

```java
public void setContentType(String type)
```

**getWriter**

**servlet**

**writer**

**HTTP**

<table>
<thead>
<tr>
<th>type</th>
<th>MIME</th>
</tr>
</thead>
</table>

**Content-Type**

| String |

**See also**

`setLocale`, `setCharacterEncoding`, `getOutputStream`, `getWriter`
**setContentType**

*void setContentType(String type)*

Sets the content type of the response being sent to the client, if the response has not been committed yet. The given content type may include a character encoding specification, for example, `text/html;charset=UTF-8`. The response's character encoding is only set from the given content type if this method is called before `getWriter` is called.

This method may be called repeatedly to change content type and character encoding. This method has no effect if called after the response has been committed. It does not set the response's character encoding if it is called after `getWriter` has been called or after the response has been committed.

Containers must communicate the content type and the character encoding used for the servlet response's writer to the client if the protocol provides a way for doing so. In the case of HTTP, the `Content-Type` header is used.

**Parameters:**
- *type* - a `String` specifying the MIME type of the content

**See Also:**
- `setLocale(java.util.Locale)`, `setCharacterEncoding(java.lang.String), getOutputStream()`, `getWriter()`

---

**public void setSize(int size)**

*javax.servlet.*

**size**

*IllegalStateException*
Throws: IllegalArgumentException
See also: getBufferSize, flushBuffer, isCommitted, reset

**setBufferSize**

```java
void setBufferSize(int size)
```

Sets the preferred buffer size for the body of the response. The servlet container will use a buffer at least as large as the size requested. The actual buffer size used can be found using getBufferSize.

A larger buffer allows more content to be written before anything is actually sent, thus providing the servlet with more time to set appropriate status codes and headers. A smaller buffer decreases server memory load and allows the client to start receiving data more quickly.

This method must be called before any response body content is written; if content has been written or the response object has been committed, this method throws an IllegalArgumentException.

**Parameters:**
- `size` - the preferred buffer size

**Throws:**
- `IllegalArgumentException` - if this method is called after content has been written

**See Also:**
- `getBufferSize()`, `flushBuffer()`, `isCommitted()`, `reset()`

---

```java
public int getBufferSize()
```

```java
0
```

```java
return setBufferSize, flushBuffer, isCommitted, reset
```

See also: setBufferSize, flushBuffer, isCommitted, reset
**getBufferSize**

```java
int getBufferSize()
```

Returns the actual buffer size used for the response. If no buffering is used, this method returns 0.

**Returns:**
the actual buffer size used

**See Also:**
`setBufferSize(int), flushBuffer(), isCommitted(), reset()`

---

**public void flushBuffer() throws java.io.IOException**

See also `setBufferSize, getBufferSize, isCommitted, reset`

---

**flushBuffer**

```java
void flushBuffer() throws IOException
```

Forces any content in the buffer to be written to the client. A call to this method automatically commits the response, meaning the status code and headers will be written.

**Throws:**
`IOException`

**See Also:**
`setBufferSize(int), getBufferSize(), isCommitted(), reset()`

---

**public void resetBuffer()**

since 2.3

See also `setBufferSize, getBufferSize, isCommitted, reset`
resetBuffer

void resetBuffer()

Clears the content of the underlying buffer in the response without clearing headers or status code. If the response has been committed, this method throws an IllegalStateException.

Since:
2.3
See Also:
setBufferSize(int), getBufferSize(), isCommitted(), reset()

public boolean isCommitted()

boolean return

See also setBufferSize, getBufferSize, flushBuffer, reset

isCommitted

boolean isCommitted()

Returns a boolean indicating if the response has been committed. A committed response has already had its status code and headers written.

Returns:
  a boolean indicating if the response has been committed
See Also:
  setBufferSize(int), getBufferSize(), flushBuffer(), reset()

public void reset()

throws

IllegalStateException:

See also setBufferSize, getBufferSize, flushBuffer, isCommitted
reset

void reset()

Clears any data that exists in the buffer as well as the status code and headers. If the response has been committed, this method throws an IllegalStateException.

Throws:

- IllegalStateException - if the response has already been committed

See Also:

- setBufferSize(int), getBufferSize(), flushBuffer(), isCommitted()

public void setLocale(java.util.Locale loc)

getWriter locale-encoding-mapping-list

charset #setContentType
#setCharacterEncoding getWriter

servlet writer HTTP Content-Language
Content-Type servlet HTTP

servlet writer loc

See also getLocale, setContentType, setCharacterEncoding

setLocale

void setLocale(Locale loc)
Sets the locale of the response, if the response has not been committed yet. It also sets the response's character encoding appropriately for the locale, if the character encoding has not been explicitly set using `setContentType(java.lang.String)` or `setCharacterEncoding(java.lang.String)`, `getWriter` hasn't been called yet, and the response hasn't been committed yet. If the deployment descriptor contains a `locale-encoding-mapping-list` element, and that element provides a mapping for the given locale, that mapping is used. Otherwise, the mapping from locale to character encoding is container dependent.

This method may be called repeatedly to change locale and character encoding. The method has no effect if called after the response has been committed. It does not set the response's character encoding if it is called after `setContentType(java.lang.String)` has been called with a charset specification, after `setCharacterEncoding(java.lang.String)` has been called, after `getWriter` has been called, or after the response has been committed.

Containers must communicate the locale and the character encoding used for the servlet response's writer to the client if the protocol provides a way for doing so. In the case of HTTP, the locale is communicated via the `Content-Language` header, the character encoding as part of the `Content-Type` header for text media types. Note that the character encoding cannot be communicated via HTTP headers if the servlet does not specify a content type; however, it is still used to encode text written via the servlet response's writer.

**Parameters:**

- `loc` - the locale of the response

**See Also:**

`getLocale()`, `setContentType(java.lang.String)`,
`setCharacterEncoding(java.lang.String)`

```java
public java.util.Locale getLocale()
#setLocale

setLocale

See also

setLocale
```
getLocale

locale getLocale()

Returns the locale specified for this response using the
setLocale(java.util.Locale) method. Calls made to setLocale after
the response is committed have no effect. If no locale has been
specified, the container's default locale is returned.

See Also:
setLocale(java.util.Locale)
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
Class ServletResponseWrapper

javax.servlet
java.lang.Object
  ↓ javax.servlet.ServletResponseWrapper

All Implemented Interfaces:
  ServletResponse

Direct Known Subclasses:
  HttpServletResponseWrapper

public class ServletResponseWrapper
extends Object
implements ServletResponse

Implements: ServletResponse
Extended by: HttpServletResponseWrapper

ServletResponse  Servlet  Wrapper  Decorator

since  v 2.3  
See also  javax.servlet.ServletResponse

Provides a convenient implementation of the ServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet. This class implements the Wrapper or Decorator pattern. Methods default to calling through to the wrapped response object.

Since:  
v 2.3
Author:  
  Various
See Also:  
ServletResponse
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServletResponseWrapper(ServletResponse response)</td>
<td>Creates a ServletResponse adaptor wrapping the given response object.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void flushBuffer()</td>
<td>The default behavior of this method is to call flushBuffer() on the wrapped response object.</td>
</tr>
<tr>
<td>int getBufferSize()</td>
<td>The default behavior of this method is to return getBufferSize() on the wrapped response object.</td>
</tr>
<tr>
<td>String getCharacterEncoding()</td>
<td>The default behavior of this method is to return getCharacterEncoding() on the wrapped response object.</td>
</tr>
<tr>
<td>String getContentType()</td>
<td>The default behavior of this method is to return getContentType() on the wrapped response object.</td>
</tr>
<tr>
<td>Locale getLocale()</td>
<td>The default behavior of this method is to return getLocale() on the wrapped response object.</td>
</tr>
<tr>
<td>ServletOutputStream getOutputStream()</td>
<td>The default behavior of this method is to return getOutputStream() on the wrapped response object.</td>
</tr>
<tr>
<td>ServletResponse getResponse()</td>
<td>Return the wrapped ServletResponse object.</td>
</tr>
<tr>
<td>PrintWriter getWriter()</td>
<td>The default behavior of this method is to return getWriter() on the wrapped response object.</td>
</tr>
<tr>
<td>boolean isCommitted()</td>
<td>The default behavior of this method is to return isCommitted() on the wrapped response object.</td>
</tr>
<tr>
<td>reset()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>void</td>
<td>The default behavior of this method is to call reset() on the wrapped response object.</td>
</tr>
<tr>
<td>resetBuffer()</td>
<td>The default behavior of this method is to call resetBuffer() on the wrapped response object.</td>
</tr>
<tr>
<td>setBufferSize(int size)</td>
<td>The default behavior of this method is to call setBufferSize(int size) on the wrapped response object.</td>
</tr>
<tr>
<td>setCharacterEncoding(String charset)</td>
<td>The default behavior of this method is to call setCharacterEncoding(String charset) on the wrapped response object.</td>
</tr>
<tr>
<td>setContentLength(int len)</td>
<td>The default behavior of this method is to call setContentLength(int len) on the wrapped response object.</td>
</tr>
<tr>
<td>setContentType(String type)</td>
<td>The default behavior of this method is to call setContentType(String type) on the wrapped response object.</td>
</tr>
<tr>
<td>setLocale(Locale loc)</td>
<td>The default behavior of this method is to call setLocale(Locale loc) on the wrapped response object.</td>
</tr>
<tr>
<td>setResponse(ServletResponse response)</td>
<td>Sets the response being wrapped.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public ServletResponseWrapper(ServletResponse response)
ServletResponse

Throws IllegalArgumentException: response null

ServletResponseWrapper

public ServletResponseWrapper(ServletResponse response)

Creates a ServletResponse adaptor wrapping the given response object.

Throws: IllegalArgumentException - if the response is null.

Method Detail

public ServletResponse getResponse()

ServletResponse

getResponse

public ServletResponse getResponse()

Return the wrapped ServletResponse object.

public void setResponse(ServletResponse response)

Throws IllegalArgumentException: response null

setResponse

public void setResponse(ServletResponse response)

Sets the response being wrapped.
Throws:

**IllegalArgumentException** - if the response is null.

```java
public void setCharacterEncoding(String charset)
```

since 2.4

**setCharacterEncoding**

```java
public void setCharacterEncoding(String charset)
```

The default behavior of this method is to call `setCharacterEncoding(String charset)` on the wrapped response object.

**Specified by:**

`setCharacterEncoding` in interface `ServletResponse`

**Parameters:**

- `charset` - a String specifying only the character set defined by IANA Character Sets
  (http://www.iana.org/assignments/character-sets)

**Since:**

2.4

**See Also:**

`#setLocale`

---

```java
public String getCharacterEncoding()
```

```java
getCharacterEncoding()
```

**getCharacterEncoding**

```java
public String getCharacterEncoding()
```

The default behavior of this method is to return
getCharacterEncoding() on the wrapped response object.

Specified by:
- getCharacterEncoding in interface ServletResponse

Returns:
- a String specifying the name of the character encoding, for example, UTF-8

public ServletOutputStream getOutputStream() throws java.io.IOException

getOutputStream()

getOutputStream

public ServletOutputStream getOutputStream() throws IOException

The default behavior of this method is to return getOutputStream() on the wrapped response object.

Specified by:
- getOutputStream in interface ServletResponse

Returns:
- a ServletOutputStream for writing binary data

Throws:
- IOException - if an input or output exception occurred

See Also:
- ServletResponse.getWriter()

public java.io.PrintWriter getWriter() throws java.io.IOException

getWriter()
public PrintWriter getWriter() throws IOException

The default behavior of this method is to return getWriter() on the wrapped response object.

Specified by:
   getWriter in interface ServletResponse

Returns:
a PrintWriter object that can return character data to the client

Throws:
   IOException - if an input or output exception occurred

See Also:
   ServletResponse.getOutputStream(), ServletResponse.setCharacterEncoding(java.lang.String)

---

public void setContentLength(int len)

setContentLength

public void setContentLength(int len)

The default behavior of this method is to call setContentLength(int len) on the wrapped response object.

Specified by:
   setContentLength in interface ServletResponse

Parameters:
   len - an integer specifying the length of the content being returned to the client; sets the Content-Length header

---

public void setContentType(String type)

setContentType(String type)
**setContentType**

public void **setContentType**(String type)

The default behavior of this method is to call setContentType(String type) on the wrapped response object.

**Specified by:**
`setContentType` in interface `ServletResponse`

**Parameters:**
type - a `String` specifying the MIME type of the content

**See Also:**

---

**public String getContentType()**

`getContentType()`

since 2.4

---

**getContentType**

public `String` `getContentType()`

The default behavior of this method is to return getContentType() on the wrapped response object.

**Specified by:**
`getContentType` in interface `ServletResponse`

**Returns:**
a `String` specifying the content type, for example, `text/html; charset=UTF-8`, or null

**Since:**
2.4
public void setBufferSize(int size)
setBufferSize(int size)

setBufferSize

public void setBufferSize(int size)

The default behavior of this method is to call setBufferSize(int size) on the wrapped response object.

Specified by:

setBufferSize in interface ServletResponse

Parameters:

size - the preferred buffer size

See Also:

ServletResponse.getBufferSize(), ServletResponse.flushBuffer(), ServletResponse.isCommitted(), ServletResponse.reset()

---------------------------

generateBuffer(int size)

generateBuffer()

generateBuffer

public int generateBuffer(int size)

generateBuffer()

The default behavior of this method is to return generateBuffer(int size) on the wrapped response object.

Specified by:

generateBuffer in interface ServletResponse

Returns:

the actual buffer size used

See Also:

ServletResponse.generateBuffer(int), ServletResponse.flushBuffer(), ServletResponse.isCommitted()
public void flushBuffer() throws java.io.IOException
flushBuffer()

flushBuffer

public void flushBuffer()
throws IOException

The default behavior of this method is to call flushBuffer() on the wrapped response object.

Specified by:
flushBuffer in interface ServletResponse

Throws:
IOException

See Also:
ServletResponse.setBufferSize(int), ServletResponse.getBufferSize(), ServletResponse.isCommitted(), ServletResponse.reset()

public boolean isCommitted()

isCommitted()

isCommitted

public boolean isCommitted()

    The default behavior of this method is to return isCommitted() on the wrapped response object.

Specified by:
isCommitted in interface ServletResponse
Returns:
a boolean indicating if the response has been committed

See Also:
ServletResponse.setBufferSize(int),
ServletResponse.getBufferSize(),
ServletResponse.flushBuffer(), ServletResponse.reset()

public void reset()
    reset()

reset

public void reset()
    The default behavior of this method is to call reset() on the wrapped response object.

Specified by:
reset in interface ServletResponse

See Also:
ServletResponse.setBufferSize(int),
ServletResponse.getBufferSize(),
ServletResponse.flushBuffer(), ServletResponse.isCommitted()

public void resetBuffer()
    resetBuffer()

resetBuffer

public void resetBuffer()
    The default behavior of this method is to call resetBuffer() on the wrapped response object.
public void setLocale(java.util.Locale loc)
setLocale(Locale loc)

setLocale

public void setLocale(Locale loc)

The default behavior of this method is to call setLocale(Locale loc) on the wrapped response object.

Specified by: setLocale in interface ServletResponse

Parameters:
loc - the locale of the response

See Also:
ServletResponse.getLocale(),
ServletResponse.setContentType(java.lang.String),
ServletResponse.setCharacterEncoding(java.lang.String)

public java.util.Locale getLocale()
getLocale()

getLocale

public Locale getLocale()

The default behavior of this method is to return getLocale() on the
wrapped response object.

**Specified by:**
getLocale in interface ServletResponse

**See Also:**
ServletRequest.setLocale(java.util.Locale)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.management.j2ee.statistics Interface ServletStats

All Superinterfaces:
   Stats

public interface ServletStats
extends Stats

Implements: Stats

Servlet

Specifies the statistics provided by a Servlet.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TimeStatistic</td>
<td>get ServiceExceptionTime()</td>
</tr>
<tr>
<td></td>
<td>Execution times for the methods.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics STATS
getStatistic, getStatisticNames, getStatistics

Method Detail

public TimeStatistic getServiceTime()

getServiceTime
TimeStatistic getServiceTime()

Execution times for the methods.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface Session

All Superinterfaces:  
Runnable

All Known Subinterfaces:  
QueueSession, TopicSession, XAQueueSession, XASession, XATopicSession

public interface Session  
extends Runnable

Implements: Runnable
Implemented by: QueueSession, TopicSession, XASession

Session Java (JVM) JMS

•
•
• TemporaryTopics  TemporaryQueues
•
•
•
•
• QueueBrowsers

MessageConsumer  Session  MessageProducer

close
A `Session` object is a single-threaded context for producing and consuming messages. Although it may allocate provider resources outside the Java virtual machine (JVM), it is considered a lightweight JMS object.

A session serves several purposes:

- It is a factory for its message producers and consumers.
- It supplies provider-optimized message factories.
- It is a factory for `TemporaryTopics` and `TemporaryQueues`.
- It provides a way to create `Queue` or `Topic` objects for those clients that need to dynamically manipulate provider-specific destination names.
- It supports a single series of transactions that combine work spanning its producers and consumers into atomic units.
- It defines a serial order for the messages it consumes and the messages it produces.
- It retains messages it consumes until they have been acknowledged.
- It serializes execution of message listeners registered with its message consumers.
- It is a factory for QueueBrowsers.

A session can create and service multiple message producers and consumers.

One typical use is to have a thread block on a synchronous MessageConsumer until a message arrives. The thread may then use one or more of the Session's MessageProducerS.

If a client desires to have one thread produce messages while others consume them, the client should use a separate session for its producing thread.

Once a connection has been started, any session with one or more registered message listeners is dedicated to the thread of control that delivers messages to it. It is erroneous for client code to use this session or any of its constituent objects from another thread of control. The only exception to this rule is the use of the session or connection close method.

It should be easy for most clients to partition their work naturally into sessions. This model allows clients to start simply and incrementally add message processing complexity as their need for concurrency grows.

The close method is the only session method that can be called while some other session method is being executed in another thread.

A session may be specified as transacted. Each transacted session supports a single series of transactions. Each transaction groups a set of message sends and a set of message receives into an atomic unit of work. In effect, transactions organize a session's input message stream and output message stream into series of atomic units. When a transaction commits, its atomic unit of input is acknowledged and its associated atomic unit of output is sent. If a transaction rollback is done, the transaction's sent messages are destroyed and the session's input is automatically recovered.
The content of a transaction's input and output units is simply those messages that have been produced and consumed within the session's current transaction.

A transaction is completed using either its session's commit method or its session's rollback method. The completion of a session's current transaction automatically begins the next. The result is that a transacted session always has a current transaction within which its work is done.

The Java Transaction Service (JTS) or some other transaction monitor may be used to combine a session's transaction with transactions on other resources (databases, other JMS sessions, etc.). Since Java distributed transactions are controlled via the Java Transaction API (JTA), use of the session's commit and rollback methods in this context is prohibited.

The JMS API does not require support for JTA; however, it does define how a provider supplies this support.

Although it is also possible for a JMS client to handle distributed transactions directly, it is unlikely that many JMS clients will do this. Support for JTA in the JMS API is targeted at systems vendors who will be integrating the JMS API into their application server products.

**Version:**
1.1 February 2, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
QueueSession, TopicSession, XASession

---

### Field Summary

<table>
<thead>
<tr>
<th>static int AUTO_ACKNOWLEDGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>With this acknowledgment mode, the session automatically acknowledges a client's receipt of a message either when the session has successfully returned from a call to receive or when the message listener the session has called to...</td>
</tr>
</tbody>
</table>
process the message successfully returns.

<table>
<thead>
<tr>
<th>static int</th>
<th>CLIENT_ACKNOWLEDGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>With this acknowledgment mode, the client acknowledges a consumed message by calling the message's acknowledge method.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>DUPS_OK_ACKNOWLEDGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>This acknowledgment mode instructs the session to lazily acknowledge the delivery of messages.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>SESSION_TRANSACTED</th>
</tr>
</thead>
<tbody>
<tr>
<td>This value is returned from the method getAcknowledgeMode if the session is transacted.</td>
<td></td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>close()</code></td>
<td>Closes the session.</td>
</tr>
<tr>
<td><code>commit()</code></td>
<td>Commits all messages done in this transaction and releases any locks currently held.</td>
</tr>
<tr>
<td><code>createBrowser(Queue queue)</code></td>
<td>Creates a QueueBrowser object to peek at the messages on the specified queue.</td>
</tr>
<tr>
<td><code>createBrowser(Queue queue, String messageSelector)</code></td>
<td>Creates a QueueBrowser object to peek at the messages on the specified queue using a message selector.</td>
</tr>
<tr>
<td><code>createBytesMessage()</code></td>
<td>Creates a BytesMessage object.</td>
</tr>
<tr>
<td><code>createConsumer(Destination destination)</code></td>
<td>Creates a MessageConsumer for the specified destination.</td>
</tr>
<tr>
<td><code>createConsumer(Destination destination, String messageSelector)</code></td>
<td>Creates a MessageConsumer for the specified destination, using a message selector.</td>
</tr>
<tr>
<td>Class</td>
<td>Method</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>MessageConsumer</td>
<td>String messageSelector, boolean NoLocal)</td>
</tr>
<tr>
<td>TopicSubscriber</td>
<td>createDurableSubscriber(Topic topic, String name)</td>
</tr>
<tr>
<td>TopicSubscriber</td>
<td>createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal)</td>
</tr>
<tr>
<td>MapMessage</td>
<td>createMapMessage()</td>
</tr>
<tr>
<td>Message</td>
<td>createMessage()</td>
</tr>
<tr>
<td>ObjectMessage</td>
<td>createObjectMessage()</td>
</tr>
<tr>
<td>ObjectMessage</td>
<td>createObjectMessage(Serializable object)</td>
</tr>
<tr>
<td>MessageProducer</td>
<td>createProducer(Destination destination)</td>
</tr>
<tr>
<td>Queue</td>
<td>createQueue(String queueName)</td>
</tr>
<tr>
<td>StreamMessage</td>
<td>createStreamMessage()</td>
</tr>
<tr>
<td>TemporaryQueue</td>
<td>createTemporaryQueue()</td>
</tr>
<tr>
<td>TemporaryTopic</td>
<td>createTemporaryTopic()</td>
</tr>
<tr>
<td>TextMessage</td>
<td>createTextMessage()</td>
</tr>
<tr>
<td>TextMessage</td>
<td>createTextMessage(String text)</td>
</tr>
<tr>
<td>Topic</td>
<td>createTopic(String topicName)</td>
</tr>
</tbody>
</table>
Creates a topic identity given a Topic name.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>int getAcknowledgeMode()</code></td>
<td>Returns the acknowledgement mode of the session.</td>
</tr>
<tr>
<td><code>MessageListener getMessageListener()</code></td>
<td>Returns the session's distinguished message listener (optional).</td>
</tr>
<tr>
<td><code>boolean getTransacted()</code></td>
<td>Indicates whether the session is in transacted mode.</td>
</tr>
<tr>
<td><code>void recover()</code></td>
<td>Stops message delivery in this session, and restarts message delivery with the oldest unacknowledged message.</td>
</tr>
<tr>
<td><code>void rollback()</code></td>
<td>Rolls back any messages done in this transaction and releases any locks currently held.</td>
</tr>
<tr>
<td><code>void run()</code></td>
<td>Optional operation, intended to be used only by Application Servers, not by ordinary JMS clients.</td>
</tr>
<tr>
<td><code>void setMessageListener(MessageListener listener)</code></td>
<td>Sets the session's distinguished message listener (optional).</td>
</tr>
<tr>
<td><code>void unsubscribe(String name)</code></td>
<td>Unsubscribes a durable subscription that has been created by a client.</td>
</tr>
</tbody>
</table>

**Field Detail**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUTO_ACKNOWLEDGE</td>
<td>static final int</td>
</tr>
</tbody>
</table>

With this acknowledgment mode, the session automatically
acknowledges a client's receipt of a message either when the session has successfully returned from a call to receive or when the message listener the session has called to process the message successfully returns.

See Also:
Constant Field Values

CLIENT_ACKNOWLEDGE

static final int CLIENT_ACKNOWLEDGE

With this acknowledgment mode, the client acknowledges a consumed message by calling the message's acknowledge method. Acknowledging a consumed message acknowledges all messages that the session has consumed.

When client acknowledgment mode is used, a client may build up a large number of unacknowledged messages while attempting to process them. A JMS provider should provide administrators with a way to limit client overrun so that clients are not driven to resource exhaustion and ensuing failure when some resource they are using is temporarily blocked.

See Also:
Message.acknowledge(), Constant Field Values

DUPS_OK_ACKNOWLEDGE

static final int DUPS_OK_ACKNOWLEDGE

This acknowledgment mode instructs the session to lazily acknowledge the delivery of messages. This is likely to result in the delivery of some duplicate messages if the JMS provider fails, so it
should only be used by consumers that can tolerate duplicate messages. Use of this mode can reduce session overhead by minimizing the work the session does to prevent duplicates.

See Also:
   Constant Field Values

SESSION_TRANSACTED

static final int SESSION_TRANSACTED

This value is returned from the method getAcknowledgeMode if the session is transacted. If a Session is transacted, the acknowledgement mode is ignored.

See Also:
   Constant Field Values

Method Detail

public BytesMessage createBytesMessage() throws JMSException

createBytesMessage

BytesMessage createBytesMessage() throws JMSException

Creates a BytesMessage object. A BytesMessage object is used to send a message containing a stream of uninterpreted bytes.

 Throws:
public MapMessage createMapMessage() throws JMSException

Throws: JMSException - if the JMS provider fails to create this message due to some internal error.

createMapMessage

public Message createMessage() throws JMSException

Throws: JMSException - if the JMS provider fails to create this message due to some internal error.

createMessage

Creates a Message object. The Message interface is the root interface of all JMS messages. A Message object holds all the standard
message header information. It can be sent when a message containing only header information is sufficient.

**Throws:**

- `JMSException` - if the JMS provider fails to create this message due to some internal error.

```java
public ObjectMessage createObjectMessage() throws JMSException
```

Creates an `ObjectMessage` object. An `ObjectMessage` object is used to send a message that contains a serializable Java object.

**Throws:**

- `JMSException` - if the JMS provider fails to create this message due to some internal error.

```java
ObjectMessage createObjectMessage(java.io.Serializable object) throws JMSException
```

```java
public ObjectMessage createObjectMessage(java.io.Serializable object) throws JMSException
```
ObjectMessage createObjectMessage(Serializable object) throws JMSException

Creates an initialized ObjectMessage object. An ObjectMessage object is used to send a message that contains a serializable Java object.

Parameters:
  object - the object to use to initialize this message

Throws:
  JMSException - if the JMS provider fails to create this message due to some internal error.

public StreamMessage createStreamMessage() throws JMSException

StreamMessage StreamMessage Java
  Throws JMSException: JMS

createStreamMessage

StreamMessage createStreamMessage() throws JMSException

Creates a StreamMessage object. A StreamMessage object is used to send a self-defining stream of primitive values in the Java programming language.

Throws:
  JMSException - if the JMS provider fails to create this message due to some internal error.

public TextMessage createTextMessage() throws JMSException

TextMessage TextMessage String
  Throws JMSException: JMS
createTextMessage

TextMessage createTextMessage() throws JMSException

Creates a TextMessage object. A TextMessage object is used to send a message containing a String object.

Throws:
   JMSException - if the JMS provider fails to create this message due to some internal error.

public TextMessage createTextMessage(String text) throws JMSException

TextMessage createTextMessage(String text)

Throws JMSException: JMS

createTextMessage

TextMessage createTextMessage(String text) throws JMSException

Creates an initialized TextMessage object. A TextMessage object is used to send a message containing a String.

Parameters:
   text - the string used to initialize this message

Throws:
   JMSException - if the JMS provider fails to create this message due to some internal error.

public boolean getTransacted() throws JMSException

return true
getTransacted

boolean getTransacted() throws JMSException

Indicates whether the session is in transacted mode.

Returns:
true if the session is in transacted mode

Throws:
JMSException - if the JMS provider fails to return the transaction mode due to some internal error.

public int getAcknowledgeMode() throws JMSException

return SESSION_TRANSACTED

Throws JMSException: JMS

since 1.1

See also createSession

getAcknowledgeMode

int getAcknowledgeMode() throws JMSException

Returns the acknowledgement mode of the session. The acknowledgement mode is set at the time that the session is created. If the session is transacted, the acknowledgement mode is ignored.

Returns:
If the session is not transacted, returns the current acknowledgement mode for the session. If the session is transacted, returns SESSION_TRANSACTED.
Throws:  
   JMSException - if the JMS provider fails to return the acknowledgment mode due to some internal error.

Since:  
   1.1

See Also:  
   Connection.createSession(boolean, int)

---

public void commit() throws JMSException

Throws   
   JMSException: JMS

Throws   
   TransactionRolledBackException:

Throws   
   IllegalStateException:

commit

void commit()  
   throws JMSException

 Commits all messages done in this transaction and releases any locks currently held.

Throws:  
   JMSException - if the JMS provider fails to commit the transaction due to some internal error.
   TransactionRolledBackException - if the transaction is rolled back due to some internal error during commit.
   IllegalStateException - if the method is not called by a transacted session.

---

public void rollback() throws JMSException

Throws   
   JMSException: JMS

Throws   
   IllegalStateException:
rollback

```java
void rollback() throws JMSException
```

Rolls back any messages done in this transaction and releases any locks currently held.

**Throws:**

- `JMSException` - if the JMS provider fails to roll back the transaction due to some internal error.
- `IllegalStateException` - if the method is not called by a transacted session.

---

public void close() throws JMSException

---

**JVM**

```java
receive receive null
```

**Session**

```java
Session JMSException.IllegalStateException
```

**Throws**

- `JMSException`: JMS

---

**close**

```java
void close() throws JMSException
```
Closes the session.

Since a provider may allocate some resources on behalf of a session outside the JVM, clients should close the resources when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

There is no need to close the producers and consumers of a closed session.

This call will block until a receive call or message listener in progress has completed. A blocked message consumer receive call returns null when this session is closed.

Closing a transacted session must roll back the transaction in progress.

This method is the only Session method that can be called concurrently.

Invoking any other Session method on a closed session must throw a JMSExcetion.IllegalArgumentException. Closing a closed session must not throw an exception.

**Throws:**
- JMSExcetion - if the JMS provider fails to close the session due to some internal error.

```java
public void recover() throws JMSExcetion
```

- 
- “”
throws **JMSException**: JMS

Throws **IllegalStateException**: 

```java
public MessageListener getMessageListener() throws JMSEException
```

**recover**

```java
void recover() throws JMSEException
```

Stops message delivery in this session, and restarts message delivery with the oldest unacknowledged message.

All consumers deliver messages in a serial order. Acknowledging a received message automatically acknowledges all messages that have been delivered to the client.

Restarting a session causes it to take the following actions:

- Stop message delivery
- Mark all messages that might have been delivered but not acknowledged as "redelivered"
- Restart the delivery sequence including all unacknowledged messages that had been previously delivered. Redelivered messages do not have to be delivered in exactly their original delivery order.

**Throws:**

- **JMSException** - if the JMS provider fails to stop and restart message delivery due to some internal error.
- **IllegalStateException** - if the method is called by a transacted session.
Thrown by

- JMSException: JMS

See also

- setMessageListener, javax.jms.ServerSessionPool,
  javax.jms.ServerSession

getMessageListener

```java
public MessageListener getMessageListener() throws JMSException
```

Returns the session's distinguished message listener (optional).

**Returns:**

the message listener associated with this session

**Throws:**

JMSException - if the JMS provider fails to get the message listener due to an internal error.

See Also:

- setMessageListener(javax.jms.MessageListener),
  ServerSessionPool, ServerSession

public void setMessageListener(MessageListener listener) throws JMSException

JMS

```java
public void setMessageListener(MessageListener listener) throws JMSException
```

Thrown by

- JMSException: JMS

See also

- getMessageListener, javax.jms.ServerSessionPool,
  javax.jms.ServerSession
**setMessageListener**

```java
void setMessageListener(MessageListener listener)
throws JMSException
```

Sets the session's distinguished message listener (optional).

When the distinguished message listener is set, no other form of message receipt in the session can be used; however, all forms of sending messages are still supported.

This is an expert facility not used by regular JMS clients.

**Parameters:**
- `listener` - the message listener to associate with this session

**Throws:**
- `JMSException` - if the JMS provider fails to set the message listener due to an internal error.

**See Also:**
- `getMessageListener()`, `ServerSessionPool`, `ServerSession`

---

**public void run()**

**Application Server JMS**

See also `javax.jms.ServerSession`

**run**

```java
void run()
```

Optional operation, intended to be used only by Application Servers, not by ordinary JMS clients.

**Specified by:**
- `run` in interface `Runnable`

**See Also:**
- `ServerSession`
public MessageProducer createProducer(Destination destination) throws JMSException

MessageProducer

<table>
<thead>
<tr>
<th>MessageProducer</th>
<th>Queue</th>
<th>Topic</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>destination</td>
<td></td>
<td></td>
<td>MessageProducer</td>
</tr>
</tbody>
</table>

Throws JMSException: MessageProducer

Throws InvalidDestinationException: 
since 1.1

createProducer

MessageProducer createProducer(Destination destination) throws JMSException

Creates a MessageProducer to send messages to the specified destination.

A client uses a MessageProducer object to send messages to a destination. Since Queue and Topic both inherit from Destination, they can be used in the destination parameter to create a MessageProducer object.

Parameters:

destination - the Destination to send to, or null if this is a producer which does not have a specified destination.

Throws:

JMSException - if the session fails to create a MessageProducer due to some internal error.

InvalidDestinationException - if an invalid destination is specified.

Since:

1.1
public MessageConsumer createConsumer(Destination destination) throws JMSException

    MessageConsumer    Queue    Topic    Destination
    destination        MessageConsumer

    Throws: JMSException:
    Throws: InvalidDestinationException:
    since            1.1

createConsumer

public MessageConsumer createConsumer(Destination destination, String messageSelector) throws JMSException

    MessageConsumer    Queue    Topic
    destination        MessageConsumer

    MessageConsumer

    Parameters:
    destination - the Destination to access.

    Throws:
    JMSException - if the session fails to create a consumer due to some internal error.
    InvalidDestinationException - if an invalid destination is specified.

    Since:
    1.1
createConsumer

MessageConsumer createConsumer(Destination destination, String messageSelector) throws JMSException

Creates a MessageConsumer for the specified destination, using a message selector. Since Queue and Topic both inherit from Destination, they can be used in the destination parameter to create a MessageConsumer.

A client uses a MessageConsumer object to receive messages that have been sent to a destination.

Parameters:
- destination - the Destination to access
- messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.

Throws:
- JMSException - if the session fails to create a MessageConsumer due to some internal error.
- InvalidDestinationException - if an invalid destination is specified.
- InvalidSelectorException - if the message selector is invalid.

Since:
1.1
public MessageConsumer createConsumer(Destination destination, String messageSelector, boolean NoLocal) throws JMSException

MessageConsumer

Queue Topic Destination destination
MessageConsumer

MessageConsumer

NoLocal
destination
messageSelector
false
true
null

Throws JMSException: MessageConsumer
Throws InvalidDestinationException:
Throws InvalidSelectorException:
since 1.1

createConsumer

MessageConsumer createConsumer(Destination destination, String messageSelector, boolean NoLocal) throws JMSException

Creates MessageConsumer for the specified destination, using a message selector. This method can specify whether messages published by its own connection should be delivered to it, if the destination is a topic.

Since Queue and Topic both inherit from Destination, they can be used in the destination parameter to create a MessageConsumer.

A client uses a MessageConsumer object to receive messages that have been published to a destination.
In some cases, a connection may both publish and subscribe to a topic. The consumer NoLocal attribute allows a consumer to inhibit the delivery of messages published by its own connection. The default value for this attribute is False. The noLocal value must be supported by destinations that are topics.

**Parameters:**
- **destination** - the Destination to access
- **messageSelector** - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- **NoLocal** - if true, and the destination is a topic, inhibits the delivery of messages published by its own connection. The behavior for NoLocal is not specified if the destination is a queue.

**Throws:**
- **JMSException** - if the session fails to create a MessageConsumer due to some internal error.
- **InvalidDestinationException** - if an invalid destination is specified.
- **InvalidSelectorException** - if the message selector is invalid.

**Since:**
- 1.1

```java
public Queue createQueue(String queueName) throws JMSException

Queue
```

**JMS API**

- `queueName` - Queue
- `return` - Queue
- `Throws` - JMSException
- `since` - 1.1
createQueue

```java
Queue createQueue(String queueName)
  throws JMSException
```

Creates a queue identity given a queue name.

This facility is provided for the rare cases where clients need to dynamically manipulate queue identity. It allows the creation of a queue identity with a provider-specific name. Clients that depend on this ability are not portable.

Note that this method is not for creating the physical queue. The physical creation of queues is an administrative task and is not to be initiated by the JMS API. The one exception is the creation of temporary queues, which is accomplished with the createTemporaryQueue method.

**Parameters:**
- `queueName` - the name of this Queue

**Returns:**
- a `Queue` with the given name

**Throws:**
- `JMSException` - if the session fails to create a queue due to some internal error.

**Since:**
- 1.1

---

public `Topic` createTopic(String topicName) throws `JMSException`

`Topic`

---

**JMS API**

`createTemporaryTopic`
createTopic

```java
topicName

return Topic

Throws JMSException:

since 1.1
```

Creates a topic identity given a Topic name.

This facility is provided for the rare cases where clients need to dynamically manipulate topic identity. This allows the creation of a topic identity with a provider-specific name. Clients that depend on this ability are not portable.

Note that this method is not for creating the physical topic. The physical creation of topics is an administrative task and is not to be initiated by the JMS API. The one exception is the creation of temporary topics, which is accomplished with the createTemporaryTopic method.

Parameters:
- `topicName` - the name of this Topic

Returns:
- a Topic with the given name

Throws:
- JMSException - if the session fails to create a topic due to some internal error.

Since:
- 1.1

---

```java
public TopicSubscriber createDurableSubscriber(Topic topic, String name) throws JMSException
```
createDurableSubscriber

```java
TopicSubscriber createDurableSubscriber(Topics topic, String name)
throws JMSException
```

Creates a durable subscriber to the specified topic.

If a client needs to receive all the messages published on a topic, including the ones published while the subscriber is inactive, it uses a durable `TopicSubscriber`. The JMS provider retains a record of this durable subscription and insures that all messages from the topic's publishers are retained until they are acknowledged by this durable subscriber or they have expired.

Sessions with durable subscribers must always provide the same client identifier. In addition, each client must specify a name that uniquely identifies (within client identifier) each durable subscription it creates. Only one session at a time can have a `TopicSubscriber` for a particular durable subscription.
A client can change an existing durable subscription by creating a durable TopicSubscriber with the same name and a new topic and/or message selector. Changing a durable subscriber is equivalent to unsubscribing (deleting) the old one and creating a new one.

In some cases, a connection may both publish and subscribe to a topic. The subscriber NoLocal attribute allows a subscriber to inhibit the delivery of messages published by its own connection. The default value for this attribute is false.

**Parameters:**
- `topic` - the non-temporary Topic to subscribe to
- `name` - the name used to identify this subscription

**Throws:**
- `JMSException` - if the session fails to create a subscriber due to some internal error.
- `InvalidDestinationException` - if an invalid topic is specified.

**Since:**
1.1

```java
public TopicSubscriber createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException
```

```
TopicSubscriber JMS

/ TopicSubscriber

  topic
  name
  messageSelector null
  noLocal

  Throws JMSException:
  Throws InvalidDestinationException:
```
createDurableSubscriber

**TopicSubscriber createDurableSubscriber***(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException**

Creates a durable subscriber to the specified topic, using a message selector and specifying whether messages published by its own connection should be delivered to it.

If a client needs to receive all the messages published on a topic, including the ones published while the subscriber is inactive, it uses a durable `TopicSubscriber`. The JMS provider retains a record of this durable subscription and insures that all messages from the topic's publishers are retained until they are acknowledged by this durable subscriber or they have expired.

Sessions with durable subscribers must always provide the same client identifier. In addition, each client must specify a name which uniquely identifies (within client identifier) each durable subscription it creates. Only one session at a time can have a `TopicSubscriber` for a particular durable subscription. An inactive durable subscriber is one that exists but does not currently have a message consumer associated with it.

A client can change an existing durable subscription by creating a durable `TopicSubscriber` with the same name and a new topic and/or message selector. Changing a durable subscriber is equivalent to unsubscribing (deleting) the old one and creating a new one.

**Parameters:**
- `topic` - the non-temporary `Topic` to subscribe to
- `name` - the name used to identify this subscription
messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.

noLocal - if set, inhibits the delivery of messages published by its own connection

Throws:

- JMSEException - if the session fails to create a subscriber due to some internal error.
- InvalidDestinationException - if an invalid topic is specified.
- InvalidSelectorException - if the message selector is invalid.

Since:

1.1

public QueueBrowser createBrowser(Queue queue) throws JMSException

QueueBrowser
queue
queue
Throws
 JMSEException:
Throws
InvalidDestinationException:
since
1.1

createBrowser

QueueBrowser createBrowser(Queue queue)
throws JMSEException

Creates a QueueBrowser object to peek at the messages on the specified queue.

Parameters:

queue - the queue to access

Throws:

- JMSEException - if the session fails to create a browser due to some internal error.
- InvalidDestinationException - if an invalid destination is
public QueueBrowser createBrowser(Queue queue, String messageSelector) throws JMSException

QueueBrowser
queue            queue
messageSelector  null

Throws
JMSException:
InvalidDestinationException:
InvalidSelectorException:
since       1.1

createBrowser

QueueBrowser createBrowser(Queue queue,
String messageSelector)
throws JMSException

Creates a QueueBrowser object to peek at the messages on the specified queue using a message selector.

Parameters:
queue - the queue to access
messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.

Throws:
JMSException - if the session fails to create a browser due to some internal error.
InvalidDestinationException - if an invalid destination is specified
InvalidSelectorException - if the message selector is invalid.

Since:
public TemporaryQueue createTemporaryQueue() throws JMSException

TemporaryQueue return
Throws JMSException: since 1.1

createTemporaryQueue

TemporaryQueue createTemporaryQueue() throws JMSException

Creates a TemporaryQueue object. Its lifetime will be that of the Connection unless it is deleted earlier.

Returns: a temporary queue identity
Throws: JMSException - if the session fails to create a temporary queue due to some internal error.
Since: 1.1

public TemporaryTopic createTemporaryTopic() throws JMSException

TemporaryTopic return
Throws JMSException: since 1.1

createTemporaryTopic
**TemporaryTopic** `createTemporaryTopic()` throws `JMSException`  

Creates a `TemporaryTopic` object. Its lifetime will be that of the `Connection` unless it is deleted earlier.

**Returns:**
- a temporary topic identity

**Throws:**
- `JMSException` - if the session fails to create a temporary topic due to some internal error.

**Since:**
- 1.1

---

```java
public void unsubscribe(String name) throws JMSException
```

**MessageConsumer** `TopicSubscriber`  

**name**

**Throws** `JMSException`:

**Throws** `InvalidDestinationException`:

**since** 1.1

**unsubscribe**

```java
void unsubscribe(String name)
```

Throws `JMSException`

Unsubscribes a durable subscription that has been created by a client.

This method deletes the state being maintained on behalf of the
subscriber by its provider.

It is erroneous for a client to delete a durable subscription while there is an active MessageConsumer or TopicSubscriber for the subscription, or while a consumed message is part of a pending transaction or has not been acknowledged in the session.

**Parameters:**
- name - the name used to identify this subscription

**Throws:**
- JMSException - if the session fails to unsubscribe to the durable subscription due to some internal error.
- InvalidDestinationException - if an invalid subscription name is specified.

**Since:**
- 1.1
<table>
<thead>
<tr>
<th>SUMMAR Y: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.mail  **Class Session**

java.lang.Object  
  
  javax.mail.Session

---

public final class **Session**

extends **Object**

---

Session Session API

Session   StoreTransport

- javamail.providers  javamail.default.providers
- javamail.address.map  javamail.default.address.map

javamail.X

1. java.home/lib/javamail.X
2. META-INF/javamail.X
3. META-INF/javamail.default.X

javamail.default.X  JavaMail  mail.jar

ClassLoader  getResource  getResource  CLASSPATH

JDK 1.1  CLASSPATH  1  SecurityManager
J2SE 1.2  1

IMAP  IMAP JavaMail API

javamail.providers  javamail.default.providers

store transport"" store transport "=" - - (",")

---
### META-INF/javamail.default.providers

<table>
<thead>
<tr>
<th>protocol</th>
<th>Transport</th>
</tr>
</thead>
<tbody>
<tr>
<td>smtp</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>type</th>
<th>store</th>
<th>transport</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>class</th>
<th>vendor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sun Microsystems, Inc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>version</th>
<th>1.76, 07/05/04</th>
</tr>
</thead>
</table>

#### javamail.address.map  javamail.default.address.map

```java
javax.mail.Address getType java.util.zip.ZipOutputStream
javax.mail.Address
javax.mail.internet.InternetAddress getType "rfc822"
news nntp Transport
javamail.address.map

rfc822=smtp
news=nntp
```

The Session class represents a mail session and is not subclassed. It collects together properties and defaults used by the mail API's. A single default session can be shared by multiple applications on the desktop. Unshared sessions can also be created.

The Session class provides access to the protocol providers that implement the Store, Transport, and related classes. The protocol providers are configured using the following files:

- javamail.providers and javamail.default.providers
- javamail.address.map and javamail.default.address.map
Each javamail.X resource file is searched for using three methods in the following order:

1. java.home/lib/javamail.X
2. META-INF/javamail.X
3. META-INF/javamail.default.X

The first method allows the user to include their own version of the resource file by placing it in the lib directory where the java.home property points. The second method allows an application that uses the JavaMail APIs to include their own resource files in their application's or jar file's META-INF directory. The javamail.default.X default files are part of the JavaMail mail.jar file.

File location depends upon how the ClassLoader method getResource is implemented. Usually, the getResource method searches through CLASSPATH until it finds the requested file and then stops. JDK 1.1 has a limitation that the number of files of each name that will be found in the CLASSPATH is limited to one. However, this only affects method two, above; method one is loaded from a specific location (if allowed by the SecurityManager) and method three uses a different name to ensure that the default resource file is always loaded successfully. J2SE 1.2 and later are not limited to one file of a given name.

The ordering of entries in the resource files matters. If multiple entries exist, the first entries take precedence over the later entries. For example, the first IMAP provider found will be set as the default IMAP implementation until explicitly changed by the application. The user- or system-supplied resource files augment, they do not override, the default files included with the JavaMail APIs. This means that all entries in all files loaded will be available.

javamail.providers and javamail.default.providers

These resource files specify the stores and transports that are available on the system, allowing an application to "discover" what store and transport implementations are available. The protocol implementations are listed one per line. The file format defines four attributes that describe a protocol implementation. Each attribute is an "="-separated name-value
pair with the name in lowercase. Each name-value pair is semi-colon (";") separated. The following names are defined.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>Name assigned to protocol. For example, smtp for Transport.</td>
</tr>
<tr>
<td>type</td>
<td>Valid entries are store and transport.</td>
</tr>
<tr>
<td>class</td>
<td>Class name that implements this protocol.</td>
</tr>
<tr>
<td>vendor</td>
<td>Optional string identifying the vendor.</td>
</tr>
<tr>
<td>version</td>
<td>Optional string identifying the version.</td>
</tr>
</tbody>
</table>

Here's an example of META-INF/javamail.default.providers file contents:

```
protocol=imap; type=store; class=com.sun.mail.imap.IMAPStore; vendor=Sun Microsystems, Inc.;
protocol=smtp; type=transport; class=com.sun.mail.smtp.SMTPTransport;
```

`javamail.address.map` and `javamail.default.address.map`

These resource files map transport address types to the transport protocol. The `getType` method of `javax.mail.Address` returns the address type. The `javamail.address.map` file maps the transport type to the protocol. The file format is a series of name-value pairs. Each key name should correspond to an address type that is currently installed on the system; there should also be an entry for each `javax.mail.Address` implementation that is present if it is to be used. For example, the `javax.mail.internet.InternetAddress` method `getType` returns "rfc822". Each referenced protocol should be installed on the system. For the case of `news`, below, the client should install a Transport provider supporting the nntp protocol.

Here are the typical contents of a `javamail.address.map` file:

```
rfc822=smtp
news=nntp
```

**Version:**

1.76, 07/05/04
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addProvider(Provider provider)</td>
<td>Add a provider to the session.</td>
</tr>
<tr>
<td>boolean getDebug()</td>
<td>Get the debug setting for this Session.</td>
</tr>
<tr>
<td>PrintStream getDebugOut()</td>
<td>Returns the stream to be used for debugging output.</td>
</tr>
<tr>
<td>static Session getDefaultInstance(Properties props)</td>
<td>Get the default Session object.</td>
</tr>
<tr>
<td>static Session getDefaultInstance(Properties props, Authenticator authenticator)</td>
<td>Get the default Session object.</td>
</tr>
<tr>
<td>Folder getFolder(URLName url)</td>
<td>Get a closed Folder object for the given URLName.</td>
</tr>
<tr>
<td>static Session getInstance(Properties props)</td>
<td>Get a new Session object.</td>
</tr>
<tr>
<td>static Session getInstance(Properties props, Authenticator authenticator)</td>
<td>Get a new Session object.</td>
</tr>
<tr>
<td>PasswordAuthentication getPasswordAuthentication(URLName url)</td>
<td>Return any saved PasswordAuthentication for this (store or transport) URLName.</td>
</tr>
<tr>
<td>Properties getProperties()</td>
<td>Returns the Properties object associated with this Session</td>
</tr>
<tr>
<td>String getProperty(String name)</td>
<td>Returns the value of the specified property.</td>
</tr>
<tr>
<td>Provider getProvider(String protocol)</td>
<td>Returns the default Provider for the protocol.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>getProviders()</code></td>
<td>This method returns an array of all the implementations installed via the javamail.</td>
</tr>
<tr>
<td><code>getStore()</code></td>
<td>Get a Store object that implements this user's desired Store protocol.</td>
</tr>
<tr>
<td><code>getStore(Provider provider)</code></td>
<td>Get an instance of the store specified by Provider.</td>
</tr>
<tr>
<td><code>getStore(String protocol)</code></td>
<td>Get a Store object that implements the specified protocol.</td>
</tr>
<tr>
<td><code>getStore(URLName url)</code></td>
<td>Get a Store object for the given URLName.</td>
</tr>
<tr>
<td><code>getTransport()</code></td>
<td>Get a Transport object that implements this user's desired Transport protocol.</td>
</tr>
<tr>
<td><code>getTransport(Address address)</code></td>
<td>Get a Transport object that can transport a Message to the specified address type.</td>
</tr>
<tr>
<td><code>getTransport(Provider provider)</code></td>
<td>Get an instance of the transport specified in the Provider.</td>
</tr>
<tr>
<td><code>getTransport(String protocol)</code></td>
<td>Get a Transport object that implements the specified protocol.</td>
</tr>
<tr>
<td><code>getTransport(URLName url)</code></td>
<td>Get a Transport object for the given URLName.</td>
</tr>
<tr>
<td><code>requestPasswordAuthentication(InetAddress addr, int port, String protocol, String prompt, String defaultUserName)</code></td>
<td>Call back to the application to get the needed user name and password.</td>
</tr>
<tr>
<td><code>void setDebug(boolean debug)</code></td>
<td></td>
</tr>
</tbody>
</table>
Set the debug setting for this Session.

```java
void setDebugOut(PrintStream out)
```
Set the stream to be used for debugging output for this session.

```java
void setPasswordAuthentication(URLName url, PasswordAuthentication pw)
```
Save a PasswordAuthentication for this (store or transport) URLName.

```java
void setProtocolForAddress(String addresstype, String protocol)
```
Set the default transport protocol to use for addresses of the specified type.

```java
void setProvider(Provider provider)
```
Set the passed Provider to be the default implementation for the protocol in Provider.prototype overriding any previous values.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Method Detail

public static Session getInstance(java.util.Properties props, Authenticator authenticator)

**Session**

- **props**
  - JavaMail A mail.store.protocol
  - mail.transport.protocol
  - mail.host.mail.user mail.from

- **authenticator**
  - Authenticator

**return**

Authenticator

See also

javax.mail.Authenticator
get Instance

public static Session getInstance(Properties props, Authenticator authenticator)

Get a new Session object.

Parameters:
- props - Properties object that hold relevant properties. It is expected that the client supplies values for the properties listed in Appendix A of the JavaMail spec (particularly mail.store.protocol, mail.transport.protocol, mail.host, mail.user, and mail.from) as the defaults are unlikely to work in all cases.
- authenticator - Authenticator object used to call back to the application when a user name and password is needed.

Returns:
a new Session object

See Also:
- Authenticator

----------

public static Session getInstance(java.util.Properties props)

Session

Properties
- props - JavaMail A mail.store.protocol mail.transport.protocol mail.host mail.user mail.from

return Session

since JavaMail 1.2

en

get Instance

public static Session getInstance(Properties props)

Get a new Session object.

Parameters:
- props - Properties object that hold relevant properties.
It is expected that the client supplies values for the properties listed in Appendix A of the JavaMail spec (particularly mail.store.protocol, mail.transport.protocol, mail.host, mail.user, and mail.from) as the defaults are unlikely to work in all cases.

Returns:
a new Session object

Since:
JavaMail 1.2

public static Session getDefaultInstance(java.util.Properties props, Authenticator authenticator)

Session Session

Java Authenticator
Authenticator Authenticator ClassLoader

Authenticator null null

Session Properties Session
Properties getInstance Session

JDK 1.2 Permission

props Properties Session
JavaMail A mail.store.protocol
mail.transport.protocolmail.hostmail.user mail.from

authenticator Authenticator Session Session
Authenticator Authenticator

return Session

getDefaultInstance
public static Session getDefaultInstance(Properties props, Authenticator authenticator)

Get the default Session object. If a default has not yet been setup, a new Session object is created and installed as the default.

Since the default session is potentially available to all code executing in the same Java virtual machine, and the session can contain security sensitive information such as user names and passwords, access to the default session is restricted. The Authenticator object, which must be created by the caller, is used indirectly to check access permission. The Authenticator object passed in when the session is created is compared with the Authenticator object passed in to subsequent requests to get the default session. If both objects are the same, or are from the same ClassLoader, the request is allowed. Otherwise, it is denied.

Note that if the Authenticator object used to create the session is null, anyone can get the default session by passing in null.

Note also that the Properties object is used only the first time this method is called, when a new Session object is created. Subsequent calls return the Session object that was created by the first call, and ignore the passed Properties object. Use the getInstance method to get a new Session object every time the method is called.

In JDK 1.2, additional security Permission objects may be used to control access to the default session.

Parameters:

props - Properties object. Used only if a new Session object is created.
It is expected that the client supplies values for the properties listed in Appendix A of the JavaMail spec (particularly mail.store.protocol, mail.transport.protocol, mail.host, mail.user, and mail.from) as the defaults are unlikely to work in all cases.

authenticator - Authenticator object. Used only if a new Session object is created. Otherwise, it must match the Authenticator used to create the Session.

Returns:
the default Session object

public static Session getDefaultInstance(java.util.Properties props)

Session  Session

Authenticator  Java

Properties  Session

props  JavaMail  A mail.store.protocol

mail.transport.protocolmail.hostmail.user  mail.from

return  Session

since  JavaMail 1.2

getDefaultInstance

public static Session getDefaultInstance(Properties props)

Get the default Session object. If a default has not yet been setup, a new Session object is created and installed as the default.

Note that a default session created with no Authenticator is available to all code executing in the same Java virtual machine, and the session can contain security sensitive information such as user names and passwords.

Parameters:

text

props - Properties object. Used only if a new Session object is created. It is expected that the client supplies values for the properties listed in Appendix A of the JavaMail spec (particularly mail.store.protocol, mail.transport.protocol, mail.host, mail.user, and mail.from) as the defaults are unlikely to work in all cases.

Returns:

text

the default Session object

Since:
public void setDebug(boolean debug)
Session

true mail.debug Session setDebug Session
mail.debug

setDebug

public void setDebug(boolean debug)

Set the debug setting for this Session.

Since the debug setting can be turned on only after the Session has been created, to turn on debugging in the Session constructor, set the property mail.debug in the Properties object passed in to the constructor to true. The value of the mail.debug property is used to initialize the per-Session debugging flag. Subsequent calls to the setDebug method manipulate the per-Session debugging flag and have no affect on the mail.debug property.

Parameters:

debug - Debug setting

public boolean getDebug()
Session

return

getDebug
public boolean getDebug()

Get the debug setting for this Session.

Returns:
   current debug setting

public void setDebugOut(java.io.PrintStream out)

Set the stream to be used for debugging output for this session. If out is null, System.out will be used. Note that debugging output that occurs before any session is created, as a result of setting the mail.debug system property, will always be sent to System.out.

Parameters:
   out - the PrintStream to use for debugging output

Since:
   JavaMail 1.3

public java.io.PrintStream getDebugOut()

Get the stream to be used for debugging output for this session.

Returns:
   current Debug output stream

Since:
   JavaMail 1.3
public PrintStream getDebugOut()

    Returns the stream to be used for debugging output. If no stream has been set, System.out is returned.

    Returns:
        the PrintStream to use for debugging output

    Since:
        JavaMail 1.3

---

public Provider[] getProviders()

javamail.[default.]providers ClassLoader

    return

getProviders

public Provider[] getProviders()

    This method returns an array of all the implementations installed via the javamail.[default.]providers files that can be loaded using the ClassLoader available to this application.

    Returns:
        Array of configured providers

---

public Provider getProvider(String protocol) throws NoSuchProviderException

    Provider mail.<protocol>.class Provider

    Provider NoSuchProviderException

    protocol smtpimap)

    return Provider

    Throws NoSuchProviderException:
getProvider

public Provider getProvider(String protocol) throws NoSuchProviderException

  Returns the default Provider for the protocol specified. Checks mail.<protocol>.class property first and if it exists, returns the Provider associated with this implementation. If it doesn't exist, returns the Provider that appeared first in the configuration files. If an implementation for the protocol isn't found, throws NoSuchProviderException

Parameters:
  protocol - Configured protocol (i.e. smtp, imap, etc)

Returns:
  Currently configured Provider for the specified protocol

Throws:
  NoSuchProviderException - If a provider for the given protocol is not found.

public void setProvider(Provider provider) throws NoSuchProviderException

  Set the passed Provider to be the default implementation for the protocol in Provider.protocol overriding any previous values.

Parameters:
  provider - Currently configured Provider which will be set as the default for the protocol

setProvider
public Store getStore() throws NoSuchProviderException

Get a Store object that implements this user's desired Store protocol. The mail.store.protocol property specifies the desired protocol. If an appropriate Store object is not obtained, NoSuchProviderException is thrown.

Returns:
   a Store object

Throws:
   NoSuchProviderException - If a provider for the given protocol is not found.

public Store getStore(String protocol) throws NoSuchProviderException

Throws:
   NoSuchProviderException - If the provider passed in is invalid.
public Store getStore(String protocol) throws NoSuchProviderException

Get a Store object that implements the specified protocol. If an appropriate Store object cannot be obtained, NoSuchProviderException is thrown.

Parameters:
  protocol -
Returns:
a Store object
Throws:
  NoSuchProviderException - If a provider for the given protocol is not found.

public Store getStore(URLName url) throws NoSuchProviderException

URLName Store Store
NoSuchProviderException URL "scheme" RFC 1738 Store

url
return
Throws NoSuchProviderException: URLName
See also getFolder(URLName), javax.mail.URLName

getStore

public Store getStore(URLName url) throws NoSuchProviderException

Get a Store object for the given URLName. If the requested Store object cannot be obtained, NoSuchProviderException is thrown. The "scheme" part of the URL string (Refer RFC 1738) is used to locate the Store protocol.
Parameters:
  url - URLName that represents the desired Store
Returns:
  a closed Store object
Throws:
  NoSuchProviderException - If a provider for the given URLName is not found.
See Also:
  getFolder(URLName), URLName

public Store getStore(Provider provider) throws NoSuchProviderException

Provider store store
  provider Store Provider
  return Store
  Throws NoSuchProviderException: Provider

getStore

public Store getStore(Provider provider) throws NoSuchProviderException

Get an instance of the store specified by Provider. Instantiates the store and returns it.

Parameters:
  provider - Store Provider that will be instantiated
Returns:
  Instantiated Store
Throws:
  NoSuchProviderException - If a provider for the given Provider is not found.

public Folder getFolder(URLName url) throws MessagingException
getFolder

public Folder getFolder(URLName url)
    throws MessagingException

Get a closed Folder object for the given URLName. If the requested Folder object cannot be obtained, null is returned.

The "scheme" part of the URL string (Refer RFC 1738) is used to locate the Store protocol. The rest of the URL string (that is, the "schemepart", as per RFC 1738) is used by that Store in a protocol dependent manner to locate and instantiate the appropriate Folder object.

Note that RFC 1738 also specifies the syntax for the "schemepart" for IP-based protocols (IMAP4, POP3, etc.). Providers of IP-based mail Stores should implement that syntax for referring to Folders.

Parameters:
    url - URLName that represents the desired folder
Returns:
   Folder

Throws:
   NoSuchProviderException - If a provider for the given URLName is not found.
   MessagingException - if the Folder could not be located or created.

See Also:
   getStore(URLName), URLName

public Transport getTransport() throws NoSuchProviderException

Transport MessagingException
return Transport
Throws NoSuchProviderException:

getTransport

public Transport getTransport() throws NoSuchProviderException

Get a Transport object that implements this user's desired Transport protocol. The mail.transport.protocol property specifies the desired protocol. If an appropriate Transport object cannot be obtained, MessagingException is thrown.

Returns:
   a Transport object

Throws:
   NoSuchProviderException - If the provider is not found.

public Transport getTransport(String protocol) throws NoSuchProviderException

Transport null
public Transport getTransport(String protocol) throws NoSuchProviderException

Get a Transport object that implements the specified protocol. If an appropriate Transport object cannot be obtained, null is returned.

Returns:
   a Transport object

Throws:
   NoSuchProviderException - If provider for the given protocol is not found.

public Transport getTransport(URLName url) throws NoSuchProviderException

See also
t javax.mail.URLName
Transport object cannot be obtained, NoSuchProviderException is thrown. The "scheme" part of the URL string (Refer RFC 1738) is used to locate the Transport protocol.

Parameters:
- url - URLName that represents the desired Transport

Returns:
a closed Transport object

Throws:
- NoSuchProviderException - If a provider for the given URLName is not found.

See Also:
URLName

---

```java
public Transport getTransport(Provider provider) throws NoSuchProviderException {
    Provider transportProvider = provider;
    Transport transport = transportProvider.getTransport();
    return transport;
}
```

getTransport

```java
public Transport getTransport(Provider provider) throws NoSuchProviderException {
    Provider transportProvider = provider;
    Transport transport = transportProvider.getTransport();
    return transport;
}
```

Get an instance of the transport specified in the Provider. Instantiates the transport and returns it.

Parameters:
- provider - Transport Provider that will be instantiated

Returns:
Instantiated Transport

Throws:
- NoSuchProviderException - If provider for the given provider is not found.
public Transport getTransport(Address address) throws NoSuchProviderException

Message Transport

address
return Transport
Throws NoSuchProviderException: Address
See also javax.mail.Address

getTransport

public Transport getTransport(Address address) throws NoSuchProviderException

Get a Transport object that can transport a Message to the specified address type.

Parameters: address
Returns: A Transport object
Throws: NoSuchProviderException - If provider for the Address type is not found
See Also: Address

public void setPasswordAuthentication(URLName url, PasswordAuthentication pw)
store transportURLName PasswordAuthentication pw null URLName
store transport
**setPasswordAuthentication**

```java
public void setPasswordAuthentication(URLName url, PasswordAuthentication pw)
```

Set a PasswordAuthentication for this (store or transport) URLName. If `pw` is null the entry corresponding to the URLName is removed.

This is normally used only by the store or transport implementations to allow authentication information to be shared among multiple uses of a session.

**getPasswordAuthentication**

```java
public PasswordAuthentication getPasswordAuthentication(URLName url)
```

```java
store transport
PasswordAuthentication store transport
return URLName PasswordAuthentication
```

Get any saved PasswordAuthentication for this (store or transport) URLName. Normally used only by store or transport implementations.

**Returns:**

the PasswordAuthentication corresponding to the URLName

**requestPasswordAuthentication**

```java
public PasswordAuthentication requestPasswordAuthentication(java.net.InetAddress addr, int port, String protocol, String prompt, String defaultUserName)
```
Connecting to <protocol> mail service on host <addr>, port <port>.

User Name: <defaultUserName>
Password:

<table>
<thead>
<tr>
<th>addr</th>
<th>InetAddress null</th>
</tr>
</thead>
<tbody>
<tr>
<td>protocol</td>
<td>imappop3</td>
</tr>
<tr>
<td>prompt</td>
<td>String null</td>
</tr>
<tr>
<td>defaultUserName</td>
<td>null</td>
</tr>
<tr>
<td>return</td>
<td>authenticator null</td>
</tr>
</tbody>
</table>

requestPasswordAuthentication

public PasswordAuthentication requestPasswordAuthentication(InetAddress addr, String protocol, String prompt, String defaultUserName)

Call back to the application to get the needed user name and password. The application should put up a dialog something like:

Connecting to <protocol> mail service on host <addr>, port <port>

User Name: <defaultUserName>
Password:

Parameters:
addr - InetAddress of the host. may be null.
protocol - protocol scheme (e.g. imap, pop3, etc.)
prompt - any additional String to show as part of the prompt; may be null.
defaultUserName - the default username. may be null.

Returns:
the authentication which was collected by the authenticator; may be null.
public java.util.Properties getProperties()
Session Properties
      return Properties

getProperties

public Properties getProperties()

      Returns the Properties object associated with this Session

      Returns:
      Properties object

public String getProperty(String name)
null
      return

getProperty

public String getProperty(String name)

      Returns the value of the specified property. Returns null if this
      property does not exist.

      Returns:
      String that is the property value

public void addProvider(Provider provider)

      provider
      since JavaMail 1.4

      en
addProvider

public void addProvider(Provider provider)

Add a provider to the session.

**Parameters:**

- provider - the provider to add

**Since:**

JavaMail 1.4

---

public void setProtocolForAddress(String addresstype, String protocol)

setProtocolForAddress

public void setProtocolForAddress(String addresstype, String protocol)

Set the default transport protocol to use for addresses of the specified type. Normally the default is set by the javamail.default.address.map or javamail.address.map files or resources.

**Parameters:**

- addresstype - type of address
- protocol - name of protocol

**Since:**

JavaMail 1.4

**See Also:**

getTransport(Address)
getTransport(Address)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Interface SessionBean

All Superinterfaces:  
EnterpriseBean, Serializable

public interface SessionBean  
extends EnterpriseBean  
Implements: EnterpriseBean

SessionBean  
Bean  SessionBean  
Bean

The SessionBean interface is implemented by every session enterprise Bean class. The container uses the SessionBean methods to notify the enterprise Bean instances of the instance's life cycle events.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| void ejbActivate()  
The activate method is called when the instance is activated from its "passive" state.  |
| void ejbPassivate()  
The passivate method is called before the instance enters the "passive" state.  |
| void ejbRemove()  
A container invokes this method before it ends the life of the session object.  |
| void setSessionContext(SessionContext ctx)  
Set the associated session context.  |

Method Detail
public void setSessionContext(SessionContext ctx) throws EJBException, java.rmi.RemoteException

Bean

crossed豪 SessionContext

Throws EJBException:
java.rmi.RemoteException: EJB 1.0  EJB 1.1

Throws Bean javax.ejb.EJBException EJB2.0 Bean
javax.ejb.EJBException

setSessionContext

void setSessionContext(SessionContext ctx)
throws EJBException,
RemoteException

Set the associated session context. The container calls this method
after the instance creation.

The enterprise Bean instance should store the reference to the
context object in an instance variable.

This method is called with no transaction context.

Parameters:
ctx - A SessionContext interface for the instance.

Throws:
EJBException - Thrown by the method to indicate a failure
caus ed by a system-level error.
RemoteException - This exception is defined in the method
signature to provide backward compatibility for applications
written for the EJB 1.0 specification. Enterprise beans written for
the EJB 1.1 specification should throw the
javax.ejb.EJBException instead of this exception. Enterprise
Beans written for the EJB2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.

```java
public void ejbRemove() throws EJBException, java.rmi.RemoteException

@throws EJBException
Java.rmi.RemoteException: EJB 1.0 Bean  EJB
@throws 1.1 Bean  javax.ejb.EJBException EJB2.0
Bean  javax.ejb.EJBException

ejbRemove

void ejbRemove()

throws EJBException, RemoteException

A container invokes this method before it ends the life of the session object. This happens as a result of a client's invoking a remove operation, or when a container decides to terminate the session object after a timeout.

This method is called with no transaction context.

Throws:

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the `javax.ejb.EJBException` instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the `javax.ejb.EJBException` instead of this exception.
**ejbActivate**

public void ejbActivate() throws EJBException, java.rmi.RemoteException

“” ejbPassivate()

**Throws**

- **EJBException**:  
  java.rmi.RemoteException: EJB 1.0 Bean EJB

**Throws 1.1**  
  Bean javax.ejb.EJBException EJB 2.0

Bean javax.ejb.EJBException

---

**ejbActivate**

void ejbActivate()

throws EJBException,

RemoteException

The activate method is called when the instance is activated from its "passive" state. The instance should acquire any resource that it has released earlier in the ejbPassivate() method.

This method is called with no transaction context.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB 2.0 and higher specifications must throw the javax.ejb.EJBException instead of this exception.

---

**ejbPassivate**

public void ejbPassivate() throws EJBException,
java.rmi.RemoteException
	"" ejbActivate()

Java

Throws  EJBException:

java.rmi.RemoteException: EJB 1.0 Bean EJB
Throws 1.1 Bean javax.ejb.EJBException EJB2.0
Bean javax.ejb.EJBException

ejbPassivate

void ejbPassivate()
throws EJBException,
RemoteException

The passivate method is called before the instance enters the "passive" state. The instance should release any resources that it can re-acquire later in the ejbActivate() method.

After the passivate method completes, the instance must be in a state that allows the container to use the Java Serialization protocol to externalize and store away the instance's state.

This method is called with no transaction context.

Throws:

EJBException - Thrown by the method to indicate a failure caused by a system-level error.
RemoteException - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB2.0 and higher specifications must throw the javax.ejb.EJBException instead of this exception.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.management.j2ee.statistics Interface SessionBeanStats

All Superinterfaces:
   EJBStats, Stats

All Known Subinterfaces:
   StatefulSessionBeanStats, StatelessSessionBeanStats

public interface SessionBeanStats
extends EJBStats

Implements: EJBStats
Implemented by: StatefulSessionBeanStats, StatelessSessionBeanStats

Bean

Specifies the statistics provided by session beans of both stateful and stateless types.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RangeStatistic</strong></td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.EJBStats
getCreateCount, getRemoveCount

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics
public RangeStatistic getMethodReadyCount()
(method-ready) Bean

ggetMethodReadyCount

RangeStatistic getMethodReadyCount()

Number of beans in the method-ready state.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.ejb Interface SessionContext

All Superinterfaces:
  EJBContext

public interface SessionContext
  extends EJBContext

Implements: EJBContext

SessionContext  Bean  SessionContext

The SessionContext interface provides access to the runtime session context that the container provides for a session enterprise Bean instance. The container passes the SessionContext interface to an instance after the instance has been created. The session context remains associated with the instance for the lifetime of the instance.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getBusinessObject(Class&lt;T&gt; businessInterface)</code></td>
<td>Obtain an object that can be used to invoke the current bean through the given business interface. Obtain a reference to the EJB local object that is associated with the instance. Obtain a reference to the EJB object that is currently associated with the instance. Obtain the business interface through which the current business method invocation was made. Obtain a reference to the JAX-RPC MessageContext.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.ejb.EJBContext

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getCallerIdentity</code>, <code>getCallerPrincipal</code>, <code>getEJBHome</code>, <code>getEJBLocalHome</code>, <code>getEnvironment</code>, <code>getRollbackOnly</code>, <code>getTimerService</code>,</td>
<td></td>
</tr>
</tbody>
</table>
Method Detail

public EJBLocalObject getEJBLocalObject() throws IllegalStateException

EJB

Bean ejbCreate()  ejbRemove()  ejbCreate()  ejbRemove()

return EJB

Throws  IllegalArgumentException:

getEJBLocalObject

EJBLocalObject getEJBLocalObject() throws IllegalStateException

Obtain a reference to the EJB local object that is associated with the instance.

An instance of a session enterprise Bean can call this method at anytime between the ejbCreate() and ejbRemove() methods, including from within the ejbCreate() and ejbRemove() methods.

An instance can use this method, for example, when it wants to pass a reference to itself in a method argument or result.

Returns:  The EJB local object currently associated with the instance.

Throws:
IllegalStateException - Thrown if the instance invokes this method while the instance is in a state that does not allow the instance to invoke this method, or if the instance does not have a local interface.

public EJBObject getEJBObject() throws IllegalStateException

EJB

Bean ejbCreate() ejbRemove() ejbCreate() ejbRemove()

return EJB

Throws IllegalArgumentException:

getEJBObject

EJBObject getEJBObject() throws IllegalStateException

Obtain a reference to the EJB object that is currently associated with the instance.

An instance of a session enterprise Bean can call this method at anytime between the ejbCreate() and ejbRemove() methods, including from within the ejbCreate() and ejbRemove() methods.

An instance can use this method, for example, when it wants to pass a reference to itself in a method argument or result.

Returns:
The EJB object currently associated with the instance.

Throws:
IllegalStateException - Thrown if the instance invokes this
Get a reference to the JAX-RPC MessageContext.

An instance of a stateless session bean can call this method from any business method invoked through its web service endpoint interface.

**Returns:**
- The MessageContext for this web service invocation.

**Throws:**
- `IllegalStateException` - Thrown if this method is invoked while the instance is in a state that does not allow access to this method.
getBusinessObject(Class<T> businessInterface) throws IllegalStateException

Obtain an object that can be used to invoke the current bean through the given business interface.

**Parameters:**
- businessInterface - One of the local business interfaces or remote business interfaces for this session bean.

**Returns:**
- The business object corresponding to the given business interface.

**Throws:**
- IllegalStateException - Thrown if this method is invoked with an invalid business interface for the current bean.

getInvokedBusinessInterface() throws IllegalStateException

**Throws**
- IllegalStateException: Bean

getInvokedBusinessInterface

Class getInvokedBusinessInterface() throws IllegalStateException

Obtain the business interface through which the current business method invocation was made.

**Throws:**
- IllegalStateException - Thrown if this method is called and the bean has not been invoked through a business interface.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Interface SessionSynchronization

public interface SessionSynchronization

SessionSynchronization Bean

Bean Bean

The SessionSynchronization interface allows a session Bean instance to be notified by its container of transaction boundaries.

An session Bean class is not required to implement this interface. A session Bean class should implement this interface only if it wishes to synchronize its state with the transactions.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void afterBegin()</td>
<td>The afterBegin method notifies a session Bean instance that a new transaction has started, and that the subsequent business methods on the instance will be invoked in the context of the transaction.</td>
</tr>
<tr>
<td>void afterCompletion(boolean committed)</td>
<td>The afterCompletion method notifies a session Bean instance that a transaction commit protocol has completed, and tells the instance whether the transaction has been committed or rolled back.</td>
</tr>
<tr>
<td>void beforeCompletion()</td>
<td>The beforeCompletion method notifies a session Bean instance that a transaction is about to be committed.</td>
</tr>
</tbody>
</table>
public void afterBegin() throws EJBException, java.rmi.RemoteException
afterBegin Bean

The afterBegin method notifies a session Bean instance that a new transaction has started, and that the subsequent business methods on the instance will be invoked in the context of the transaction.

The instance can use this method, for example, to read data from a database and cache the data in the instance fields.

This method executes in the proper transaction context.

Throws:
- EJBException - Thrown by the method to indicate a failure caused by a system-level error.
- RemoteException - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 and higher specifications should throw the
javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB 2.0 and higher specifications must not throw the java.rmi.RemoteException.

```java
class Bean {
    public void beforeCompletion() throws EJBException, java.rmi.RemoteException {
        // Method implementation...
    }
}
```

**setRollbackOnly()**

**Throws**

- `EJBException`:
  - java.rmi.RemoteException: EJB 1.0 Bean EJB 1.1 Bean javax.ejb.EJBException EJB 2.0 Bean java.rmi.RemoteException

**beforeCompletion**

```java
class Bean {
    public void beforeCompletion() throws EJBException, RemoteException {
        // Method implementation...
    }
}
```

The `beforeCompletion` method notifies a session Bean instance that a transaction is about to be committed. The instance can use this method, for example, to write any cached data to a database.

This method executes in the proper transaction context.

**Note:** The instance may still cause the container to rollback the transaction by invoking the `setRollbackOnly()` method on the instance context, or by throwing an exception.

**Throws:**

- `EJBException` - Thrown by the method to indicate a failure caused by a system-level error.
- `RemoteException` - This exception is defined in the method
signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 and higher specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB 2.0 and higher specifications must not throw the java.rmi.RemoteException.

```java
public void afterCompletion(boolean committed) throws EJBException, java.rmi.RemoteException

afterCompletion   Bean
```

<table>
<thead>
<tr>
<th>committed</th>
<th>true false</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>EJBException: java.rmi.RemoteException: EJB 1.0 Bean</td>
</tr>
<tr>
<td>Throws</td>
<td>EJB 1.1 Bean javax.ejb.EJBException EJB 2.0 Bean java.rmi.RemoteException</td>
</tr>
</tbody>
</table>

**afterCompletion**

```java
void afterCompletion(boolean committed) throws EJBException, RemoteException
```

The afterCompletion method notifies a session Bean instance that a transaction commit protocol has completed, and tells the instance whether the transaction has been committed or rolled back.

This method executes with no transaction context.

This method executes with no transaction context.

**Parameters:**

- committed: True if the transaction has been committed, false if
is has been rolled back.

**Throws:**

- **EJBException** - Thrown by the method to indicate a failure caused by a system-level error.
- **RemoteException** - This exception is defined in the method signature to provide backward compatibility for enterprise beans written for the EJB 1.0 specification. Enterprise beans written for the EJB 1.1 and higher specification should throw the javax.ejb.EJBException instead of this exception. Enterprise beans written for the EJB 2.0 and higher specifications must not throw the java.rmi.RemoteException.
javax.mail.util  Class SharedByteArrayInputStream

java.lang.Object  
\- java.io.InputStream  
  \- java.io.ByteArrayInputStream  
  \- javax.mail.util.SharedByteArrayInputStream

All Implemented Interfaces:  
Closeable, SharedInputStream

public class SharedByteArrayInputStream
extends ByteArrayInputStream
implements SharedInputStream

Extends: java.io.InputStream > java.io.ByteArrayInputStream  
Implements: SharedInputStream

SharedInputStream  ByteArrayInputStream  reader  byte

version  1.5, 07/05/04
since  JavaMail 1.4

A ByteArrayInputStream that implements the SharedInputStream interface, allowing the underlying byte array to be shared between multiple readers.

Since:  
JavaMail 1.4
Version:  
1.5, 07/05/04
Author:  
Bill Shannon
### Field Summary

<table>
<thead>
<tr>
<th>protected int</th>
<th>start</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Position within shared buffer that this stream starts at.</td>
</tr>
</tbody>
</table>

### Fields inherited from class java.io.ByteArrayInputStream

- buf
- count
- mark
- pos

### Constructor Summary

- **SharedByteArrayInputStream**(byte[] buf)
  - Create a SharedByteArrayInputStream representing the entire byte array.

- **SharedByteArrayInputStream**(byte[] buf, int offset, int length)
  - Create a SharedByteArrayInputStream representing the part of the byte array from offset for length bytes.

### Method Summary

<table>
<thead>
<tr>
<th>long</th>
<th>getPosition()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return the current position in the InputStream, as an offset from the beginning of the InputStream.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>InputStream</th>
<th>newStream(long start, long end)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.io.ByteArrayInputStream

- available, close, mark, markSupported, read, read, reset, skip

### Methods inherited from class java.io.InputStream

- read

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
protected int start

Position within shared buffer that this stream starts at.

public SharedByteArrayInputStream(byte[] buf)
byte SharedByteArrayInputStream

Create a SharedByteArrayInputStream representing the entire byte array.

Parameters:
buf - the byte array

public SharedByteArrayInputStream(byte[] buf, int offset, int length)
byte offset length
SharedByteArrayInputStream


**SharedByteArrayInputStream**

public SharedByteArrayInputStream(byte[] buf,
       int offset,
       int length)

Create a SharedByteArrayInputStream representing the part of the byte array from offset for length bytes.

**Parameters:**
- buf - the byte array
- offset - offset in byte array to first byte to include
- length - number of bytes to include

### Method Detail

**public long getPosition()**

InputStream InputStream

return

**getPosition**

public long getPosition()

Return the current position in the InputStream, as an offset from the beginning of the InputStream.

**Specified by:**

getPosition in interface SharedInputStream

**Returns:**

the current position
newStream

public InputStream newStream(long start, long end)

Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive). start must be non-negative. If end is -1, the new stream ends at the same place as this stream. The returned InputStream will also implement the SharedInputStream interface.

Specified by:
    newStream in interface SharedInputStream

Parameters:
    start - the starting position
    end - the ending position + 1

Returns:
    the new stream
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
<th>FRAMES</th>
<th>NO FRAMES</th>
<th>DETAIL: FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
</table>

---
javax.mail.util Class SharedFileInputStream

java.lang.Object  
   ↓ java.io.InputStream  
      ↓ java.io.FilterInputStream  
         ↓ java.io.BufferedInputStream  
            ↓ javax.mail.util.SharedFileInputStream

All Implemented Interfaces:
   Closeable, SharedInputStream

public class SharedFileInputStream
      extends BufferedInputStream
      implements SharedInputStream

Implements: SharedInputStream

SharedFileInputStream  BufferedInputStream  mark  reset
newStream  RandomAccessFile

SharedFileInputStream  newStream
SharedFileInputStream  SharedInputStream
      since  JavaMail 1.4

A SharedFileInputStream is a BufferedInputStream that buffers data from the file and supports the mark and reset methods. It also supports the newStream method that allows you to create other streams that represent subsets of the file. A RandomAccessFile object is used to access the file data.

Note that when the SharedFileInputStream is closed, all streams created with the newStream method are also closed. This allows the creator of the SharedFileInputStream object to control access to the underlying file and
ensure that it is closed when needed, to avoid leaking file descriptors. Note also that this behavior contradicts the requirements of SharedInputStream and may change in a future release.

Since:
JavaMail 1.4
Author:
Bill Shannon

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>protected long</strong></td>
</tr>
<tr>
<td>The file offset that corresponds to the first byte in the read buffer.</td>
</tr>
<tr>
<td><strong>protected int</strong></td>
</tr>
<tr>
<td>The normal size of the read buffer.</td>
</tr>
<tr>
<td><strong>protected long</strong></td>
</tr>
<tr>
<td>The amount of data in this subset of the file.</td>
</tr>
<tr>
<td><strong>protected RandomAccessFile</strong></td>
</tr>
<tr>
<td>The file containing the data.</td>
</tr>
<tr>
<td><strong>protected long</strong></td>
</tr>
<tr>
<td>The file offset of the start of data in this subset of the file.</td>
</tr>
</tbody>
</table>

Fields inherited from class java.io.BufferedInputStream
buf, count, marklimit, markpos, pos

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SharedFileInputStream(File file)</td>
</tr>
<tr>
<td>Creates a SharedFileInputStream for the file.</td>
</tr>
<tr>
<td>SharedFileInputStream(File file, int size)</td>
</tr>
<tr>
<td>Creates a SharedFileInputStream with the specified buffer size.</td>
</tr>
<tr>
<td>SharedFileInputStream(String file)</td>
</tr>
<tr>
<td>Creates a SharedFileInputStream for the named file</td>
</tr>
</tbody>
</table>
**SharedFileInputStream**(String file, int size)

Creates a SharedFileInputStream with the specified buffer size.

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>int available()</strong></td>
<td>Returns the number of bytes that can be read from this input stream without blocking.</td>
</tr>
<tr>
<td><strong>void close()</strong></td>
<td>Closes this input stream and releases any system resources associated with the stream.</td>
</tr>
<tr>
<td><strong>protected void finalize()</strong></td>
<td>Force this stream to close.</td>
</tr>
<tr>
<td><strong>long getPosition()</strong></td>
<td>Return the current position in the InputStream, as an offset from the beginning of the InputStream.</td>
</tr>
<tr>
<td><strong>void mark(int readlimit)</strong></td>
<td>See the general contract of the mark method of InputStream.</td>
</tr>
<tr>
<td><strong>boolean markSupported()</strong></td>
<td>Tests if this input stream supports the mark and reset methods.</td>
</tr>
<tr>
<td><strong>InputStream newStream(long start, long end)</strong></td>
<td>Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).</td>
</tr>
<tr>
<td><strong>int read()</strong></td>
<td>See the general contract of the read method of InputStream.</td>
</tr>
<tr>
<td><strong>int read(byte[] b, int off, int len)</strong></td>
<td>Reads bytes from this stream into the specified byte array, starting at the given offset.</td>
</tr>
<tr>
<td><strong>void reset()</strong></td>
<td>See the general contract of the reset method of InputStream.</td>
</tr>
<tr>
<td><strong>skip(long n)</strong></td>
<td></td>
</tr>
</tbody>
</table>
See the general contract of the `skip` method of `InputStream`.

**Methods inherited from class `java.io.FilterInputStream`**

- `read`

**Methods inherited from class `java.lang.Object`**

- `clone`, `equals`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

---

**Field Detail**

- `in`

  protected `RandomAccessFile` in

  The file containing the data. Shared by all related `SharedFileInputStreams`.

---

- `bufsize`

  protected `int` `bufsize`

  The normal size of the read buffer.

---

- `bufpos`

  protected `long` `bufpos`
The file offset that corresponds to the first byte in the read buffer.

start

protected long start

The file offset of the start of data in this subset of the file.

datalen

protected long datalen

The amount of data in this subset of the file.

Constructor Detail

public SharedFileInputStream(java.io.File file) throws java.io.IOException
  
  Creates a SharedFileInputStream for the file.

  Parameters:
    file - the file

  Throws:
public SharedFileInputStream(String file) throws IOException

    SharedFileInputStream
    file

SharedFileInputStream

public SharedFileInputStream(String file) throws IOException

    Creates a SharedFileInputStream for the named file

    Parameters:
    file - the file

    Throws:
    IOException

public SharedFileInputStream(java.io.File file, int size) throws java.io.IOException

    SharedFileInputStream
    file
    size

    Throws
    IllegalArgumentException: size <= 0

SharedFileInputStream

public SharedFileInputStream(File file, int size) throws IOException

    Creates a SharedFileInputStream with the specified buffer size.
public SharedFileInputStream(String file, int size) throws IOException

Creates a SharedFileInputStream with the specified buffer size.

Parameters:
   file - the file
   size - the buffer size.

Throws:
   IllegalArgumentException - if size <= 0.
   IOException

Method Detail

public int read() throws java.io.IOException

InputStream read

return -1

Throws java.io.IOException: I/O
read

public int read() throws IOException

See the general contract of the read method of InputStream.

Overrides:
    read in class BufferedInputStream
Returns:
    the next byte of data, or -1 if the end of the stream is reached.
Throws:
    IOException - if an I/O error occurs.

read

public int read(byte[] b, int off, int len) throws java.io.IOException

Reads bytes from this stream into the specified byte array, starting at the given offset.

public int read(byte[] b, int off, int len) throws java.io.IOException

java.io.InputStream read
    b
    off
    len
return

Throws
    java.io.IOException: I/O
This method implements the general contract of the corresponding read method of the InputStream class.

**Overrides:**
read in class BufferedInputStream

**Parameters:**
b - destination buffer.
off - offset at which to start storing bytes.
len - maximum number of bytes to read.

**Returns:**
the number of bytes read, or -1 if the end of the stream has been reached.

**Throws:**
IOException - if an I/O error occurs.

```java
public long skip(long n) throws IOException
```

See the general contract of the skip method of InputStream.

**Overrides:**
skip in class BufferedInputStream

**Parameters:**
n - the number of bytes to be skipped.

**Returns:**
the actual number of bytes skipped.

**Throws:**
IOException - if an I/O error occurs.
public int available() throws java.io.IOException

        return
    Throws          java.io.IOException: I/O

available

public int available() throws IOException

    Returns the number of bytes that can be read from this input stream without blocking.

    Overrides: available in class BufferedInputStream
    Returns: the number of bytes that can be read from this input stream without blocking.
    Throws: IOException - if an I/O error occurs.

public void mark(int readlimit)

    InputStream mark
    readlimit
    See also     reset()

mark

public void mark(int readlimit)

    See the general contract of the mark method of InputStream.

    Overrides: mark in class BufferedInputStream
Parameters:
readlimit - the maximum limit of bytes that can be read before the mark position becomes invalid.

See Also:
reset()

public void reset() throws java.io.IOException
InputStream reset

markpos -1 IOException pos markpos

Throws java.io.IOException: See also mark(int)

reset

public void reset() throws IOException

See the general contract of the reset method of InputStream.

If markpos is -1 (no mark has been set or the mark has been invalidated), an IOException is thrown. Otherwise, pos is set equal to markpos.

Overrides:
reset in class BufferedInputStream

Throws:
IOException - if this stream has not been marked or if the mark has been invalidated.

See Also:
mark(int)

public boolean markSupported()
mark Supported

public boolean markSupported()

Tests if this input stream supports the mark and reset methods. The markSupported method of SharedFileInputStream returns true.

Overrides:

markSupported in class BufferedInputStream

Returns:

a boolean indicating if this stream type supports the mark and reset methods.

See Also:

InputStream.mark(int), InputStream.reset()

public void close() throws java.io.IOException

Throws

java.io.IOException: I/O

close

public void close()

throws IOException

Closes this input stream and releases any system resources associated with the stream.

Specified by:

close in interface Closeable

Overrides:
close in class BufferedInputStream

Throws:
IOException - if an I/O error occurs.

public long getPosition()
InputStream InputStream
return

g getPosition

public long getPosition()

Return the current position in the InputStream, as an offset from the beginning of the InputStream.

Specified by:
getPosition in interface SharedInputStream

Returns:
the current position

public java.io.InputStream newStream(long start, long end)
InputStream InputStream
start
InputStream SharedInputStream
start
end
-1 InputStream
SharedInputStream
start
end
+ 1
return

newStream

public InputStream newStream(long start, long end)
Return a new InputStream representing a subset of the data from this InputStream, starting at `start` (inclusive) up to `end` (exclusive). `start` must be non-negative. If `end` is -1, the new stream ends at the same place as this stream. The returned InputStream will also implement the `SharedInputStream` interface.

**Specified by:**

`newStream` in interface `SharedInputStream`

**Parameters:**

- `start` - the starting position
- `end` - the ending position + 1

**Returns:**

the new stream

```java
protected void finalize() throws Throwable
```

**finalize**

protected void `finalize()` throws `Throwable`

Force this stream to close.

**Overrides:**

`finalize` in class `Object`

**Throws:**

`Throwable`
PS:
<table>
<thead>
<tr>
<th>SUMMARY:</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
<th>DETAIL:</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
An InputStream that is backed by data that can be shared by multiple readers may implement this interface. This allows users of such an InputStream to determine the current position in the InputStream, and to create new InputStreams representing a subset of the data in the original InputStream. The new InputStream will access the same underlying data as the original, without copying the data.

Note that implementations of this interface must ensure that the close method does not close any underlying stream that might be shared by multiple instances of SharedInputStream until all shared instances have been closed.

Since:
JavaMail 1.2

Version:
1.5, 07/05/04

Author:
Bill Shannon

javax.mail.internet Interface SharedInputStream

All Known Implementing Classes:
SharedByteArrayInputStream, SharedFileInputStream

public interface SharedInputStream

Implemented by: SharedByteArrayInputStream, SharedFileInputStream

reader InputStream InputStream InputStream
InputStream InputStream InputStream InputStream

version 1.5, 07/05/04
since JavaMail 1.2

en
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getPosition()</td>
<td>Return the current position in the InputStream, as an offset from the beginning of the InputStream.</td>
</tr>
<tr>
<td>newStream()</td>
<td>Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).</td>
</tr>
</tbody>
</table>

Method Detail

```java
public long getPosition()

InputStream inputStream

return

g getPosition()

long getPosition()

Return the current position in the InputStream, as an offset from the beginning of the InputStream.

Returns: the current position
```

```java
public java.io.InputStream newStream(long start, long end)

InputStream inputStream

start

SharedInputStream

start

end

+1
```
newStream

`InputStream newStream(long start, long end)`

Return a new InputStream representing a subset of the data from this InputStream, starting at `start` (inclusive) up to `end` (exclusive). `start` must be non-negative. If `end` is -1, the new stream ends at the same place as this stream. The returned InputStream will also implement the SharedInputStream interface.

**Parameters:**
- `start` - the starting position
- `end` - the ending position + 1

**Returns:**
- the new stream
javax.resource.spi  Class SharingViolationException

java.lang.Object
  ▼ java.lang.Throwable
    ▼ java.lang.Exception
      ▼ javax.resource.ResourceException
        ▼ javax.resource.spi.SharingViolationException

All Implemented Interfaces:
  Serializable

public class SharingViolationException
  extends ResourceException

Extends: Throwable > Exception > ResourceException

version 1.0

This is thrown to indicate a connection sharing violation.

This may be thrown by a resource adapter when an application uses a shareable connection in an unshareable manner.

Version: 1.0
Author: Ram Jeyaraman
See Also: Serialized Form
Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SharingViolationException()</code></td>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
<tr>
<td><code>SharingViolationException(String message)</code></td>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
<tr>
<td><code>SharingViolationException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and error code.</td>
</tr>
<tr>
<td><code>SharingViolationException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>SharingViolationException(Throwable cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class `javax.resource.ResourceException`
- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`

Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Constructor Detail

public `SharingViolationException()`
null

SharingViolationException

public class SharingViolationException() {
    Constructs a new instance with null as its detail message.
}

public class SharingViolationException(String message) {
    Constructs a new instance with the specified detail message.
    Parameters:
    message - the detail message.
}

public class SharingViolationException(Throwable cause) {
    Constructs a new throwable with the specified cause.
    Parameters:
    cause - a chained exception of type Throwable.
}
public SharingViolationException(String message, Throwable cause)

cause throwable

message

cause Throwable

SharingViolationException

public SharingViolationException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

SharingViolationException

public SharingViolationException(String message, String errorCode)

throwable

message

errorCode

SharingViolationException

public SharingViolationException(String message, String errorCode)

Constructs a new throwable with the specified detail message and error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
<table>
<thead>
<tr>
<th>Summary: Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail: Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS | NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTRUCTOR | METHOD
FRAMES | NO FRAMES
DETAIL: FIELD | CONSTRUCTOR | METHOD
javax.faces.convert Class ShortConverter

java.lang.Object
   javax.faces.convert.ShortConverter

All Implemented Interfaces:
   Converter

public class ShortConverter

extends Object
implements Converter

Implements: Converter

java.lang.Short short )

Converter implementation for java.lang.Short (and short primitive) values.

Field Summary

| static String | CONVERTER_ID                  | The standard converter id for this converter.
| static String | SHORT_ID                      | The message identifier of the FacesMessage to be created if the conversion to Short fails.
| static String | STRING_ID                     | The message identifier of the FacesMessage to be created if the conversion of the Short value to String fails.

Constructor Summary
### ShortConverter()

#### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getAsObject</code></td>
<td><code>(FacesContext context, UIComponent component, String value)</code></td>
<td>Convert the specified string value, which is associated with the specified <code>UIComponent</code>, into a model data object that is appropriate for being stored during the <code>Apply Request Values</code> phase of the request processing lifecycle.</td>
</tr>
<tr>
<td><code>getAsString</code></td>
<td><code>(FacesContext context, UIComponent component, Object value)</code></td>
<td>Convert the specified model object value, which is associated with the specified <code>UIComponent</code>, into a String that is suitable for being included in the response generated during the <code>Render Response</code> phase of the request processing lifecycle.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `java.lang.Object`

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>toString</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
<td></td>
</tr>
</tbody>
</table>

### Field Detail

`CONVERTER_ID`

public static final `String` `CONVERTER_ID`

The standard converter id for this converter.

**See Also:**

`Constant Field Values`
The message identifier of the `FacesMessage` to be created if the conversion to `Short` fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by an example value.
- `{2}` replaced by a `String` whose value is the label of the input component that produced this message.

See Also:

- [Constant Field Values](#)

---

The message identifier of the `FacesMessage` to be created if the conversion of the `Short` value to `String` fails. The message format string for this message may optionally include the following placeholders:

- `{0}` replaced by the unconverted value.
- `{1}` replaced by a `String` whose value is the label of the input component that produced this message.

See Also:

- [Constant Field Values](#)
public ShortConverter()

Method Detail

public Object getAsObject(FacesContext context, UIComponent component, String value)

Throws ConverterException: NullPointerException
Throws NullPointerException: NullPointerException in context component

getAsObject

public Object getAsObject(FacesContext context, UIComponent component, String value)

Description copied from interface: Converter

Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

Specified by:
getAsObject in interface Converter

Parameters:
context - FacesContext for the request being processed
cOMPONENT component - UIComponent with which this model object value is associated
dvalue - String value to be converted (may be null)

Returns:
null if the value to convert is null, otherwise the result of the
public String getAsString(FacesContext context, UIComponent component, Object value)

Throws ConverterException: Nullable

Throws NullPointerException: Nullable

getAsString

public String getAsString(FacesContext context, UIComponent component, Object value)

Description copied from interface: Converter

Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

Specified by: getAsString in interface Converter

Parameters:
context - FacesContext for the request being processed
component - UIComponent with which this model object value is associated
value - Model object value to be converted (may be null)

Returns:
a zero-length String if value is null, otherwise the result of the conversion

Throws:
ConverterException - if conversion cannot be successfully performed
NullPointerException - if context or component is null
performed

NullPointerException - if context or component is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.rpc.holders  Class ShortHolder

java.lang.Object
   ↳ javax.xml.rpc.holders.ShortHolder

All Implemented Interfaces:
   Holder

public final class ShortHolder
   extends Object
   implements Holder

Implements: Holder

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>value</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShortHolder()</td>
</tr>
<tr>
<td>ShortHolder(short myshort)</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
value

public short value

Constructor Detail

public ShortHolder()

ShortHolder

public ShortHolder()

public ShortHolder(short myshort)

ShortHolder

public ShortHolder(short myshort)
PS:
javax.xml.rpc.holders Class ShortWrapperHolder

java.lang.Object
   \ javax.xml.rpc.holders.ShortWrapperHolder

All Implemented Interfaces:
   Holder

public final class ShortWrapperHolder

extends Object
implements Holder

Implements: Holder

Field Summary

| Short | value |

Constructor Summary

| ShortWrapperHolder() |
| ShortWrapperHolder(Short myshort) |

Method Summary

| Methods inherited from class java.lang.Object clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |
value
public Short value

Constructor Detail

public ShortWrapperHolder()

ShortWrapperHolder

public ShortWrapperHolder()

public ShortWrapperHolder(Short myshort)

ShortWrapperHolder

public ShortWrapperHolder(Short myshort)
javax.servlet.jsp.tagext Interface SimpleTag

All Superinterfaces:
   JspTag

All Known Implementing Classes:
   SimpleTagSupport

public interface SimpleTag
   extends JspTag

   Implements: JspTag
   Implemented by: SimpleTagSupport

   doStartTag()  doEndTag()  SimpleTag  doTag()  BodyTag
   setJspBody()  JspFragment  setJspBody()  invoke()

SimpleTag  SimpleTag  SimpleTagSupport

SimpleTag  JSP

1. JSP
2. setJspContext()  setParent()  setParent()
3. setter
4. setJspBody()  JspFragment
5. doTag()
6. doTag()

   since 2.0

See also javax.servlet.jsp.tagext.SimpleTagSupport
Interface for defining Simple Tag Handlers.

Simple Tag Handlers differ from Classic Tag Handlers in that instead of supporting `doStartTag()` and `doEndTag()`, the `SimpleTag` interface provides a simple `doTag()` method, which is called once and only once for any given tag invocation. All tag logic, iteration, body evaluations, etc. are to be performed in this single method. Thus, simple tag handlers have the equivalent power of `BodyTag`, but with a much simpler lifecycle and interface.

To support body content, the `setJspBody()` method is provided. The container invokes the `setJspBody()` method with a `JspFragment` object encapsulating the body of the tag. The tag handler implementation can call `invoke()` on that fragment to evaluate the body as many times as it needs.

A SimpleTag handler must have a public no-args constructor. Most SimpleTag handlers should extend `SimpleTagSupport`.

**Lifecycle**

The following is a non-normative, brief overview of the SimpleTag lifecycle. Refer to the JSP Specification for details.

1. A new tag handler instance is created each time by the container by calling the provided zero-args constructor. Unlike classic tag handlers, simple tag handlers are never cached and reused by the JSP container.
2. The `setJspContext()` and `setParent()` methods are called by the container. The `setParent()` method is only called if the element is nested within another tag invocation.
3. The setters for each attribute defined for this tag are called by the container.
4. If a body exists, the `setJspBody()` method is called by the container to set the body of this tag, as a `JspFragment`. If the action element is empty in the page, this method is not called at all.
5. The `doTag()` method is called by the container. All tag logic, iteration, body evaluations, etc. occur in this method.
6. The `doTag()` method returns and all variables are synchronized.

Since:
JSP 2.0

See Also:
`SimpleTagSupport`

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void doTag()</code></td>
<td>Called by the container to invoke this tag.</td>
</tr>
<tr>
<td><code>JspTag getParent()</code></td>
<td>Returns the parent of this tag, for collaboration purposes.</td>
</tr>
<tr>
<td><code>void setJspBody(JspFragment jspBody)</code></td>
<td>Provides the body of this tag as a JspFragment object, able to be invoked zero or more times by the tag handler.</td>
</tr>
<tr>
<td><code>void setJspContext(JspContext pc)</code></td>
<td>Called by the container to provide this tag handler with the JspContext for this invocation.</td>
</tr>
<tr>
<td><code>void setParent(JspTag parent)</code></td>
<td>Sets the parent of this tag, for collaboration purposes.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void doTag() throws JspException, java.io.IOException
```

`doTag()` JSP AT_BEGIN AT_END
TagExtraInfo TLD

**Throws**

- `JspException`:
- `SkipPageException`: SKIP_PAGE
- `SkipPageException` JSP `SkipPageException` `java.io.IOException`:
doTag

void doTag()
    throws JspException, IOException

Called by the container to invoke this tag. The implementation of this method is provided by the tag library developer, and handles all tag processing, body iteration, etc.

The JSP container will resynchronize any AT_BEGIN and AT_END variables (defined by the associated tag file, TagExtraInfo, or TLD) after the invocation of doTag().

Throws:
    JspException - If an error occurred while processing this tag.
    SkipPageException - If the page that (either directly or indirectly) invoked this tag is to cease evaluation. A Simple Tag Handler generated from a tag file must throw this exception if an invoked Classic Tag Handler returned SKIP_PAGE or if an invoked Simple Tag Handler threw SkipPageException or if an invoked Jsp Fragment threw a SkipPageException.
    IOException - If there was an error writing to the output stream.

public void setParent(JspTag parent)

setParent

void setParent(JspTag parent)
Sets the parent of this tag, for collaboration purposes.

The container invokes this method only if this tag invocation is nested within another tag invocation.

**Parameters:**

parent - the tag that encloses this tag

---

**public JspTag getParent()**

```
return
```

**getParent**

**JspTag getParent()**

Returns the parent of this tag, for collaboration purposes.

**Returns:**

the parent of this tag

---

**public void setJspContext(JspContext pc)**

```
JspContext
pc
```

See also **setPageContext**

**setJspContext**

```
void setJspContext(JspContext pc)
```

Called by the container to provide this tag handler with the `JspContext` for this invocation. An implementation should save this value.
Parameters:
   pc - the page context for this invocation

See Also:
   Tag.setPageContext(javax.servlet.jsp.PageContext)

public void setJspBody(JspFragment jspBody)

JspFragment

JSP
   doTag()

   jspBody

setJspBody

void setJspBody(JspFragment jspBody)

Provides the body of this tag as a JspFragment object, able to be invoked zero or more times by the tag handler.

This method is invoked by the JSP page implementation object prior to doTag(). If the action element is empty in the page, this method is not called at all.

Parameters:
   jspBody - The fragment encapsulating the body of this tag.
PS:
Class SimpleTagSupport

public class SimpleTagSupport

extends Object
implements SimpleTag

Implements: SimpleTag

SimpleTag

SimpleTagSupport SimpleTagSupport  SimpleTag
SimpleTag  getter

since 2.0

A base class for defining tag handlers implementing SimpleTag.

The SimpleTagSupport class is a utility class intended to be used as the base class for new simple tag handlers. The SimpleTagSupport class implements the SimpleTag interface and adds additional convenience methods including getter methods for the properties in SimpleTag.

Since:

JSP 2.0

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SimpleTagSupport()</td>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong> doTag()</td>
<td>Default processing of the tag does nothing.</td>
</tr>
<tr>
<td><strong>static</strong> JspTag findAncestorOfClass(JspTag from, Class&lt;?&gt; klass)</td>
<td>Find the instance of a given class type that is closest to a given instance.</td>
</tr>
<tr>
<td><strong>protected</strong> JspFragment getJspBody()</td>
<td>Returns the body passed in by the container via setJspBody.</td>
</tr>
<tr>
<td><strong>protected</strong> JspContext getJspContext()</td>
<td>Returns the page context passed in by the container via setJspContext.</td>
</tr>
<tr>
<td><strong>JspTag</strong> getParent()</td>
<td>Returns the parent of this tag, for collaboration purposes.</td>
</tr>
<tr>
<td><strong>void</strong> setJspBody(JspFragment jspBody)</td>
<td>Stores the provided JspFragment.</td>
</tr>
<tr>
<td><strong>void</strong> setJspContext(JspContext pc)</td>
<td>Stores the provided JSP context in the private jspContext field.</td>
</tr>
<tr>
<td><strong>void</strong> setParent(JspTag parent)</td>
<td>Sets the parent of this tag, for collaboration purposes.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
- wait

### Constructor Detail

**public SimpleTagSupport()**
SimpleTagSupport

```java
public SimpleTagSupport()
```

Sole constructor. (For invocation by subclass constructors, typically implicit.)

### Method Detail

```java
public void doTag() throws JspException, java.io.IOException
```

Throws:
- `JspException`: JspException
- `SkipPageException`: SKIP_PAGE

`SkipPageException`:
- JSP SkipPageException
- JspException JSP SkipPageException
- java.io.IOException: IOException

See also `doTag()`

### doTag

```java
public void doTag()
```

Throws:
- `JspException`: JspException
- `IOException`: IOException

Default processing of the tag does nothing.

Specified by:
- `doTag` in interface `SimpleTag`

Throws:
- `JspException` - Subclasses can throw JspException to indicate an error occurred while processing this tag.
- `SkipPageException` - If the page that (either directly or indirectly) invoked this tag is to cease evaluation. A Simple Tag Handler generated from a tag file must throw this exception if an invoked Classic Tag Handler returned SKIP_PAGE or if an invoked Simple Tag Handler threw SkipPageException or if an invoked
Jsp Fragment threw a SkipPageException.  
 IOException - Subclasses can throw IOException if there was an error writing to the output stream.

See Also:  
 SimpleTag.doTag()

---

public void setParent(JspTag parent)

parent

setParent

public void setParent(JspTag parent)

Sets the parent of this tag, for collaboration purposes.

The container invokes this method only if this tag invocation is nested within another tag invocation.

Specified by:  
 setParent in interface SimpleTag

Parameters:  
parent - the tag that encloses this tag

---

public JspTag getParent()

return

g getParent

public JspTag getParent()
Returns the parent of this tag, for collaboration purposes.

Specified by:

getParent in interface SimpleTag

Returns:

the parent of this tag

---

**public void setJspContext(JspContext pc)**

**jspContext** JSP

**pc**

Get JspContext

See also

setJspContext

---

**setJspContext**

**public void setJspContext(JspContext pc)**

Stores the provided JSP context in the private jspContext field. Subclasses can access the JspContext via getJspContext().

Specified by:

setJspContext in interface SimpleTag

Parameters:

**pc** - the page context for this invocation

See Also:


---

**protected JspContext getJspContext()**

Get JspContext

---

**getJspContext**

**protected JspContext getJspContext()**
Returns the page context passed in by the container via `setJspContext`.

**Returns:**
the page context for this invocation

---

```java
public void setJspBody(JspFragment jspBody)
JspFragment
jspBody
See also
setJspBody
```

**setJspBody**

```java
public void setJspBody(JspFragment jspBody)
```

Stores the provided JspFragment.

**Specified by:**
`setJspBody` in interface `SimpleTag`

**Parameters:**
jspBody - The fragment encapsulating the body of this tag. If the action element is empty in the page, this method is not called at all.

**See Also:**
`SimpleTag.setJspBody(javax.servlet.jsp.tagext.JspFragment)`

---

```java
protected JspFragment getJspBody()
```

```java
setJspBody
return null
```

**getJspBody**

```java
protected JspFragment getJspBody()
```
Returns the body passed in by the container via setJspBody.

**Returns:**
- the fragment encapsulating the body of this tag, or null if the action element is empty in the page.

```java
final public static JspTag findAncestorWithClass(JspTag from, Class<T> klass)
Tag / SimpleTag getParent
TagAdapter TagAdapter.getAdaptee()
TagAdpater klass TagAdapter
observable observable void
JSP

findAncestorWithClass
observable
from klass JspTag
return
```

Find the instance of a given class type that is closest to a given instance. This method uses the getParent method from the Tag and/or SimpleTag interfaces. This method is used for coordination among cooperating tags.

For every instance of TagAdapter encountered while traversing the ancestors, the tag handler returned by TagAdapter.getAdaptee() -
instead of the TagAdapter itself - is compared to \texttt{klass}. If the tag handler matches, it - and not its TagAdapter - is returned.

The current version of the specification only provides one formal way of indicating the observable type of a tag handler: its tag handler implementation class, described in the tag-class subelement of the tag element. This is extended in an informal manner by allowing the tag library author to indicate in the description subelement an observable type. The type should be a subtype of the tag handler implementation class or void. This additional constraint can be exploited by a specialized container that knows about that specific tag library, as in the case of the JSP standard tag library.

When a tag library author provides information on the observable type of a tag handler, client programmatic code should adhere to that constraint. Specifically, the Class passed to \texttt{findAncestorWithClass} should be a subtype of the observable type.

\textbf{Parameters:}
\begin{itemize}
  \item \texttt{from} - The instance from where to start looking.
  \item \texttt{klass} - The subclass of JspTag or interface to be matched
\end{itemize}

\textbf{Returns:}
the nearest ancestor that implements the interface or is an instance of the class specified

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to \texttt{license terms}.
Interface `SingleThreadModel`

**Deprecated.** As of Java Servlet API 2.4, with no direct replacement.

```java
public interface SingleThreadModel
```

Ensures that servlets handle only one request at a time. This interface has no methods.

If a servlet implements this interface, you are *guaranteed* that no two threads will execute concurrently in the servlet's `service` method. The servlet container can make this guarantee by synchronizing access to a single instance of the servlet, or by maintaining a pool of servlet instances and dispatching each new request to a free servlet.

Note that `SingleThreadModel` does not solve all thread safety issues. For example, session attributes and static variables can still be accessed by multiple requests on multiple threads at the same time, even when `SingleThreadModel` servlets are used. It is recommended that a developer take other means to resolve those issues instead of implementing this interface, such as avoiding the usage of an instance variable or synchronizing the block of the code accessing those resources. This interface is deprecated in Servlet API version 2.4.

**Author:**
Various

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.search  **Class SizeTerm**

java.lang.Object  
    | javax.mail.search.SearchTerm  
    |    | javax.mail.search.ComparisonTerm  
    |    |    | javax.mail.search.IntegerComparisonTerm  
    |    |    | javax.mail.search.SizeTerm

All Implemented Interfaces:  
    Serializable

```
public final class SizeTerm  
extends IntegerComparisonTerm

Extends:  
    searchTerm > comparisonTerm > integerComparisonTerm
```

**Message**

This class implements comparisons for Message sizes.

**Author:**  
    Bill Shannon, John Mani

**See Also:**  
    Serialized Form

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.IntegerComparisonTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>number</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.ComparisonTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>comparison, EQ, GE, GT, LE, LT, NE</td>
</tr>
</tbody>
</table>
**Constructor Summary**

`SizeTerm(int comparison, int size)`

Constructor.

**Method Summary**

bool `equals(Object obj)`

Equality comparison.

bool `match(Message msg)`

The match method.

**Methods inherited from class** javax.mail.search.IntegerComparisonTerm

getComparison, getNumber, hashCode, match

**Methods inherited from class** java.lang.Object

cloned, finalize, getClass, notify, notifyAll, toString, wait, wait

**Constructor Detail**

`public SizeTerm(int comparison, int size)`

comparison

size

Comparison

**SizeTerm**

`public SizeTerm(int comparison, int size)`

Constructor.
Parameters:
- comparison - the Comparison type
- size - the size

Method Detail

public boolean match(Message msg)

    msg Message
    return true false

match

public boolean match(Message msg)

    The match method.

    Specified by:
    match in class SearchResult

    Parameters:
    msg - the size comparator is applied to this Message's size

    Returns:
    true if the size is equal, otherwise false

public boolean equals(Object obj)

equals

public boolean equals(Object obj)

    Equality comparison.

    Overrides:
equals in class IntegerComparisonTerm

PS:
javax.servlet.jsp  **Class SkipPageException**

**java.lang.Object**
  └ **java.lang.Throwable**
      └ **java.lang.Exception**
          └ **javax.servlet.jsp.JspException**
              └ **javax.servlet.jsp.SkipPageException**

**All Implemented Interfaces:**
  [Serializable](https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/lang/Serializable.html)

---

**public class SkipPageException**

extends [JspException](https://docs.oracle.com/en/java/javase/11/docs/api/jdk/api/jdk/jsp/jsp/JspException.html)

**Extends:** Throwable > Exception > [JspException](https://docs.oracle.com/en/java/javase/11/docs/api/jdk/api/jdk/jsp/jsp/JspException.html)

**doEndTag()** Tag.SKIP_PAGE Jsp
JSP - SimpleTag JSP

**since** 2.0


Exception to indicate the calling page must cease evaluation. Thrown by a simple tag handler to indicate that the remainder of the page must not be evaluated. The result is propagated back to the page in the case where one tag invokes another (as can be the case with tag files). The effect is similar to that of a Classic Tag Handler returning Tag.SKIP_PAGE from doEndTag(). Jsp Fragments may also throw this exception. This exception should not be thrown manually in a JSP page or tag file - the behavior is undefined. The exception is intended to be thrown inside SimpleTag handlers and in JSP fragments.

**Since:**
JSP 2.0

**See Also:**
Constructor Summary

SkipPageException()
   Creates a SkipPageException with no message.

SkipPageException(String message)
   Creates a SkipPageException with the provided message.

SkipPageException(String message, Throwable rootCause)
   Creates a SkipPageException with the provided message and root cause.

SkipPageException(Throwable rootCause)
   Creates a SkipPageException with the provided root cause.

Method Summary

Methods inherited from class javax.servlet.jsp.JspException
getRootCause

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public SkipPageException()
SkipPageException
**SkipPageException**

```java
public SkipPageException()

    Creates a SkipPageException with no message.
```

```java
public SkipPageException(String message)

    Creates a SkipPageException with the provided message.

    Parameters:
    message - the detail message
```

```java
public SkipPageException(String message, Throwable rootCause)

    Creates a SkipPageException with the provided message and root
    cause.

    Parameters:
```
public SkipPageException(Throwable rootCause)

SkipPageException

rootCause

SkipPageException

public SkipPageException(Throwable rootCause)

Creates a SkipPageException with the provided root cause.

Parameters:

rootCause - the originating cause of this exception
public interface Slot

Slot RegistryObject RegistryObject (Registry Information Model)

RegistryObject 0 Slot Slot slotType Collection
Slot RegistryObject Slot Slot Slot
Collection Slot CollectionslotType Slot

See also javax.xml.registry.infomodel.ExtensibleObject

Slot instances provide a dynamic way to add arbitrary attributes to RegistryObject instances. This ability to add attributes dynamically to RegistryObject instances enables extensibility within the Registry Information Model.

A RegistryObject may have 0 or more Slots. A slot is composed of a name, a slotType and a collection of values. The name of a slot is locally unique within the RegistryObject instance. Similarly, the value of a Slot is locally unique within a slot instance. Since a Slot represents an extensible attribute whose value may be a collection, a Slot is allowed to have a collection of values rather than a single value. The slotType attribute may optionally specify a type or category for the slot.

Author:
    Farrukh S. Najmi

See Also:
    ExtensibleObject

Field Summary

| ADDRESS_LINES_SLOT |
### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String</code></td>
<td><code>getName()</code></td>
<td>Gets the name for this Slot.</td>
</tr>
<tr>
<td><code>String</code></td>
<td><code>getSlotType()</code></td>
<td>Gets the slotType for this Slot.</td>
</tr>
<tr>
<td><code>Collection</code></td>
<td><code>getValues()</code></td>
<td>Gets the values for this Slot.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setName(String name)</code></td>
<td>Sets the name for this Slot.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setSlotType(String slotType)</code></td>
<td>Sets the slotType for this Slot.</td>
</tr>
<tr>
<td><code>void</code></td>
<td><code>setValues(Collection values)</code></td>
<td>Sets the values for this Slot.</td>
</tr>
</tbody>
</table>

### Field Detail

**SORT_CODE_SLOT**

static final `String` `SORT_CODE_SLOT`
Name for pre-defined Slot used in PostalAddress by JAXR UDDI provider.

See Also:  
 Constant Field Values

ADDRESS_LINES_SLOT

static final String ADDRESS_LINES_SLOT

Name for pre-defined Slot used in PostalAddress by JAXR UDDI provider.

See Also:  
 Constant Field Values

AUTHORIZED_NAME_SLOT

static final String AUTHORIZED_NAME_SLOT

Name for pre-defined Slot used in Organization and ClassificationScheme by JAXR UDDI provider.

See Also:  
 Constant Field Values

OPERATOR_SLOT

static final String OPERATOR SLOT

Name for pre-defined Slot used in Organization and
public String getName() throws JAXRException
Slot NULL

0

return

Throws JAXRException: JAXR

getName

String getName() throws JAXRException

Gets the name for this Slot. Default is a NULL String.

Capability Level: 0

Returns: the name

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setName(String name) throws JAXRException
Slot NULL

0
**setName**

```java
void setName(String name)
    throws JAXRException
```

Sets the name for this Slot. Default is a NULL String.

**Capability Level: 0**

**Parameters:**
- name - the name

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

**public String getSlotType() throws JAXRException**

Slot slotType NULL

```
return String Slot
```

**getSlotType**

```java
String getSlotType()
    throws JAXRException
```

Gets the slotType for this Slot. Default is a NULL String.

**Capability Level: 0**

**Returns:**
the slot type which is an arbitrary String

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

```java
public void setSlotType(String slotType) throws JAXRException

Slot slotType
```

Sets the slotType for this Slot.

**Capability Level:** 0

**Parameters:**
- `slotType` - the slot type which is an arbitrary String

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

```java
public java.util.Collection<E> getValues() throws JAXRException

Slot
```

```java
void setSlotType(String slotType)
throws JAXRException
```

```java
setSlotType
```

- `String slotType` - the slot type which is an arbitrary String

- *JAXRException* - If the JAXR provider encounters an internal error
getValues

Collection getValues() throws JAXRException

Gets the values for this Slot.

Capability Level: 0

Returns:
Collection of String instances representing the values for this Slot. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
String

public void setValues(java.util.Collection<E> values)
throws JAXRException

setValues

void setValues(Collection values)
throws JAXRException
Sets the values for this Slot.

**Capability Level: 0**

**Parameters:**
values - the values for this Slot

**Throws:**
JAXRException - If the JAXR provider encounters an internal error

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.jws.soap  Annotation Type SOAPBinding

@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})
public @interface SOAPBinding

**Implements:** Annotation  
**Inner classes:** SOAPBinding.Style, SOAPBinding.Use, SOAPBinding.ParameterStyle

@Retention(value=RUNTIME)
@Target(value={TYPE, METHOD})

Web Service  SOAP

Specifies the mapping of the Web Service onto the SOAP message protocol.

**Author:**
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

**Optional Element Summary**

<table>
<thead>
<tr>
<th>SOAPBinding.ParameterStyle</th>
<th>parameterStyle</th>
<th>Determines whether method parameters represent the entire message body, or whether the parameters are elements wrapped inside a top-level element named after the operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPBinding.Style</td>
<td>style</td>
<td>Defines the encoding style for messages send to and from the Web Service.</td>
</tr>
<tr>
<td>SOAPBinding.Use</td>
<td>use</td>
<td>Defines the formatting style for messages sent to and from the Web Service.</td>
</tr>
</tbody>
</table>
abstract public SOAPBinding.Style style()
Web Service  Web Service

style

public abstract SOAPBinding.Style style

Defines the encoding style for messages send to and from the Web Service.

Default: DOCUMENT

-----------------------------

abstract public SOAPBinding.Use use()
Web Service  Web Service

use

public abstract SOAPBinding.Use use

Defines the formatting style for messages sent to and from the Web Service.

Default: LITERAL

-----------------------------

abstract public SOAPBinding.ParameterStyle parameterStyle()

parameterStyle

public abstract SOAPBinding.ParameterStyle parameterStyle
Determines whether method parameters represent the entire message body, or whether the parameters are elements wrapped inside a top-level element named after the operation

**Default:**

WRAPPED
javax.xml.ws.soap  Interface SOAPBinding

All Superinterfaces:
    Binding

public interface SOAPBinding
    extends Binding

Implements: Binding

SOAPBinding SOAP
    since JAX-WS 2.0

The SOAPBinding interface is an abstraction for the SOAP binding.

Since:
    JAX-WS 2.0

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>static String</td>
</tr>
</tbody>
</table>
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageFactory.getMessageFactory()</td>
<td>Gets the SAAJ MessageFactory instance used by this SOAP binding.</td>
</tr>
<tr>
<td>Set&lt;String&gt; getRoles()</td>
<td>Gets the roles played by the SOAP binding instance.</td>
</tr>
<tr>
<td>SOAPFactory.getSOAPFactory()</td>
<td>Gets the SAAJ SOAPFactory instance used by this SOAP binding.</td>
</tr>
<tr>
<td>boolean isMTOMEnabled()</td>
<td>Returns true if the use of MTOM is enabled.</td>
</tr>
<tr>
<td>void setMTOMEnabled(boolean flag)</td>
<td>Enables or disables use of MTOM.</td>
</tr>
<tr>
<td>void setRoles(Set&lt;String&gt; roles)</td>
<td>Sets the roles played by the SOAP binding instance.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.ws.Binding
getHandlerChain, setHandlerChain

Field Detail

SOAP11HTTP_BINDING

static final String SOAP11HTTP_BINDING

A constant representing the identity of the SOAP 1.1 over HTTP binding.

See Also:
Constant Field Values
SOAP12HTTP_BINDING

static final String SOAP12HTTP_BINDING

A constant representing the identity of the SOAP 1.2 over HTTP binding.

See Also:
Constant Field Values

SOAP11HTTP_MTOM_BINDING

static final String SOAP11HTTP_MTOM_BINDING

A constant representing the identity of the SOAP 1.1 over HTTP binding with MTOM enabled by default.

See Also:
Constant Field Values

SOAP12HTTP_MTOM_BINDING

static final String SOAP12HTTP_MTOM_BINDING

A constant representing the identity of the SOAP 1.2 over HTTP binding with MTOM enabled by default.

See Also:
Constant Field Values
public java.util.Set<E> getRoles()

return Set

getRoles

Set<String> getRoles()

Gets the roles played by the SOAP binding instance.

Returns:
Set The set of roles played by the binding instance.

setRoles

void setRoles(Set<String> roles)

Sets the roles played by the SOAP binding instance.

Parameters:
roles - The set of roles played by the binding instance.

Throws:
WebServiceException - On an error in the configuration of the list of roles.

public boolean isMTOMEnabled()

return MTOM true

isMTOMEnabled

boolean isMTOMEnabled()
Returns true if the use of MTOM is enabled.

Returns: 
true if and only if the use of MTOM is enabled.

public void setMTOMEnabled(boolean flag)

<table>
<thead>
<tr>
<th>MTOM</th>
<th>flag</th>
<th>MTOM</th>
<th>MTOM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>boolean</td>
</tr>
</tbody>
</table>

Throws WebServiceException:

setMTOMEnabled

void setMTOMEnabled(boolean flag)

Enables or disables use of MTOM.

Parameters:
flag - A boolean specifying whether the use of MTOM should be enabled or disabled.

Throws:
WebServiceException - If the specified setting is not supported by this binding instance.

public SOAPFactory getSOAPFactory()

<table>
<thead>
<tr>
<th>SOAP</th>
<th>SAAJ</th>
<th>SOAPFactory</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>SOAP</td>
<td>SOAPFactory</td>
</tr>
</tbody>
</table>

getSOAPFactory

SOAPFactory getSOAPFactory()

Gets the SAAJ SOAPFactory instance used by this SOAP binding.

Returns:
SOAPFactory instance used by this SOAP binding.

```java
public MessageFactory getMessageFactory()

SOAP SAAJ MessageFactory
return SOAP MessageFactory
```

getMessageFactory

```java
MessageFactory getMessageFactory()

Gets the SAAJ MessageFactory instance used by this SOAP binding.

Returns:

MessageFactory instance used by this SOAP binding.
```
javax.jws.soap  Enum SOAPBinding.ParameterStyle

java.lang.Object
  - java.lang.Enum&lt;SOAPBinding.ParameterStyle&gt;
  - javax.jws.soap.SOAPBinding.ParameterStyle

All Implemented Interfaces:
  Serializable, Comparable&lt;SOAPBinding.ParameterStyle&gt;

Enclosing class:
  SOAPBinding

____________________________

public static enum SOAPBinding.ParameterStyle
  extends Enum&lt;SOAPBinding.ParameterStyle&gt;

Extends: Enum&lt;E&gt;
Contained within: SOAPBinding

SOAP

The style of mapping parameters onto SOAP messages

____________________________

Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BARE</td>
<td></td>
</tr>
<tr>
<td>WRAPPED</td>
<td></td>
</tr>
</tbody>
</table>

____________________________

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static SOAPBinding.ParameterStyle.valueOf(String name)</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
</tbody>
</table>
static \texttt{SOAPBinding.ParameterStyle[]} \texttt{values()} \texttt{values()} \texttt{values()}

Returns an array containing the constants of this enum type, in the order they're declared.

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.\texttt{Enum}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{clone}, \texttt{compareTo}, \texttt{equals}, \texttt{getDeclaringClass}, \texttt{hashCode}, \texttt{name}, \texttt{ordinal}, \texttt{toString}, \texttt{valueOf}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.\texttt{Object}</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{finalize}, \texttt{getClass}, \texttt{notify}, \texttt{notifyAll}, \texttt{wait}, \texttt{wait}, \texttt{wait}</td>
</tr>
</tbody>
</table>

## Enum Constant Detail

**BARE**

public static final \texttt{SOAPBinding.ParameterStyle} \texttt{BARE}

**WRAPPED**

public static final \texttt{SOAPBinding.ParameterStyle} \texttt{WRAPPED}

## Method Detail

final public static \texttt{SOAPBinding.ParameterStyle[]} \texttt{values()}

values

public static final \texttt{SOAPBinding.ParameterStyle[]} \texttt{values()}
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(SOAPBinding.ParameterStyle c : SOAPBinding.ParameterStyle.values())
    System.out.println(c);
```

**Returns:**
an array containing the constants of this enum type, in the order they're declared

---

**public static SOAPBinding.ParameterStyle valueOf(String name)**

**valueOf**

**valueOf**

**public static SOAPBinding.ParameterStyle valueOf(String name)**

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
name - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**
IllegalArgumentException - if this enum type has no constant with the specified name

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
javax.jws.soap  Enum SOAPBinding.Style

java.lang.Object
    java.lang.Enum<SOAPBinding.Style>
    javax.jws.soap.SOAPBinding.Style

All Implemented Interfaces:
    Serializable, Comparable<SOAPBinding.Style>

Enclosing class:
    SOAPBinding

public static enum SOAPBinding.Style
    extends Enum<SOAPBinding.Style>

Extends: Enum<E>
Contained within: SOAPBinding

SOAP

The SOAP binding style

---

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Document</th>
<th>RPC</th>
</tr>
</thead>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static SOAPBinding.Style.valueOf(String name)</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
</tbody>
</table>
static SOAPBinding.Style[] values() Returns an array containing the constants of this enum type, in the order they’re declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

DOCUMENT

public static final SOAPBinding.Style DOCUMENT

RPC

public static final SOAPBinding.Style RPC

Method Detail

final public static SOAPBinding.Style[] values()

values

public static final SOAPBinding.Style[] values()
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(SOAPBinding.Style c : SOAPBinding.Style.values())
    System.out.println(c);
```

**Returns:**
an array containing the constants of this enum type, in the order they're declared

### public static `SOAPBinding.Style` `valueOf(String name)`

**valueOf**

```java
public static SOAPBinding.Style `valueOf(String name)`
```

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
- `name` - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**
- `IllegalArgumentException` - if this enum type has no constant with the specified name
PS:
javax.jws.soap  Enum SOAPBinding.Use

java.lang.Object
   - java.lang.Enum<SOAPBinding.Use>
   - javax.jws.soap.SOAPBinding.Use

All Implemented Interfaces:
    Serializable, Comparable<SOAPBinding.Use>

Enclosing class:
    SOAPBinding

public static enum SOAPBinding.Use
extends Enum<SOAPBinding.Use>

Extends: Enum<E>
Contained within: SOAPBinding

SOAP

The SOAP binding use

## Enum Constant Summary

<table>
<thead>
<tr>
<th>ENCODED</th>
</tr>
</thead>
<tbody>
<tr>
<td>LITERAL</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>static SOAPBinding.Use</th>
<th>valueOf(String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
</tbody>
</table>
static SOAPBinding.Use[] values()    Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

LITERAL
public static final SOAPBinding.Use LITERAL

ENCODED
public static final SOAPBinding.Use ENCODED

Method Detail

final public static SOAPBinding.Use[] values()
Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(SOAPBinding.Use c : SOAPBinding.Use.values())
    System.out.println(c);
```

**Returns:**
an array containing the constants of this enum type, in the order they're declared

---

**public static SOAPBinding.Use valueOf(String name)**

**valueOf**

**public static SOAPBinding.Use valueOf(String name)**

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
name - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**
IllegalArgumentException - if this enum type has no constant with the specified name

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.soap  Interface SOAPBody

All Superinterfaces:
   Element, Node, SOAPElement

public interface SOAPBody
   extends SOAPElement

Implements: SOAPElement

SOAP  SOAP  SOAP  XML

SOAPBody  SOAPBodyElement  SOAP /  SOAPFault
SOAPBodyElement

See also  javax.xml.soap.SOAPFault

An object that represents the contents of the SOAP body element in a SOAP message. A SOAP body element consists of XML data that affects the way the application-specific content is processed.

A SOAPBody object contains SOAPBodyElement objects, which have the content for the SOAP body. A SOAPFault object, which carries status and/or error information, is an example of a SOAPBodyElement object.

See Also:
   SOAPFault

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPBodyElement</td>
<td><code>addBodyElement(Name name)</code></td>
<td>Creates a new SOAPBodyElement object with the specified name and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPBodyElement</td>
<td><code>addBodyElement(QName qname)</code></td>
<td>Creates a new SOAPBodyElement object with the specified QName and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPBodyElement</td>
<td><code>addDocument(Document document)</code></td>
<td>Adds the root node of the DOM Document to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>addFault()</code></td>
<td>Creates a new SOAPFault object and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>addFault(Name faultCode, String faultString)</code></td>
<td>Creates a new SOAPFault object and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>addFault(Name faultCode, String faultString, Locale locale)</code></td>
<td>Creates a new SOAPFault object and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>addFault(QName faultCode, String faultString)</code></td>
<td>Creates a new SOAPFault object and adds it to this SOAPBody object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>SOAPBody</strong> object. <strong>addFault</strong>( QName faultCode, String faultString, Locale locale)</td>
<td>Creates a new SOAPFault object and adds it to this SOAPBody object.</td>
<td></td>
</tr>
<tr>
<td><strong>Document</strong> <strong>extractContentAsDocument</strong>()</td>
<td>Creates a new DOM Document and sets the first child of this SOAPBody as its document element.</td>
<td></td>
</tr>
<tr>
<td><strong>SOAPFault</strong> <strong>getFault</strong>()</td>
<td>Returns the SOAPFault object in this SOAPBody object.</td>
<td></td>
</tr>
<tr>
<td>boolean <strong>hasFault</strong>()</td>
<td>Indicates whether a SOAPFault object exists in this SOAPBody object.</td>
<td></td>
</tr>
</tbody>
</table>

**Methods inherited from interface javax.xml.soap.SOAPElement**
- addAttribute, addAttribute, addChildElement, addChildElement, addChildElement, addChildElement, addNamespaceDeclaration, addTextNode, createQName, getAllAttributes, getAllAttributesAsQNames, getAttributeValue, getAttributeValue, getChildElements, getChildElements, getChildElements, getElementName, getEncodingStyle, getNamespacePrefixes, getNamespaceURI, getVisibleNamespacePrefixes, removeAttribute, removeAttributes, removeContents, removeNamespaceDeclaration, setElementQName, setEncodingStyle

**Methods inherited from interface javax.xml.soap.Node**
- detachNode, getParentElement, getValue, recycleNode, setParentElement, setValue

**Methods inherited from interface org.w3c.dom.Node**
- appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setData
Methods inherited from interface org.w3c.dom.Element

- getAttribute
- getAttributeNode
- getAttributeNodeNS
- getAttributeNS
- getElementsByTagName
- getElementsByTagNameNS
- getSchemaTypeInfo
- getTagName
- hasAttribute
- hasAttributeNS
- removeAttribute
- removeAttributeNode
- removeAttributeNS
- setAttribute
- setAttributeNode
- setAttributeNodeNS
- setAttributeNS
- setIdAttribute
- setIdAttributeNode
- setIdAttributeNS

Methods inherited from interface org.w3c.dom.Node

- appendChild
- cloneNode
- compareDocumentPosition
- getAttributes
- getBaseURI
- getChildNodes
- getFeature
- getFirstChild
- getLastChild
- getLocalName
- getNamespaceURI
- getNextSibling
- getNodeName
- getNodeType
- getNodeValue
- getOwnerDocument
- getParentNode
- getPrefix
- getPreviousSibling
- getElementsByTagName
- getElementsByTagNameNS
- getSchemaTypeInfo
- getTagName
- hasAttributes
- hasChildNodes
- insertBefore
- isDefaultNamespace
- isEqualNode
- isSameNode
- isSupported
- lookupNamespaceURI
- lookupPrefix
- normalize
- removeChild
- replaceChild
- setNodeValue
- setPrefix
- setTextContent
- setUserData

Method Detail

public SOAPFault addFault() throws SOAPException

SOAPFault SOAPBody SOAPFault (mandatory)
SOAPFault SOAP 1.1 SOAP 1.2 SOAPFault

MessageFactory protocol

SOAPBody SOAPFault
return SOAPFault

Throws SOAPException: SOAP

addFault

SOAPFault addFault() throws SOAPException
Creates a new SOAPFault object and adds it to this SOAPBody object. The new SOAPFault will have default values set for the mandatory child elements. The type of the SOAPFault will be a SOAP 1.1 or a SOAP 1.2 SOAPFault depending on the protocol specified while creating the MessageFactory instance.

A SOAPBody may contain at most one SOAPFault child element.

**Returns:**
the new SOAPFault object

**Throws:**
SOAPException - if there is a SOAP error

```java
public SOAPFault addFault(Name faultCode, String faultString, java.util.Locale locale) throws SOAPException

SOAPFault     SOAPBody     SOAPFault   SOAP 1.1
SOAP 1.2    SOAPFault    MessageFactory    protocol

SOAP 1.2    faultCode    Fault/Code/Value
faultString    Fault/Reason/Text    SOAP
1.1    faultCode    faultcode    faultString    faultstring
```

**SOAPBody**     **SOAPFault**

| faultCode | Name | SOAP |
| faultString | String |     |
| locale | java.util.Locale | faultString |
| return | SOAPFault |        |
| Throws | SOAPException: SOAP |
| since | SAAJ 1.2 |
| See also | setFaultCode, setFaultString |

**addFault**
SOAPFault addFault(Name faultCode, String faultString, Locale locale) throws SOAPException

Creates a new SOAPFault object and adds it to this SOAPBody object. The type of the SOAPFault will be a SOAP 1.1 or a SOAP 1.2 SOAPFault depending on the protocol specified while creating the MessageFactory instance.

For SOAP 1.2 the faultCode parameter is the value of the Fault/Code/Value element and the faultString parameter is the value of the Fault/Reason/Text element. For SOAP 1.1 the faultCode parameter is the value of the faultcode element and the faultString parameter is the value of the faultstring element.

A SOAPBody may contain at most one SOAPFault child element.

Parameters:
- faultCode - a Name object giving the fault code to be set; must be one of the fault codes defined in the Version of SOAP specification in use
- faultString - a String giving an explanation of the fault
- locale - a Locale object indicating the native language of the faultString

Returns:
- the new SOAPFault object

Throws:
- SOAPException - if there is a SOAP error

Since:
- SAAJ 1.2

See Also:
- SOAPFault.setFaultCode(javax.xml.soap.Name)
- SOAPFault.setFaultString(java.lang.String)

public SOAPFault addFault(javax.xml.namespace.QName faultCode, String faultString, java.util.Locale locale) throws SOAPException

SOAPFault SOAPBody SOAPFault SOAP 1.1
SOAP 1.2 SOAPFault MessageFactory protocol

SOAP 1.2 faultCode Fault/Code/Value
faultString Fault/Reason/Text SOAP
1.1 faultCode faultcode faultString faultstring

SOAPBody SOAPFault

faultCode QName SOAP
faultString String
locale Locale faultString
return SOAPFault
Throws SOAPException: SOAP
since SAAJ 1.3
See also setFaultCode, setFaultString, addFault(Name faultCode, String faultString, Locale locale)

addFault

SOAPFault addFault(QName faultCode,
String faultString,
Locale locale)
throws SOAPException

Creates a new SOAPFault object and adds it to this SOAPBody object. The type of the SOAPFault will be a SOAP 1.1 or a SOAP 1.2 SOAPFault depending on the protocol specified while creating the MessageFactory instance.

For SOAP 1.2 the faultCode parameter is the value of the Fault/Code/Value element and the faultString parameter is the value of the Fault/Reason/Text element. For SOAP 1.1 the faultCode parameter is the value of the faultcode element and the faultString parameter is the value of the faultstring element.

A SOAPBody may contain at most one SOAPFault child element.
Parameters:
faultCode - a QName object giving the fault code to be set; must be one of the fault codes defined in the version of SOAP specification in use.
faultString - a String giving an explanation of the fault
locale - a Locale object indicating the native language of the faultString

Returns:
the new SOAPFault object

Throws:
SOAPException - if there is a SOAP error

Since:
SAAJ 1.3

See Also:
SOAPFault.setFaultCode(javax.xml.soap.Name),
SOAPFault.setFaultString(java.lang.String), addFault(Name faultCode, String faultString, Locale locale)

public SOAPFault addFault(Name faultCode, String faultString) throws SOAPException

SOAP 1.2 SOAPFault SOAPBody SOAPFault SOAP 1.1
SOAP 1.2 SOAPFault MessageFactory protocol

SOAP 1.2 faultCode Fault/Code/Value
faultString Fault/Reason/Text SOAP 1.1
faultCode faultcode faultString faultstring

SOAP 1.2 Fault/Reason/Text xml:lang
java.util.Locale.getDefault()

SOAPBody SOAPFault
faultCode Name SOAP
faultString String
return SOAPFault
Throws SOAPException: SOAP
addFault

SOAPFault addFault(Name faultCode, String faultString)
throws SOAPException

Creates a new SOAPFault object and adds it to this SOAPBody object. The type of the SOAPFault will be a SOAP 1.1 or a SOAP 1.2 SOAPFault depending on the protocol specified while creating the MessageFactory instance.

For SOAP 1.2 the faultCode parameter is the value of the Fault/Code/Value element and the faultString parameter is the value of the Fault/Reason/Text element. For SOAP 1.1 the faultCode parameter is the value of the faultcode element and the faultString parameter is the value of the faultstring element.

In case of a SOAP 1.2 fault, the default value for the mandatory xml:lang attribute on the Fault/Reason/Text element will be set to java.util.Locale.getDefault()

A SOAPBody may contain at most one SOAPFault child element.

Parameters:
  - faultCode - a Name object giving the fault code to be set; must be one of the fault codes defined in the version of SOAP specification in use
  - faultString - a String giving an explanation of the fault

Returns:
  the new SOAPFault object

Throws:
  SOAPException - if there is a SOAP error

Since:
  SAAJ 1.2

See Also:
public SOAPFault addFault(javax.xml.namespace.QName faultCode, String faultString) throws SOAPException

SOAPFault SOAPBody SOAPFault SOAP 1.1
SOAP 1.2 SOAPFault MessageFactory protocol

SOAP 1.2  
faultCode  Fault/Code/Value
faultString  Fault/Reason/Text  SOAP
1.1  faultCode  faultcode  faultString  faultstring

SOAP 1.2  Fault/Reason/Text  xml:lang
java.util.Locale.getDefault()

SOAPBody SOAPFault
faultCode  QName SOAP
faultString  String
return  SOAPFault
Throws SOAPException: SOAP
since  SAAJ 1.3
See also  setFaultCode, setFaultString, addFault(Name faultCode, String faultString)

addFault

SOAPFault addFault(QName faultCode,
String faultString)
throws SOAPException

Creates a new SOAPFault object and adds it to this SOAPBody object. The type of the SOAPFault will be a SOAP 1.1 or a SOAP 1.2 SOAPFault depending on the protocol specified while creating the
MessageFactory instance.

For SOAP 1.2 the faultCode parameter is the value of the Fault/Code/Value element and the faultString parameter is the value of the Fault/Reason/Text element. For SOAP 1.1 the faultCode parameter is the value of the faultcode element and the faultString parameter is the value of the faultstring element.

In case of a SOAP 1.2 fault, the default value for the mandatory xml:lang attribute on the Fault/Reason/Text element will be set to java.util.Locale.getDefault()

A SOAPBody may contain at most one SOAPFault child element

Parameters:
- faultCode - a QName object giving the fault code to be set; must be one of the fault codes defined in the version of SOAP specification in use
- faultString - a String giving an explanation of the fault

Returns:
- the new SOAPFault object

Throws:
- SOAPException - if there is a SOAP error

Since:
- SAAJ 1.3

See Also:

---

public boolean hasFault()

SOAPBody SOAPFault
return SOAPBody SOAPFault true false

hasFault

boolean hasFault()
Indicates whether a SOAPFault object exists in this SOAPBody object.

Returns:
true if a SOAPFault object exists in this SOAPBody object; false otherwise.

```java
public SOAPFault getFault()
    SOAPBody SOAPFault
    return SOAPBody SOAPFault null
```

getFault

SOAPFault getFault()

Returns the SOAPFault object in this SOAPBody object.

Returns:
the SOAPFault object in this SOAPBody object if present, null otherwise.

```java
public SOAPBodyElement addBodyElement(Name name)
    throws SOAPException
```

addBodyElement

SOAPBodyElement addBodyElement(Name name)
    throws SOAPException

Creates a new SOAPBodyElement object with the specified name and
addBodyElement

Parameters:
name - a Name object with the name for the new SOAPBodyElement object

Returns:
the new SOAPBodyElement object

Throws:
SOAPException - if a SOAP error occurs

See Also:
addBodyElement(javax.xml.namespace.QName)

public SOAPBodyElement addBodyElement(javax.xml.namespace.QName qname)
throws SOAPException

QName SOAPBodyElement SOAPBody
qname QName SOAPBodyElement
return SOAPBodyElement
Throws SOAPException: SOAP
since SAAJ 1.3
See also addBodyElement(Name)

addBodyElement

SOAPBodyElement addBodyElement(QName qname)
throws SOAPException

Creates a new SOAPBodyElement object with the specified QName and adds it to this SOAPBody object.

Parameters:
qname - a QName object with the qname for the new SOAPBodyElement object

Returns:
the new SOAPBodyElement object

Throws:
SOAPException - if a SOAP error occurs

Since:
SAAJ 1.3

See Also:
addBodyElement(Name)

---

```java
default

public SOAPBodyElement addDocument(org.w3c.dom.Document document) throws SOAPException

DOM org.w3c.dom.Document SOAPBody
document addDocument Document
return
Threws SOAPException: Document
since SAAJ 1.2
```

**addDocument**

SOAPBodyElement addDocument(Document document) throws SOAPException

Adds the root node of the DOM Document to this SOAPBody object.

Calling this method invalidates the document parameter. The client application should discard all references to this Document and its contents upon calling addDocument. The behavior of an application that continues to use such references is undefined.

**Parameters:**
- document - the Document object whose root node will be added to this SOAPBody.

**Returns:**
- the SOAPBodyElement that represents the root node that was added.
Throws:

SOAPException - if the Document cannot be added

Since:

SAAJ 1.2

---

public org.w3c.dom.Document extractContentAsDocument() throws SOAPException

DOM org.w3c.dom.Document SOAPBody SOAPElement

return SOAPBody org.w3c.dom.Document

Throws SOAPException: SOAPBody SOAPElement

since SAAJ 1.3

extractContentAsDocument

Document extractContentAsDocument() throws SOAPException

Creates a new DOM Document and sets the first child of this SOAPBody as its document element. The child SOAPElement is removed as part of the process.

Returns:

the Document representation of the SOAPBody content.

Throws:

SOAPException - if there is not exactly one child SOAPElement of the SOAPBody.

Since:

SAAJ 1.3

---

Submit a bug or feature
javax.xml.soap  Interface SOAPBodyElement

All Superinterfaces:
   Element, Node, SOAPElement

All Known Subinterfaces:
   SOAPFault

public interface SOAPBodyElement
extends SOAPElement

Implements: SOAPElement
Implemented by: SOAPFault

SOAPBody  SOAPBodyElement  SOAPFault  SOAPBodyElement
SOAPBodyElement  SOAPBody  addBodyElement  SOAPBody
sb  SOAPBody  myName  Name

SOAPBodyElement sbe = sb.addBodyElement(myName);

A SOAPBodyElement object represents the contents in a SOAPBody object. The SOAPFault interface is a SOAPBodyElement object that has been defined.

A new SOAPBodyElement object can be created and added to a SOAPBody object with the SOAPBody method addBodyElement. In the following line of code, sb is a SOAPBody object, and myName is a Name object.

   SOAPBodyElement sbe = sb.addBodyElement(myName);
# Field Summary

Fields inherited from interface org.w3c.dom.*Node*

- ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,
  DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,
  DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,
  DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,
  DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,
  DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,
  ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,
  PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface org.w3c.dom.*Node*

- ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,
  DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,
  DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,
  DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,
  DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,
  DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,
  ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,
  PROCESSING_INSTRUCTION_NODE, TEXT_NODE

# Method Summary

Methods inherited from interface javax.xml.soap.*SOAPElement*

- addAttribute, addAttribute, addChildElement, addChildElement,
  addChildElement, addChildElement, addChildElement,
  createQName, getAllAttributes, getAllAttributesAsQNames,
  getAttributeValue, getAttributeValue, getChildElements,
  getChildElements, getChildElements, getElementName,
  getElementQName, getEncodingStyle, getNamespacePrefixes,
  getNamespaceURI, getVisibleNamespacePrefixes, removeAttribute,
  removeAttribute, removeContents, removeNamespaceDeclaration,
  setElementQName, setEncodingStyle

Methods inherited from interface javax.xml.soap.*Node*

- detachNode, getParentElement, getValue, recycleNode,
  setParentElement, setValue
<table>
<thead>
<tr>
<th>Methods inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getFeature, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface org.w3c.dom.Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAttribute, getAttributeNode, getAttributeNodeNS, getAttributeNS, getElementsByTagName, getElementsByTagNameNS, getSchemaTypeInfo, getTagName, hasAttribute, hasAttributeNS, removeAttribute, removeAttributeNode, removeAttributeNS, setAttribute, setAttributeNode, setAttributeNodeNS, setAttributeNS, setIdAttribute, setIdAttributeNode, setIdAttributeNS</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getFeature, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData</td>
</tr>
</tbody>
</table>
PS:
javax.xml.soap  **Class SOAPConnection**

```java
java.lang.Object
   - javax.xml.soap.SOAPConnection
```

**public abstract class SOAPConnection**

**extends Object**

**URL**

```java
SOAPConnectionFactory factory = SOAPConnectionFactory.newInstance();
SOAPConnection con = factory.createConnection();
```

A **point-to-point** connection that a client can use for sending messages directly to a remote party (represented by a URL, for instance).

The SOAPConnection class is optional. Some implementations may not implement this interface in which case the call to `SOAPConnectionFactory.newInstance()` (see below) will throw an `UnsupportedOperationException`.

A client can obtain a **SOAPConnection** object using a **SOAPConnectionFactory** object as in the following example:

```java
SOAPConnectionFactory factory = SOAPConnectionFactory.newInstance();
SOAPConnection con = factory.createConnection();
```

A **SOAPConnection** object can be used to send messages directly to a URL
following the request/response paradigm. That is, messages are sent using the method call, which sends the message and then waits until it gets a reply.

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPConnection()</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract SOAPMessage call(SOAPMessage request, Object to)</td>
<td>Sends the given message to the specified endpoint and blocks until it has returned the response.</td>
</tr>
<tr>
<td>abstract void close()</td>
<td>Closes this SOAPConnection object.</td>
</tr>
<tr>
<td>SOAPMessage get(Object to)</td>
<td>Gets a message from a specific endpoint and blocks until it receives,</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

### Constructor Detail

public SOAPConnection()

SOAPConnection

public SOAPConnection()
abstract public **SOAPMessage** call(**SOAPMessage** request, Object to) throws **SOAPException**

<table>
<thead>
<tr>
<th>request</th>
<th>SOAPMessage</th>
</tr>
</thead>
<tbody>
<tr>
<td>to</td>
<td>Object</td>
</tr>
<tr>
<td>return</td>
<td><strong>SOAPMessage</strong></td>
</tr>
<tr>
<td>Throws</td>
<td><strong>SOAPException</strong>: SOAP</td>
</tr>
</tbody>
</table>

**call**

public abstract **SOAPMessage** call(**SOAPMessage** request, Object to) throws **SOAPException**

Sends the given message to the specified endpoint and blocks until it has returned the response.

**Parameters:**
- request - the SOAPMessage object to be sent
- to - an Object that identifies where the message should be sent.
  It is required to support Objects of type `java.lang.String`, `java.net.URL`, and when JAXM is present `javax.xml.messaging.URL Endpoint`

**Returns:**
- the SOAPMessage object that is the response to the message that was sent

**Throws:**
- **SOAPException** - if there is a SOAP error

public **SOAPMessage** get(Object to) throws **SOAPException**
get

public SOAPMessage get(Object to)
    throws SOAPException

    Gets a message from a specific endpoint and blocks until it receives,

Parameters:
    to - an Object that identifies where the request should be sent. Objects of type java.lang.String and java.net.URL must be supported.

Returns:
    the SOAPMessage object that is the response to the get message request

Throws:
    SOAPException - if there is a SOAP error

Since:
    SAAJ 1.3

abstract public void close() throws SOAPException

SoapConnection

Throws
    SOAPException: SOAP

close

public abstract void close()
    throws SOAPException

Closes this SOAPConnection object.

Throws:
SOAPException - if there is a SOAP error
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.soap Class SOAPConnectionFactory

java.lang.Object
  javax.xml.soap.SOAPConnectionFactory

public abstract class SOAPConnectionFactory extends Object

SOAPConnection SOAPConnectionFactory.newInstance()

UnsupportedOperationException SAAJ SOAPConnectionFactory.createConnection()

A factory for creating SOAPConnection objects. Implementation of this class is optional. If SOAPConnectionFactory.newInstance() throws an UnsupportedOperationException then the implementation does not support the SAAJ communication infrastructure. Otherwise SOAPConnection objects can be created by calling createConnection() on the newly created SOAPConnectionFactory object.

Constructor Summary

SOAPConnectionFactory()}

Method Summary

abstract SOAPConnection createConnection() Create a new SOAPConnection.

static SOAPConnectionFactory newInstance() Creates an instance of the default SOAPConnectionFactory object.
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public SOAPConnectionFactory()

SOAPConnectionFactory

public SOAPConnectionFactory()

Method Detail

public static SOAPConnectionFactory newInstance()
throws SOAPException, UnsupportedOperationException

SOAPConnectionFactory

return SOAPConnectionFactory

Throws SOAPException: SOAPConnectionFactory

Throws UnsupportedOperationException: SOAPConnectionFactory

newInstance

Create an instance of the default SOAPConnectionFactory object.

Returns:
a new instance of a default SOAPConnectionFactory object

Throws:
SOAPException - if there was an error creating the
SOAPConnectionFactory

UnsupportedOperationException - if newInstance is not supported.

abstract public SOAPConnection createConnection()

throws SOAPException

SOAPConnection

return SOAPConnection

Throws SOAPException: SOAPConnection

createConnection

public abstract SOAPConnection createConnection()

throws SOAPException

Create a new SOAPConnection.

Returns: the new SOAPConnection object.

Throws: SOAPException - if there was an exception creating the SOAPConnection object.
public interface SOAPConstants

SOAP

The definition of constants pertaining to the SOAP protocol.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static String</strong> DEFAULT_SOAP_PROTOCOL</td>
</tr>
<tr>
<td><strong>static String</strong> DYNAMIC_SOAP_PROTOCOL</td>
</tr>
<tr>
<td><strong>static String</strong> SOAP_1_1_CONTENT_TYPE</td>
</tr>
<tr>
<td><strong>static String</strong> SOAP_1_1_PROTOCOL</td>
</tr>
<tr>
<td><strong>static String</strong> SOAP_1_2_CONTENT_TYPE</td>
</tr>
<tr>
<td><strong>static String</strong> SOAP_1_2_PROTOCOL</td>
</tr>
<tr>
<td><strong>static String</strong> SOAP_DATAENCODINGUNKNOWN_FAULT</td>
</tr>
<tr>
<td>static QName</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The default namespace prefix for <a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></td>
</tr>
<tr>
<td>static QName</td>
</tr>
<tr>
<td>SOAP 1.2 MustUnderstand Fault</td>
</tr>
<tr>
<td>static QName</td>
</tr>
<tr>
<td>SOAP 1.2 Receiver Fault</td>
</tr>
<tr>
<td>static QName</td>
</tr>
<tr>
<td>SOAP 1.2 Sender Fault</td>
</tr>
<tr>
<td>static QName</td>
</tr>
<tr>
<td>SOAP 1.2 VersionMismatch Fault</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The namespace identifier for the SOAP 1.1 envelope.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The namespace identifier for the SOAP 1.2 encoding.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The namespace identifier for the SOAP 1.2 envelope.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The namespace identifier for the SOAP 1.1 encoding.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The namespace identifier for the SOAP 1.1 envelope, All SOAPEElements in this namespace are defined by the SOAP 1.1 specification.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The URI identifying the next application processing a SOAP request as the intended role for a SOAP 1.2 header entry (see section 2.2 of part 1 of the SOAP 1.2 specification).</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The URI specifying the role None in SOAP 1.2.</td>
</tr>
<tr>
<td>static String</td>
</tr>
<tr>
<td>The URI identifying the ultimate receiver of the SOAP 1.2 message.</td>
</tr>
</tbody>
</table>
The URI identifying the next application processing a SOAP request as the intended actor for a SOAP 1.1 header entry (see section 4.2.2 of the SOAP 1.1 specification).

### DYNAMIC_SOAP_PROTOCOL

```java
static final String DYNAMIC_SOAP_PROTOCOL
```

Used to create `MessageFactory` instances that create `SOAPMessage` whose concrete type is based on the `Content-Type` MIME header passed to the `createMessage` method. If no `Content-Type` header is passed then the `createMessage` may throw an `IllegalArgumentException` or, in the case of the no argument version of `createMessage`, an `UnsupportedOperationException`.

**Since:**

SAAJ 1.3

**See Also:**

- [Constant Field Values](#)

### SOAP_1_1_PROTOCOL

```java
static final String SOAP_1_1_PROTOCOL
```

Used to create `MessageFactory` instances that create `SOAPMessage` whose behavior supports the SOAP 1.1 specification.

**Since:**

SAAJ 1.3
SOAP_1_2_PROTOCOL

static final String SOAP_1_2_PROTOCOL

Used to create MessageFactory instances that create SOAPMessages whose behavior supports the SOAP 1.2 specification

Since:
SAAJ 1.3
See Also:
Constant Field Values

DEFAULT_SOAP_PROTOCOL

static final String DEFAULT_SOAP_PROTOCOL

The default protocol: SOAP 1.1 for backwards compatibility.

Since:
SAAJ 1.3
See Also:
Constant Field Values

URI_NS_SOAP_1_1_ENVELOPE

static final String URI_NS_SOAP_1_1_ENVELOPE

The namespace identifier for the SOAP 1.1 envelope.
URI_NS_SOAP_1_2_ENVELOPE

static final String URI_NS_SOAP_1_2_ENVELOPE

The namespace identifier for the SOAP 1.2 envelope.

Since:
SAAJ 1.3
See Also:
Constant Field Values

URI_NS_SOAP_ENVELOPE

static final String URI_NS_SOAP_ENVELOPE

The namespace identifier for the SOAP 1.1 envelope, All SOAPElements in this namespace are defined by the SOAP 1.1 specification.

See Also:
Constant Field Values

URI_NS_SOAP_ENCODING

static final String URI_NS_SOAP_ENCODING
The namespace identifier for the SOAP 1.1 encoding. An attribute named encodingStyle in the URI_NS_SOAP_ENVELOPE namespace and set to the value URI_NS_SOAP_ENCODING can be added to an element to indicate that it is encoded using the rules in section 5 of the SOAP 1.1 specification.

See Also:
Constant Field Values

---

URI_NS_SOAP_1_2_ENCODING

static final String URI_NS_SOAP_1_2_ENCODING

The namespace identifier for the SOAP 1.2 encoding.

Since:
SAAJ 1.3

See Also:
Constant Field Values

---

SOAP_1_1_CONTENT_TYPE

static final String SOAP_1_1_CONTENT_TYPE

The media type of the Content-Type MIME header in SOAP 1.1.

Since:
SAAJ 1.3

See Also:
Constant Field Values
SOAP_1_2_CONTENT_TYPE

static final String SOAP_1_2_CONTENT_TYPE

The media type of the Content-Type MIME header in SOAP 1.2.

Since:
SAAJ 1.3
See Also:
Constant Field Values

URI_SOAP_ACTOR_NEXT

static final String URI_SOAP_ACTOR_NEXT

The URI identifying the next application processing a SOAP request as the intended actor for a SOAP 1.1 header entry (see section 4.2.2 of the SOAP 1.1 specification).

This value can be passed to
SOAPHeader.examineMustUnderstandHeaderElements(String),
SOAPHeader.examineHeaderElements(String) and
SOAPHeader.extractHeaderElements(String)

See Also:
Constant Field Values

URI_SOAP_1_2_ROLE_NEXT

static final String URI_SOAP_1_2_ROLE_NEXT

The URI identifying the next application processing a SOAP request as the intended role for a SOAP 1.2 header entry (see section 2.2 of part 1 of the SOAP 1.2 specification).
URI_SOAP_1_2_ROLE_NONE

static final String URI_SOAP_1_2_ROLE_NONE

The URI specifying the role None in SOAP 1.2.

Since: SAAJ 1.3
See Also: Constant Field Values

------------------------------------------------------

URI_SOAP_1_2_ROLE_ULTIMATE_RECEIVER

static final String URI_SOAP_1_2_ROLE_ULTIMATE_RECEIVER

The URI identifying the ultimate receiver of the SOAP 1.2 message.

Since: SAAJ 1.3
See Also: Constant Field Values

------------------------------------------------------

SOAP_ENV_PREFIX

static final String SOAP_ENV_PREFIX
The default namespace prefix for http://www.w3.org/2003/05/soap-envelope

**Since:**
SAAJ 1.3

**See Also:**
Constant Field Values

---

**SOAP_VERSIONMISMATCH_FAULT**

static final QName SOAP_VERSIONMISMATCH_FAULT

SOAP 1.2 VersionMismatch Fault

**Since:**
SAAJ 1.3

---

**SOAP_MUSTUNDERSTAND_FAULT**

static final QName SOAP_MUSTUNDERSTAND_FAULT

SOAP 1.2 MustUnderstand Fault

**Since:**
SAAJ 1.3

---

**SOAP_DATAENCODINGUNKNOWN_FAULT**

static final QName SOAP_DATAENCODINGUNKNOWN_FAULT

SOAP 1.2 DataEncodingUnknown Fault
Since:
   SAAJ 1.3

SOAP_SENDER_FAULT

static final QName SOAP_SENDER_FAULT

SOAP 1.2 Sender Fault

Since:
   SAAJ 1.3

SOAP_RECEIVER_FAULT

static final QName SOAP_RECEIVER_FAULT

SOAP 1.2 Receiver Fault

Since:
   SAAJ 1.3
javax.xml.soap Interface SOAPElement

All Superinterfaces:
   Element, Node

All Known Subinterfaces:
   Detail, DetailEntry, SOAPBody, SOAPBodyElement, SOAPEnvelope, SOAPFault, SOAPFaultElement, SOAPHeader, SOAPHeaderElement

public interface SOAPElement
   extends Node, Element

Implements: Node, org.w3c.dom.Element
Implemented by: DetailEntry, SOAPBody, SOAPBodyElement, SOAPEnvelope, SOAPFaultElement, SOAPHeader, SOAPHeaderElement

SOAP SOAP SOAP

SAAJ "" summary.html#package_description">javax.xml.soap

An object representing an element of a SOAP message that is allowed but not specifically prescribed by a SOAP specification. This interface serves as the base interface for those objects that are specifically prescribed by a SOAP specification.

Methods in this interface that are required to return SAAJ specific objects may "silently" replace nodes in the tree as required to successfully return objects of the correct type. See getChildElements() and javax.xml.soap for details.
Field Summary

Fields inherited from interface org.w3c.dom.Node
 ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Fields inherited from interface org.w3c.dom.Node
 ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPElement.addAttribute(Name name, String value)</td>
<td>Adds an attribute with the specified name and value to this SOAPElement object.</td>
</tr>
<tr>
<td>SOAPElement.addAttribute(QName qname, String value)</td>
<td>Adds an attribute with the specified name and value to this SOAPElement object.</td>
</tr>
<tr>
<td>SOAPElement.addChildElement(Name name)</td>
<td>Creates a new SOAPElement object initialized with the given Name object and adds the new element to this SOAPElement object.</td>
</tr>
<tr>
<td>SOAPElement.addChildElement(QName qname)</td>
<td>Creates a new SOAPElement object initialized with the given QName object and adds the new element to this SOAPElement object.</td>
</tr>
<tr>
<td>SOAPElement.addChildElement(SOAPElement element)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>addChildElement(String localName)</code></td>
<td>Creates a new SOAPElement object initialized with the specified local name and adds the new element to this SOAPElement object.</td>
</tr>
<tr>
<td><code>addChildElement(String localName, String prefix)</code></td>
<td>Creates a new SOAPElement object initialized with the specified local name and prefix and adds the new element to this SOAPElement object.</td>
</tr>
<tr>
<td><code>addChildElement(String localName, String prefix, String uri)</code></td>
<td>Creates a new SOAPElement object initialized with the specified local name, prefix, and URI and adds the new element to this SOAPElement object.</td>
</tr>
<tr>
<td><code>addNamespaceDeclaration(String prefix, String uri)</code></td>
<td>Adds a namespace declaration with the specified prefix and URI to this SOAPElement object.</td>
</tr>
<tr>
<td><code>addTextNode(String text)</code></td>
<td>Creates a new Text object initialized with the given String and adds it to this SOAPElement object.</td>
</tr>
<tr>
<td><code>createQName(String localName, String prefix)</code></td>
<td>Creates a QName whose namespace URI is the one associated with the parameter, prefix, in the context of this SOAPElement.</td>
</tr>
<tr>
<td><code>getAttributeValue(Name name)</code></td>
<td>Returns the value of the attribute with the specified name.</td>
</tr>
<tr>
<td><code>getAttributeValue(QName qname)</code></td>
<td>Returns the value of the attribute with the specified QName.</td>
</tr>
<tr>
<td><code>getAllAttributes()</code></td>
<td>Returns an Iterator over all of the attribute Name objects in this SOAPElement object.</td>
</tr>
<tr>
<td><code>getAllAttributesAsQNames()</code></td>
<td>Returns an Iterator over all of the attributes in this SOAPElement as QName objects.</td>
</tr>
<tr>
<td>Method (Type)</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Iterator <strong>getChildElements()</strong></td>
<td>Returns an Iterator over all the immediate child nodes of this element.</td>
</tr>
<tr>
<td>Iterator <strong>getChildElements(Name name)</strong></td>
<td>Returns an Iterator over all the immediate child nodes of this element with the specified name.</td>
</tr>
<tr>
<td>Iterator <strong>getChildElements(QName qname)</strong></td>
<td>Returns an Iterator over all the immediate child nodes of this element with the specified qname.</td>
</tr>
<tr>
<td>Name <strong>getElementName()</strong></td>
<td>Returns the name of this SOAPElement object.</td>
</tr>
<tr>
<td>QName <strong>getElementQName()</strong></td>
<td>Returns the qname of this SOAPElement object.</td>
</tr>
<tr>
<td>String <strong>getEncodingStyle()</strong></td>
<td>Returns the encoding style for this SOAPElement object.</td>
</tr>
<tr>
<td>Iterator <strong>getNamespacePrefixes()</strong></td>
<td>Returns an Iterator over the namespace prefix Strings declared by this element.</td>
</tr>
<tr>
<td>String <strong>getNamespaceURI(String prefix)</strong></td>
<td>Returns the URI of the namespace that has the given prefix.</td>
</tr>
<tr>
<td>Iterator <strong>getVisibleNamespacePrefixes()</strong></td>
<td>Returns an Iterator over the namespace prefix Strings visible to this element.</td>
</tr>
<tr>
<td>boolean <strong>removeAttribute(Name name)</strong></td>
<td>Removes the attribute with the specified name.</td>
</tr>
<tr>
<td>boolean <strong>removeAttribute(QName qname)</strong></td>
<td>Removes the attribute with the specified qname.</td>
</tr>
<tr>
<td>void <strong>removeContents()</strong></td>
<td>Detaches all children of this SOAPElement.</td>
</tr>
<tr>
<td>boolean <strong>removeNamespaceDeclaration(String prefix)</strong></td>
<td>Removes the namespace declaration corresponding to the given prefix.</td>
</tr>
<tr>
<td>QName <strong>setElementQName(QName newName)</strong></td>
<td></td>
</tr>
</tbody>
</table>
### SOAPElement

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>setEncodingStyle</strong>(String encodingStyle)</td>
<td>Sets the encoding style for this SOAPElement object to one specified.</td>
</tr>
</tbody>
</table>

Changes the name of this Element to **newName** if possible.

**Methods inherited from interface javax.xml.soap.Node**

- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

**Methods inherited from interface org.w3c.dom.Node**

- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildNodes`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getParentNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`, `replaceChild`, `setNodeValue`, `setPrefix`, `setTextContent`, `setUserData`

**Methods inherited from interface org.w3c.dom.Element**

- `getAttribute`, `getAttributeNode`, `getAttributeNodeNS`, `getAttributeNS`, `getElementsByTagName`, `getElementsByTagNameNS`, `getSchemaTypeInfo`, `getTagName`, `hasAttribute`, `hasAttributeNS`, `removeAttribute`, `removeAttributeNode`, `removeAttributeNS`, `setAttribute`, `setAttributeNode`, `setAttributeNodeNS`, `setAttributeNS`, `setIdAttribute`, `setIdAttributeNode`, `setIdAttributeNS`

**Methods inherited from interface org.w3c.dom.Node**

- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildNodes`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getParentNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`, `replaceChild`, `setNodeValue`, `setPrefix`, `setTextContent`, `setUserData`
## Method Detail

**public SOAPElement addChildElement(Name name) throws SOAPException**

<table>
<thead>
<tr>
<th>Name</th>
<th>SOAPElement</th>
<th>SOAPElement</th>
</tr>
</thead>
</table>

**SAAJ**

**addChildElement(javax.xml.namespace.QName)**

<table>
<thead>
<tr>
<th>name</th>
<th>Name XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>SOAPElement</td>
</tr>
<tr>
<td>Throws</td>
<td>SOAPException: SOAPElement</td>
</tr>
<tr>
<td>See also</td>
<td>addChildElement(javax.xml.namespace.QName)</td>
</tr>
</tbody>
</table>

### addChildElement

**SOAPElement addChildElement(Name name) throws SOAPException**

Creates a new SOAPElement object initialized with the given Name object and adds the new element to this SOAPElement object.

This method may be deprecated in a future release of SAAJ in favor of addChildElement(javax.xml.namespace.QName)

**Parameters:**

- **name**: a Name object with the XML name for the new element

**Returns:**

- the new SOAPElement object that was created

**Throws:**

- SOAPException - if there is an error in creating the SOAPElement object

**See Also:**

- addChildElement(javax.xml.namespace.QName)
public <var>SOAPElement</var> addChildElement(<var>javax.xml.namespace.QName</var> qname) throws <var>SOAPException</var>

    QName SOAPElement SOAPElement SOAPElement
namespace localname prefix qname
    qname QName XML
return SOAPElement
Throws <var>SOAPException</var>: SOAPElement
since SAAJ 1.3
See also <var>addChildElement(Name)</var>

**addChildElement**

<var>SOAPElement</var> addChildElement(<var>QName</var> qname) throws <var>SOAPException</var>

Creates a new <var>SOAPElement</var> object initialized with the given <var>QName</var> object and adds the new element to this <var>SOAPElement</var> object. The namespace, localname and prefix of the new <var>SOAPElement</var> are all taken from the qname argument.

**Parameters:**
qname - a QName object with the XML name for the new element

**Returns:**
the new <var>SOAPElement</var> object that was created

**Throws:**
<var>SOAPException</var> - if there is an error in creating the <var>SOAPElement</var> object

**Since:**
SAAJ 1.3

**See Also:**
<var>addChildElement(Name)</var>

---

public <var>SOAPElement</var> addChildElement(<var>String</var> localName) throws <var>SOAPException</var>
addChildElement

SOAPElement addChildElement(String localName) throws SOAPException

Creates a new SOAPElement object initialized with the specified local name and adds the new element to this SOAPElement object. The new SOAPElement inherits any in-scope default namespace.

Parameters:
  localName - a String giving the local name for the element

Returns:
  the new SOAPElement object that was created

Throws:
  SOAPException - if there is an error in creating the SOAPElement object

public SOAPElement addChildElement(String localName, String prefix) throws SOAPException

SOAPElement addChildElement(String localName, String prefix)

SOAPElement addChildElement(SOAPElement localName, SOAPElement prefix) throws SOAPException

addChildElement

SOAPElement addChildElement(String localName, String prefix)
throws SOAPException

Creates a new SOAPElement object initialized with the specified local name and prefix and adds the new element to this SOAPElement object.

Parameters:
localName - a String giving the local name for the new element
prefix - a String giving the namespace prefix for the new element

Returns:
the new SOAPElement object that was created

Throws:
SOAPException - if the prefix is not valid in the context of this SOAPElement or if there is an error in creating the SOAPElement object

public SOAPElement addChildElement(String localName, String prefix, String uri) throws SOAPException

URI

localName
prefix
uri

return

SOAPElement

Throws

SOAPException

SOAPElement

addChildElement

SOAPElement addChildElement(String localName,
String prefix,
String uri)
throws SOAPException

Creates a new SOAPElement object initialized with the specified local name, prefix, and URI and adds the new element to this SOAPElement object.
Parameters:
- **localName** - a **String** giving the local name for the new element
- **prefix** - a **String** giving the namespace prefix for the new element
- **uri** - a **String** giving the URI of the namespace to which the new element belongs

Returns:
- the new **SOAPElement** object that was created

Throws:
- **SOAPException** - if there is an error in creating the **SOAPElement** object

```java
public **SOAPElement** addChildElement(**SOAPElement** element) throws **SOAPException**
```

**SOAPElement** **soapElement** **SOAPFactory**

**SOAPElement** **XML**

**SOAPElement** **SOAPElement**

**SOAPHeader** **addChildElement**() **element**

**SOAPHeaderElement**

**element**

**element** "Envelope""Header" "Body" SOAP

**element** **SOAPElement**

**Throws** **SOAPException**:

```java
return **SOAP**
```

**addChildElement**

**SOAPElement** **addChildElement**(**SOAPElement** element) throws **SOAPException**

**SOAPElement** **addChildElement**(**SOAPElement** element)

**throws** **SOAPException**

Add a **SOAPElement** as a child of this **SOAPElement** instance. The **SOAPElement** is expected to be created by a **SOAPFactory**. Callers
should not rely on the element instance being added as is into the XML tree. Implementations could end up copying the content of the SOAPElement passed into an instance of a different SOAPElement implementation. For instance if addChildElement() is called on a SOAPHeader, element will be copied into an instance of a SOAPHeaderElement.

The fragment rooted in element is either added as a whole or not at all, if there was an error.

The fragment rooted in element cannot contain elements named "Envelope", "Header" or "Body" and in the SOAP namespace. Any namespace prefixes present in the fragment should be fully resolved using appropriate namespace declarations within the fragment itself.

**Parameters:**
- element - the SOAPElement to be added as a new child

**Returns:**
- an instance representing the new SOAP element that was actually added to the tree.

**Throws:**
- SOAPException - if there was an error in adding this element as a child

```java
public void removeContents()

SOAPElement

SOAPElement

SOAPHeaders  SOAPBodys  SOAP

since  SAAJ 1.2

removeContents

void removeContents()

Detaches all children of this SOAPElement.
This method is useful for rolling back the construction of partially completed SOAPHeaders and SOAPBodys in preparation for sending a fault when an error condition is detected. It is also useful for recycling portions of a document within a SOAP message.

Since:
  SAAJ 1.2

```java
public SOAPElement addTextNode(String text) throws SOAPException

String text
return SOAPElement

Throws: SOAPException
```

```
addTextNode

SOAPElement addTextNode(String text)
throws SOAPException

Creates a new Text object initialized with the given String and adds it to this SOAPElement object.

Parameters:
  text - a String object with the textual content to be added

Returns:
  the SOAPElement object into which the new Text object was inserted

Throws:
  SOAPException - if there is an error in creating the new Text object or if it is not legal to attach it as a child to this SOAPElement
```

```java
public SOAPElement addAttribute(Name name, String value) throws SOAPException

SOAPElement
```
**addAttribute**

```java
SOAPElement addAttribute(Name name, String value) throws SOAPException
```

Adds an attribute with the specified name and value to this `SOAPElement` object.

**Parameters:**
- `name` - a `Name` object with the name of the attribute
- `value` - a `String` giving the value of the attribute

**Returns:**
the `SOAPElement` object into which the attribute was inserted

**Throws:**
- `SOAPException` - if there is an error in creating the Attribute, or it is invalid to set an attribute with `Name` name on this `SOAPElement`.

**See Also:**
`addAttribute(javax.xml.namespace.QName, String)`

---

**public SOAPElement**

```java
addAttribute(javax.xml.namespace.QName qname, String value) throws SOAPException
```

```java
SOAPElement
qname QName
value String
return SOAPElement
```

**Throws**
- `SOAPException` - `SOAPElement`

**since**
SAAJ 1.3

**See also**
`addAttribute(Name, String)`
addAttribute

**SOAPElement addAttribute**(QName qname, String value) throws SOAPException

Adds an attribute with the specified name and value to this SOAPElement object.

**Parameters:**
- qname - a QName object with the name of the attribute
- value - a String giving the value of the attribute

**Returns:**
the SOAPElement object into which the attribute was inserted

**Throws:**
SOAPException - if there is an error in creating the Attribute, or it is invalid to set an attribute with QName qname on this SOAPElement.

**Since:**
SAAJ 1.3

**See Also:**
addAttribute(Name, String)

---

public **SOAPElement addNamespaceDeclaration**(String prefix, String uri) throws SOAPException

**URI**

prefix String

uri String

**Return**

SOAPElement

**Throws**

SOAPException:

addNamespaceDeclaration

**SOAPElement addNamespaceDeclaration**(String prefix, String uri)
Adds a namespace declaration with the specified prefix and URI to this SOAPElement object.

**Parameters:**
- prefix - a String giving the prefix of the namespace
- uri - a String giving the uri of the namespace

**Returns:**
the SOAPElement object into which this namespace declaration was inserted.

**Throws:**
SOAPException - if there is an error in creating the namespace

---

**public String getAttributeValue(Name name)**

- **name** Name
- **return** String Null

See also [getAttributeValue(javax.xml.namespace.QName)]

---

**getAttributeValue**

**Public String getAttributeValue(Name name)**

Returns the value of the attribute with the specified name.

**Parameters:**
- name - a Name object with the name of the attribute

**Returns:**
- a String giving the value of the specified attribute, Null if there is no such attribute

See Also:
- [getAttributeValue(javax.xml.namespace.QName)]
**getAttributeValue(java.xml.namespace.QName qname)**

Parameter:
- qname - a QName object with the qname of the attribute

Returns:
- a String giving the value of the specified attribute, Null if there is no such attribute

Since:
- SAAJ 1.3

See Also:
- `getAttributeValue(Name)`

---

**public java.util.Iterator&lt;E&gt; getAllAttributes()**

Parameters:
- SOAPElement - Name

Returns:
- an Iterator over all of the attribute Name objects in this
SOAPElement object. The iterator can be used to get the attribute names, which can then be passed to the method `getAttributeValue` to retrieve the value of each attribute.

**Returns:**

an iterator over the names of the attributes

**See Also:**

`getAllAttributesAsQNames()`

---

```java
public java.util.Iterator<E> getAllAttributesAsQNames()
```

**SOAPElement QName Iterator QName**

`getAttributeValue`

```java
return
```

QNames

```java
since SAAJ 1.3
```

**See also**

`getAllAttributesAsQNames()`

---

**getAllAttributesAsQNames**

```java
Iterator getAllAttributesAsQNames()
```

Returns an `Iterator` over all of the attributes in this `SOAPElement` as `QName` objects. The iterator can be used to get the attribute `QName`, which can then be passed to the method `getAttributeValue` to retrieve the value of each attribute.

**Returns:**

an iterator over the `QNames` of the attributes

**Since:**

SAAJ 1.3

**See Also:**

`getAllAttributes()`

---

```java
public String getNamespaceURI(String prefix)
```

**URI**

```java
prefix String
```

prefix
getNamespaceURI

String getNamespaceURI(String prefix)

Returns the URI of the namespace that has the given prefix.

Parameters:
prefix - a String giving the prefix of the namespace for which to search

Returns:
a String with the uri of the namespace that has the given prefix

public java.util.Iterator<E> getNamespacePrefixes()

String Iterator getNamespaceURI

return SOAPElement

generatePrefixes

Iterator getNamespacePrefixes()

Returns an Iterator over the namespace prefix strings declared by this element. The prefixes returned by this iterator can be passed to the method getNamespaceURI to retrieve the URI of each namespace.

Returns:
an iterator over the namespace prefixes in this SOAPElement object

public java.util.Iterator<E> getVisibleNamespacePrefixes()

String Iterator getNamespaceURI

return SOAPElement

since SAAJ 1.2
**getVisibleNamespacePrefixes**

```
Iterator getVisibleNamespacePrefixes()
```

Returns an `Iterator` over the namespace prefix `String`s visible to this element. The prefixes returned by this iterator can be passed to the method `getNamespaceURI` to retrieve the URI of each namespace.

Returns:
- an `Iterator` over the namespace prefixes are within scope of this `SOAPElement` object

Since:
- SAAJ 1.2

---

**publicjavax.xml.namespace.QName createQName(String localName, String prefix) throws SOAPException**

```
QName QName URI SOAPElement prefix QName
localName prefix
localName String prefix String
return QName localName prefix SOAPElement prefix
prefix #getNamespaceURI(String) exception
```

Throws: `SOAPException`

Since: SAAJ 1.3

---

**createQName**

```
QName createQName(String localName, String prefix) throws SOAPException
```

Creates a `QName` whose namespace URI is the one associated with the parameter, `prefix`, in the context of this `SOAPElement`. The remaining elements of the new `QName` are taken directly from the
Parameters:

localName - a String containing the local part of the name.
prefix - a String containing the prefix for the name.

Returns:

a QName with the specified localName and prefix, and with a namespace that is associated with the prefix in the context of this SOAPElement. This namespace will be the same as the one that would be returned by getNamespaceURI(String) if it were given prefix as its parameter.

Throws:

SOAPException - if the QName cannot be created.

Since:

SAAJ 1.3

public Name getElementName()   

SOAPElement  

return  

SOAPElement   Name

g getElementName

Name  getgetElementName()    

Returns the name of this SOAPElement object.

Returns:

a Name object with the name of this SOAPElement object

g public javax.xml.namespace.QName getElementQName()   

SOAPElement  qname

return  

SOAPElement   QName

since  

SAAJ 1.3

See also  

g getElementName()
getElementQName

QName getElementQName()

Returns the qname of this SOAPElement object.

Returns:
   a QName object with the qname of this SOAPElement object

Since:
   SAAJ 1.3

See Also:
   getElementName()

public SOAPElement setElementQName(javax.xml.namespace.QName newName) throws SOAPException

setElementQName(QName newName)

Changes the name of this Element to newName if possible. SOAP Defined elements such as SOAPEnvelope, SOAPHeader, SOAPBody etc. cannot have their names changed using this method. Any attempt to do so will result in a SOAPException being
thrown.

Callers should not rely on the element instance being renamed as is. Implementations could end up copying the content of the SOAPElement to a renamed instance.

**Parameters:**
- `newName` - the new name for the `Element`.

**Returns:**
The renamed Node

**Throws:**
- `SOAPException` - if changing the name of this `Element` is not allowed.

**Since:**
SAAJ 1.3

---

**public boolean removeAttribute(Name name)**

**name**
- Name

**return**
- true false

**See also**
- `removeAttribute(javax.xml.namespace.QName)`

---

**removeAttribute**

boolean **removeAttribute**(Name name)

Removes the attribute with the specified name.

**Parameters:**
- `name` - the `Name` object with the name of the attribute to be removed

**Returns:**
- true if the attribute was removed successfully; false if it was not removed

**See Also:**
- `removeAttribute(javax.xml.namespace.QName)`
public boolean removeAttribute(javax.xml.namespace.QName qname)

        qname

        return

        since

        See also

        removeAttribute(Name)

removeAttribute

boolean removeAttribute(QName qname)

    Removes the attribute with the specified qname.

    Parameters:
    qname - the QName object with the qname of the attribute to be removed

    Returns:
    true if the attribute was removed successfully; false if it was not

Since:
SAAJ 1.3
See Also:
removeAttribute(Name)

public boolean removeNamespaceDeclaration(String prefix)

        prefix

        return

        String

        true  false

removeNamespaceDeclaration

boolean removeNamespaceDeclaration(String prefix)
Removes the namespace declaration corresponding to the given prefix.

**Parameters:**
- `prefix` - a String giving the prefix for which to search

**Returns:**
- true if the namespace declaration was removed successfully;
- false if it was not

---

```java
public java.util.Iterator<E> getChildElements()
```

Returns an Iterator over all the immediate child [Nodes] of this element. This includes `javax.xml.soap.Text` objects as well as `SOAPElement` objects.

Calling this method may cause child `Element`, `SOAPElement` and `org.w3c.dom.Text` nodes to be replaced by `SOAPElement`, `SOAPHeaderElement`, `SOAPBodyElement` or `javax.xml.soap.Text` nodes as appropriate for the type of this parent node. As a result the calling application must treat any existing references to these child nodes that have been obtained through DOM APIs as invalid and either
discard them or refresh them with the values returned by this Iterator. This behavior can be avoided by calling the equivalent DOM APIs. See javax.xml.soap for more details.

**Returns:**
an iterator with the content of this SOAPElement object

```java
public java.util.Iterator<E> getChildElements(Name name)
```

ElementSOAPElement org.w3c.dom.Text SOAPElementSOAPHeaderElementSOAPBodyElement javax.xml.soap.Text DOM API Iterator

HREF="package-summary.html#package_description">javax.xml.soap

**name** Name

**return** SOAPElement Iterator

See also getChildElements(javax.xml.namespace.QName)

getChildElements

```java
Iterator getChildElements(Name name)
```

Returns an Iterator over all the immediate child Nodes of this element with the specified name. All of these children will be SOAPElement nodes.

Calling this method may cause child Element, SOAPElement and org.w3c.dom.Text nodes to be replaced by SOAPElement, SOAPHeaderElement, SOAPBodyElement or javax.xml.soap.Text nodes as appropriate for the type of this parent node. As a result the calling application must treat any existing references to these child nodes that have been obtained through DOM APIs as invalid and either discard them or refresh them with the values returned by this
Iterator. This behavior can be avoided by calling the equivalent DOM APIs. See javax.xml.soap for more details.

**Parameters:**
- `name` - a `Name` object with the name of the child elements to be returned

**Returns:**
- an `Iterator` object over all the elements in this `SOAPElement` object with the specified name

**See Also:**
- `get Child Elements(javax.xml.namespace.QName)`

```java
public java.util.Iterator<E>
getChildElements(javax.xml.namespace.QName qname)
```

Returns an `Iterator` over all the immediate child `Node`s of this element with the specified `qname`. All of these children will be `SOAPElement` nodes.
Calling this method may cause child Element, SOAPElement and org.w3c.dom.Text nodes to be replaced by SOAPElement, SOAPHeaderElement, SOAPBodyElement or javax.xml.soap.Text nodes as appropriate for the type of this parent node. As a result the calling application must treat any existing references to these child nodes that have been obtained through DOM APIs as invalid and either discard them or refresh them with the values returned by this Iterator. This behavior can be avoided by calling the equivalent DOM APIs. See javax.xml.soap for more details.

Parameters:
   qname - a QName object with the qname of the child elements to be returned

Returns:
   an Iterator object over all the elements in this SOAPElement object with the specified qname

Since:
   SAAJ 1.3

See Also:
   getChildElements(Name)

public void setEncodingStyle(String encodingStyle) throws SOAPException

Sets the encoding style for this SOAPElement object to one specified.
Parameters:

- `encodingStyle` - a String giving the encoding style

Throws:

- `IllegalArgumentException` - if there was a problem in the encoding style being set.
- `SOAPException` - if setting the `encodingStyle` is invalid for this `SOAPElement`.

See Also:

- `getEncodingStyle()`

---

```java
public String getEncodingStyle()
```

```java
SOAPElement

return String
```

See also `setEncodingStyle`

---

**getEncodingStyle**

```java
String getEncodingStyle()
```

Returns the encoding style for this `SOAPElement` object.

**Returns:**

- a String giving the encoding style

See Also:

- `setEncodingStyle(java.lang.String)`

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
<table>
<thead>
<tr>
<th>Summary: Nested</th>
<th>Field</th>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detail: Field</td>
<td>Constructor</td>
<td>Method</td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.soap  **Class SOAPElementFactory**

**java.lang.Object**

| javax.xml.soap.SOAPElementFactory |

---

**Deprecated.** - *Use javax.xml.soap.SOAPFactory for creating SOAPElements.*

SOAPElementFactory  SOAP  XML  SOAPHeader  SOAPBody

SOAPXML
deprecated - javax.xml.soap.SOAPFactory SOAPElements

See also  javax.xml.soap.SOAPFactory

**public class SOAPElementFactory**

extends **Object**

SOAPElementFactory is a factory for XML fragments that will eventually end up in the SOAP part. These fragments can be inserted as children of the SOAPHeader or SOAPBody or SOAPEnvelope.

Elements created using this factory do not have the properties of an element that lives inside a SOAP header document. These elements are copied into the XML document tree when they are inserted.

See Also:  SOAPFactory

---

**Method Summary**

| **create**(Name name) |
SOAPElement

**Deprecated.** Use `javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name)` instead

```java
SOAPElement create(String localName)

**Deprecated.** Use `javax.xml.soap.SOAPFactory.createElement(String)` instead

```java
SOAPElement create(String localName, String prefix, String uri)

**Deprecated.** Use `javax.xml.soap.SOAPFactory.createElement(String, String, String)` instead
```

```java
static SOAPElementFactory newInstance()

**Deprecated.** Creates a new instance of SOAPElementFactory
```

### Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

### Method Detail

#### public SOAPElement create(Name name) throws SOAPException

<table>
<thead>
<tr>
<th>Name</th>
<th>SOAPElement</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>name</code></td>
<td>XML</td>
</tr>
<tr>
<td><code>return</code></td>
<td>SOAPElement</td>
</tr>
</tbody>
</table>

**Deprecated** `javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name)`

**See also**

- `createElement(javax.xml.soap.Name)`,
- `createElement(javax.xml.namespace.QName)`

**create**
public SOAPElement create(Name name) throws SOAPException

Deprecated. Use javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name) instead

Create a SOAPElement object initialized with the given Name object.

Parameters:
  name - a Name object with the XML name for the new element

Returns:
  the new SOAPElement object that was created

Throws:
  SOAPException - if there is an error in creating the SOAPElement object

See Also:
  SOAPFactory.createElement(javax.xml.soap.Name),
  SOAPFactory.createElement(javax.xml.namespace.QName)

public SOAPElement create(String localName) throws SOAPException

SOAPElement

localName String
return SOAPElement

Throws SOAPException: SOAPElement

deprecated javax.xml.soap.SOAPFactory.createElement(String localName)

See also createElement(java.lang.String)

create

public SOAPElement create(String localName) throws SOAPException

Deprecated. Use javax.xml.soap.SOAPFactory.createElement(String localName)
instead

Create a SOAPElement object initialized with the given local name.

**Parameters:**
- `localName` - a String giving the local name for the new element

**Returns:**
- the new SOAPElement object that was created

**Throws:**
- SOAPException - if there is an error in creating the SOAPElement object

**See Also:**
- SOAPFactory.createElement(java.lang.String)

```java
public SOAPElement create(String localName, String prefix, String uri) throws SOAPException

URL SOAPElement
  localName String
  prefix SOAPElement
  uri URI

Throws SOAPException: SOAPElement

deprecated javax.xml.soap.SOAPFactory.createElement(String
  localName, String prefix, String uri)

See also createElement(java.lang.String, java.lang.String,
  java.lang.String)
```

**create**

```java
public SOAPElement create(String localName, String prefix, String uri) throws SOAPException

Deprecated. Use
javax.xml.soap.SOAPFactory.createElement(String localName,
  String prefix, String uri) instead
```
Create a new SOAPElement object with the given local name, prefix and uri.

Parameters:
- localName - a String giving the local name for the new element
- prefix - the prefix for this SOAPElement
- uri - a String giving the URI of the namespace to which the new element belongs

Throws:
- SOAPException - if there is an error in creating the SOAPElement object

See Also:
- SOAPFactory.createElement(java.lang.String, java.lang.String, java.lang.String)

```
public static SOAPElementFactory newInstance() throws SOAPException

SOAPElementFactory
return

Throws

SOAPException:

```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.soap  Interface SOAPEnvelope

All Superinterfaces:
   Element, Node, SOAPElement

public interface SOAPEnvelope
extends SOAPElement

Implements: SOAPElement

SOAPPart  SOAPHeader  SOAPBody  SOAPEnvelope
SOAPPart  SOAPMessage  SOAPEnvelope  SOAPBody  SOAPHeader
   SOAPBody  SOAPHeader  SOAPHeader
   SOAPEnvelope  se

SOAPPart  sp = message.getSOAPPart();
SOAPEnvelope  se = sp.getEnvelope();
SOAPHeader  sh = se.getHeader();
SOAPBody  sb = se.getBody();

SOAPEnvelope  se
   se.getBody().detachNode();

SOAPEnvelope  se
   se.addHeader();
   se.getBody().detachNode();

SOAPHeader  sh
   se.addBody();
The container for the SOAPHeader and SOAPBody portions of a SOAPPart object. By default, a SOAPMessage object is created with a SOAPPart object that has a SOAPEnvelope object. The SOAPEnvelope object by default has an empty SOAPBody object and an empty SOAPHeader object. The SOAPBody object is required, and the SOAPHeader object, though optional, is used in the majority of cases. If the SOAPHeader object is not needed, it can be deleted, which is shown later.

A client can access the SOAPHeader and SOAPBody objects by calling the methods SOAPEnvelope.getHeader and SOAPEnvelope.getBody. The following lines of code use these two methods after starting with the SOAPMessage object message to get the SOAPPart object sp, which is then used to get the SOAPEnvelope object se.

```java
SOAPPart sp = message.getSOAPPart();
SOAPEnvelope se = sp.getEnvelope();
SOAPHeader sh = se.getHeader();
SOAPBody sb = se.getBody();
```

It is possible to change the body or header of a SOAPEnvelope object by retrieving the current one, deleting it, and then adding a new body or header. The javax.xml.soap.Node method deleteNode deletes the XML element (node) on which it is called. For example, the following line of code deletes the SOAPBody object that is retrieved by the method getBody.

```java
se.getBody().detachNode();
```

To create a SOAPHeader object to replace the one that was removed, a client uses the method SOAPEnvelope.addHeader, which creates a new

```java
SOAPEnvelope se = ...; SOAPHeader sh = ...; se.addHeader(sh);
```
header and adds it to the SOAPEnvelope object. Similarly, the method addBody creates a new SOAPBody object and adds it to the SOAPEnvelope object. The following code fragment retrieves the current header, removes it, and adds a new one. Then it retrieves the current body, removes it, and adds a new one.

```java
SOAPPart sp = message.getSOAPPart();
SOAPEnvelope se = sp.getEnvelope();
se.getHeader().detachNode();
SOAPHeader sh = se.addHeader();
se.getBody().detachNode();
SOAPBody sb = se.addBody();
```

It is an error to add a SOAPBody or SOAPHeader object if one already exists.

The SOAPEnvelope interface provides three methods for creating Name objects. One method creates Name objects with a local name, a namespace prefix, and a namespace URI. The second method creates Name objects with a local name and a namespace prefix, and the third creates Name objects with just a local name. The following line of code, in which se is a SOAPEnvelope object, creates a new Name object with all three.

```java
Name name = se.createName("GetLastTradePrice", "WOMBAT", "http://www.wombat.org/trader");
```

---

**Field Summary**

**Fields inherited from interface org.w3c.dom.Node**

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,
DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,
DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,
DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,
DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,
ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,
PROCESSING_INSTRUCTION_NODE, TEXT_NODE
<table>
<thead>
<tr>
<th>Field Inherited from interface org.w3c.dom.*Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,</td>
</tr>
<tr>
<td>ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>SOAPBody</th>
<th>addBody()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a SOAPBody object and sets it as the SOAPBody object for this SOAPEnvelope object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SOAPHeader</th>
<th>addHeader()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a SOAPHeader object and sets it as the SOAPHeader object for this SOAPEnvelope object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>createName(String localName)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a new Name object initialized with the given local name.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>createName(String localName, String prefix, String uri)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SOAPBody</th>
<th>getBody()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the SOAPBody object associated with this SOAPEnvelope object.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SOAPHeader</th>
<th>getHeader()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns the SOAPHeader object for this SOAPEnvelope object.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface javax.xml.soap.*SOAPElement*

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>addAttribute, addAttribute, addChildElement, addChildElement, addChildElement, addChildElement, addChildElement, addChildElement, createQName, getAllAttributes, getAllAttributesAsQNames, getAttributeValue, getAttributeValue, getChildElements, getChildElements, getChildElements, getElementName, getElementQName, getEncodingStyle, getNamespacePrefixes,</td>
</tr>
</tbody>
</table>
**getNamespaceURI, getVisibleNamespacePrefixes, removeAttribute, removeAttribute, removeContents, removeNamespaceDeclaration, setElementQName, setEncodingStyle**

**Methods inherited from interface javax.xml.soap.Node**
detachNode, getParentElement, getValue, recycleNode, setParentElement, setValue

**Methods inherited from interface org.w3c.dom.Node**
appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, get主要内容未列出

**Methods inherited from interface org.w3c.dom.Element**
getAttribute, getAttributeNode, getAttributeNodeNS, getAttributeNS, getElementsByTagName, getElementsByTagNode, getElementsByTagNodeNS, getSchemaTypeInfo, getTagName, hasAttribute, hasAttributeNS, removeAttribute, removeAttributeNode, removeAttributeNS, setAttribute, setAttributeNode, setAttributeNodeNS, setIdAttribute, setIdAttributeNode, setIdAttributeNS

**Methods inherited from interface org.w3c.dom.Node**
appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData

**Method Detail**
public Name createName(String localName, String prefix, String uri) throws SOAPException

createName

Name createName(String localName, String prefix, String uri)
throws SOAPException

Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.

This factory method creates Name objects for use in the SOAP/XML document.

Parameters:
- localName - a String giving the local name
- prefix - a String giving the prefix of the namespace
- uri - a String giving the URI of the namespace

Returns:
- a Name object initialized with the given local name, namespace prefix, and namespace URI

Throws:
- SOAPException - if there is a SOAP error
SOAPException

SOAP/XML

createName

```java
Name createName(String localName)
throws SOAPException
```

Creates a new `Name` object initialized with the given local name.

This factory method creates `Name` objects for use in the SOAP/XML document.

**Parameters:**
- `localName` - a `String` giving the local name

**Returns:**
- a `Name` object initialized with the given local name

**Throws:**
- `SOAPException` - if there is a SOAP error

public `SOAPHeader` getHeader() throws `SOAPException`

```java
SOAPHeader getHeader()
throws SOAPException
```

**Return:**
- `SOAPHeader` null

**Throws:**
- `SOAPException` - `SOAPHeader`
**getHeader**

```java
SOAPHeader getHeader() throws SOAPException
```

Returns the SOAPHeader object for this SOAPEnvelope object.

A new SOAPMessage object is by default created with a SOAPEnvelope object that contains an empty SOAPHeader object. As a result, the method getHeader will always return a SOAPHeader object unless the header has been removed and a new one has not been added.

**Returns:**
the SOAPHeader object or null if there is none

**Throws:**
SOAPException - if there is a problem obtaining the SOAPHeader object

---

**public** SOAPBody getBody() throws SOAPException

```java
SOAPEnvelope SOAPBody
```

```java
SOAPBody SOAPEnvelope SOAPMessage
getBody SOAPBody
return
SOAPEnvelope SOAPBody null

**Throws**
SOAPException: SOAPBody

---

**getBody**

```java
SOAPBody getBody() throws SOAPException
```

Returns the SOAPBody object associated with this SOAPEnvelope object.

A new SOAPMessage object is by default created with a SOAPEnvelope object that contains an empty SOAPBody object. As a result, the method getBody will always return a SOAPBody object unless the body
has been removed and a new one has not been added.

**Returns:**
the SOAPBody object for this SOAPEnvelope object or null if there is none

**Throws:**
SOAPException - if there is a problem obtaining the SOAPBody object

---

```java
public SOAPHeader addHeader() throws SOAPException

SOAPHeader SOAPEnvelope SOAPHeader
```

---

**public SOAPHeader addHeader() throws SOAPException**

**SOAPHeader** **SOAPEnvelope** **SOAPHeader**

**return** **SOAPHeader**

**Throws** **SOAPException:** **SOAPEnvelope** **SOAPHeader**

---

**addHeader**

**SOAPHeader addHeader()** **throws SOAPException**

Creates a SOAPHeader object and sets it as the SOAPHeader object for this SOAPEnvelope object.

It is illegal to add a header when the envelope already contains a header. Therefore, this method should be called only after the existing header has been removed.

**Returns:**
the new SOAPHeader object

**Throws:**
SOAPException - if this SOAPEnvelope object already contains a valid SOAPHeader object
public SOAPBody addBody() throws SOAPException

SOAPBody
SOAPEnvelope
SOAPBody

return SOAPBody

Throws SOAPException: SOAPEnvelope SOAPBody

addBody

SOAPBody addBody() throws SOAPException

Creates a SOAPBody object and sets it as the SOAPBody object for this SOAPEnvelope object.

It is illegal to add a body when the envelope already contains a body. Therefore, this method should be called only after the existing body has been removed.

Returns: the new SOAPBody object

Throws: 
SOAPException - if this SOAPEnvelope object already contains a valid SOAPBody object
javax.xml.soap  Class SOAPException

java.lang.Object  ↳ java.lang.Throwable  ↳ java.lang.Exception  ↳ javax.xml.soap.SOAPException

All Implemented Interfaces:
Serializable

public class SOAPException
extends Exception

Extends: Throwable > Exception

SOAP Throwable SOAPException String Throwable

SOAPException Throwable /

An exception that signals that a SOAP exception has occurred. A
SOAPException object may contain a String that gives the reason for the exception, an embedded Throwable object, or both. This class provides
methods for retrieving reason messages and for retrieving the embedded Throwable object.

Typical reasons for throwing a SOAPException object are problems such as difficulty setting a header, not being able to send a message, and not
being able to get a connection with the provider. Reasons for embedding a Throwable object include problems such as input/output errors or a
parsing problem, such as an error in parsing a header.

See Also:
Serialized Form
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SOAPException()</code></td>
<td>Constructs a <code>SOAPException</code> object with no reason or embedded Throwable object.</td>
</tr>
<tr>
<td><code>SOAPException(String reason)</code></td>
<td>Constructs a <code>SOAPException</code> object with the given <code>String</code> as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><code>SOAPException(String reason, Throwable cause)</code></td>
<td>Constructs a <code>SOAPException</code> object with the given <code>String</code> as the reason for the exception being thrown and the given <code>Throwable</code> object as an embedded exception.</td>
</tr>
<tr>
<td><code>SOAPException(Throwable cause)</code></td>
<td>Constructs a <code>SOAPException</code> object initialized with the given <code>Throwable</code> object.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getCause()</code></td>
<td>Returns the <code>Throwable</code> object embedded in this <code>SOAPException</code> if there is one.</td>
</tr>
<tr>
<td><code>getMessage()</code></td>
<td>Returns the detail message for this <code>SOAPException</code> object.</td>
</tr>
<tr>
<td><code>initCause(Throwable cause)</code></td>
<td>Initializes the <code>cause</code> field of this <code>SOAPException</code> object with the given <code>Throwable</code> object.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Throwable`:
- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
Constructor Detail

public SOAPException()

SOAPException

public SOAPException()

Constructs a SOAPException object with no reason or embedded Throwable object.

public SOAPException(String reason)

SOAPException String reason

SOAPException

public SOAPException(String reason)

Constructs a SOAPException object with the given String as the reason for the exception being thrown.

Parameters:
reason - a description of what caused the exception

public SOAPException(String reason, Throwable cause)

SOAPException String Throwable reason cause

SOAPException Throwable
SOAPException

public SOAPException(String reason, Throwable cause)

Constructs a SOAPException object with the given string as the reason for the exception being thrown and the given Throwable object as an embedded exception.

Parameters:
- reason - a description of what caused the exception
- cause - a Throwable object that is to be embedded in this SOAPException object

public SOAPException(Throwable cause)

Constructs a SOAPException object initialized with the given Throwable object.

Method Detail

public String getMessage()

Throws: SOAPException
public String getMessage()

Returns the detail message for this SOAPException object.

If there is an embedded Throwable object, and if the SOAPException object has no detail message of its own, this method will return the detail message from the embedded Throwable object.

Overrides:
getMessage in class Throwable

Returns:
the error or warning message for this SOAPException or, if it has none, the message of the embedded Throwable object, if there is one

public Throwable getCause()

SOAPException Throwable null
return Throwable null

getCause

gpublic Throwable getCause()

Returns the Throwable object embedded in this SOAPException if there is one. Otherwise, this method returns null.

Overrides:
getCause in class Throwable

Returns:
the embedded Throwable object or null if there is none

public Throwable initCause(Throwable cause)

Throwable SOAPException cause
initCause

public Throwable initCause(Throwable cause)

Initializes the cause field of this SOAPException object with the given Throwable object.

This method can be called at most once. It is generally called from within the constructor or immediately after the constructor has returned a new SOAPException object. If this SOAPException object was created with the constructor SOAPException(Throwable) or SOAPException(String, Throwable), meaning that its cause field already has a value, this method cannot be called even once.

Overrides:

initCause in class Throwable

Parameters:

cause - the Throwable object that caused this SOAPException object to be thrown. The value of this parameter is saved for later retrieval by the getCause() method. A null value is permitted and indicates that the cause is nonexistent or unknown.

Returns:

a reference to this SOAPException instance

Throws:

IllegalArgumentException - if cause is this Throwable object. (A Throwable object cannot be its own cause.)
IllegalStateException - if the cause for this SOAPException object has already been initialized
public abstract class SOAPFactory extends Object

SOAPFactory is a factory for creating various objects that exist in the SOAP XML tree. SOAPFactory can be used to create XML fragments that will eventually end up in the SOAP part. These fragments can be inserted as children of the SOAPHeaderElement or SOAPBodyElement or SOAPEnvelope or other SOAPElement objects. SOAPFactory also has methods to create javax.xml.soap.Detail objects as well as java.xml.soap.Name objects.

**Constructor Summary**

| SOAPFactory() |

**Method Summary**

<p>| abstract Detail createDetail() | Creates a new Detail object which serves as a container for DetailEntry objects. |
| SOAPElement createElement(Element domElement) | Creates a SOAPElement object from an existing DOM Element. |
| createElement(Name name) | |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPElement</td>
<td><code>createElement(QName qname)</code> Creates a SOAPElement object initialized with the given QName object.</td>
</tr>
<tr>
<td>SOAPElement</td>
<td><code>createElement(String localName)</code> Creates a SOAPElement object initialized with the given local name.</td>
</tr>
<tr>
<td>SOAPElement</td>
<td><code>createElement(String localName, String prefix, String uri)</code> Creates a new SOAPElement object with the given local name, prefix and uri.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>createFault()</code> Creates a new default SOAPFault object.</td>
</tr>
<tr>
<td>SOAPFault</td>
<td><code>createFault(String reasonText, QName faultCode)</code> Creates a new SOAPFault object initialized with the given reasonText and faultCode.</td>
</tr>
<tr>
<td>Name</td>
<td><code>createName(String localName)</code> Creates a new Name object initialized with the given local name.</td>
</tr>
<tr>
<td>Name</td>
<td><code>createName(String localName, String prefix, String uri)</code> Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.</td>
</tr>
<tr>
<td>SOAPFactory</td>
<td><code>newInstance()</code> Creates a new SOAPFactory object that is an instance of the default implementation (SOAP 1.1). This method uses the following ordered lookup procedure to determine the SOAPFactory implementation class to load: Use the javax.xml.soap.SOAPFactory system property.</td>
</tr>
<tr>
<td>SOAPFactory</td>
<td><code>newInstance(String protocol)</code> Creates a new SOAPFactory object that is an instance of the specified implementation, this method uses the SAAJMetaFactory to locate the implementation class and create the SOAPFactory instance.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

public SOAPFactory()

SOAPFactory

public SOAPFactory()

Method Detail

public SOAPElemen creates an Element from an existing DOM Element. If the DOM Element that is passed in as an argument is already a
SOAPElement then this method must return it unmodified without any further work. Otherwise, a new SOAPElement is created and a deep copy is made of the domElement argument. The concrete type of the return value will depend on the name of the domElement argument. If any part of the tree rooted in domElement violates SOAP rules, a SOAPException will be thrown.

Parameters:
  domElement -- the Element to be copied.

Returns:
a new SOAPElement that is a copy of domElement.

Throws:
  SOAPException - if there is an error in creating the SOAPElement object

Since:
  SAAJ 1.3

```java
abstract public SOAPElement createElement(Name name)
throws SOAPException
```

`Soap 1.2` SOAPElement "{http://www.w3.org/2003/05/soap-envelope}Envelope"

SOAPElement SOAP 1.2 SOAPEnvelope

name XML Name

return SOAPElement

Throws SOAPException: SOAPElement

See also createElement(javax.xml.namespace.QName)

createElement

```java
public abstract SOAPElement createElement(Name name)
throws SOAPException
```

Creates a SOAPElement object initialized with the given Name object. The concrete type of the return value will depend on the name given to the new SOAPElement. For instance, a new SOAPElement with the name "{http://www.w3.org/2003/05/soap-envelope}Envelope" would
cause a SOAPEnvelope that supports SOAP 1.2 behavior to be created.

**Parameters:**
- `name` - a Name object with the XML name for the new element

**Returns:**
- the new SOAPElement object that was created

**Throws:**
- SOAPException - if there is an error in creating the SOAPElement object

**See Also:**
- `createElement(javax.xml.namespace.QName)`

---

```java
public SOAPElement createElement(javax.xml.namespace.QName qname)
throws SOAPException
```

QName SOAPElement SOAPElement "
{http://www.w3.org/2003/05/soap-envelope}Envelope"

SOAPElement SOAP 1.2 SOAPEnvelope
qname XML QName
return SOAPElement

Throws SOAPException: SOAPElement

since SAAJ 1.3

See also `createElement(Name)`

---

**createElement**

```java
public SOAPElement createElement(QName qname)
throws SOAPException
```

Creates a SOAPElement object initialized with the given QName object. The concrete type of the return value will depend on the name given to the new SOAPElement. For instance, a new SOAPElement with the name "{http://www.w3.org/2003/05/soap-envelope}Envelope" would cause a SOAPEnvelope that supports SOAP 1.2 behavior to be created.
Parameters:
qname - a QName object with the XML name for the new element

Returns:
the new SOAPElement object that was created

Throws:
SOAPException - if there is an error in creating the SOAPElement object

Since:
SAAJ 1.3

See Also:
createElement(Name)

abstract public SOAPElement createElement(String localName) throws SOAPException

SOAPElement

localName

String

return

SOAPElement

Throws

SOAPException:

SOAPElement

createElement

public abstract SOAPElement createElement(String localName) throws SOAPException

Creates a SOAPElement object initialized with the given local name.

Parameters:
localName - a String giving the local name for the new element

Returns:
the new SOAPElement object that was created

Throws:
SOAPException - if there is an error in creating the SOAPElement object

abstract public SOAPElement createElement(String
createElement

public abstract SOAPElement createElement(String localName, String prefix, String uri) throws SOAPException

Creates a new SOAPElement object with the given local name, prefix and uri. The concrete type of the return value will depend on the name given to the new SOAPElement. For instance, a new SOAPElement with the name "{http://www.w3.org/2003/05/soap-envelope}Envelope" would cause a SOAPEnvelope that supports SOAP 1.2 behavior to be created.

Parameters:
- localName - a String giving the local name for the new element
- prefix - the prefix for this SOAPElement
- uri - a String giving the URI of the namespace to which the new element belongs

Throws:
- SOAPException - if there is an error in creating the SOAPElement object
createDetail

public abstract Detail createDetail() throws SOAPException

Creates a new Detail object which serves as a container for DetailEntry objects.

This factory method creates Detail objects for use in situations where it is not practical to use the SOAPFault abstraction.

Returns:
a Detail object

Throws:
   SOAPException - if there is a SOAP error
   UnsupportedOperationException - if the protocol specified for the SOAPFactory was DYNAMIC_SOAP_PROTOCOL

abstract public SOAPFault createFault(String reasonText, javax.xml.namespace.QName faultCode) throws SOAPException

reasonText   faultCode   SOAPFault
reasonText   ReasonText/FaultString
faultCode    FaultCode
return       SOAPFault
Throws       SOAPException: SOAP
since        SAAJ 1.3

createFault
public abstract SOAPFault createFault(String reasonText, QName faultCode) throws SOAPException

Creates a new SOAPFault object initialized with the given reasonText and faultCode

Parameters:
- reasonText - the ReasonText/FaultString for the fault
- faultCode - the FaultCode for the fault

Returns:
a SOAPFault object

Throws:
- SOAPException - if there is a SOAP error

Since:
SAAJ 1.3

abstract public SOAPFault createFault() throws SOAPException

SOAPFault

return SOAPFault

Throws

SOAPException: SOAP

since SAAJ 1.3

createFault

public abstract SOAPFault createFault() throws SOAPException

Creates a new default SOAPFault object

Returns:
a SOAPFault object

Throws:
- SOAPException - if there is a SOAP error

Since:
SAAJ 1.3
abstract public Name createName(String localName, String prefix, String uri) throws SOAPException

createName

public abstract Name createName(String localName, String prefix, String uri)

throws SOAPException

Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.

This factory method creates Name objects for use in situations where it is not practical to use the SOAPEnvelope abstraction.

Parameters:
- localName - a String giving the local name
- prefix - a String giving the prefix of the namespace
- uri - a String giving the URI of the namespace

Returns:
- a Name object initialized with the given local name, namespace prefix, and namespace URI

Throws:
- SOAPException - if there is a SOAP error
abstract public Name createName(String localName) throws SOAPException

createName

public abstract Name createName(String localName) throws SOAPException

Creates a new Name object initialized with the given local name.

This factory method creates Name objects for use in situations where it is not practical to use the SOAPEnvelope abstraction.

Parameters:
localName - a String giving the local name

Returns:
a Name object initialized with the given local name

Throws:
SOAPException - if there is a SOAP error

public static SOAPFactory newInstance() throws SOAPException

SOAPFactory (SOAP 1.1) SOAP

- javax.xml.soap.SOAPFactory
- JRE "lib/jaxm.properties" java.util.Properties
- Services API JAR Services API jar
META-INF/services/javax.xml.soap.SOAPFactory

- **SAAJMetaFactory**  SOAPFactory

```
return SOAPFactory
Throws SOAPException: SOAPFactory
See also javax.xml.soap.SAAJMetaFactory
```

**newInstance**

```
public static SOAPFactory newInstance() throws SOAPException
```

Creates a new SOAPFactory object that is an instance of the default implementation (SOAP 1.1). This method uses the following ordered lookup procedure to determine the SOAPFactory implementation class to load:

- Use the javax.xml.soap.SOAPFactory system property.
- Use the properties file "lib/jaxm.properties" in the JRE directory. This configuration file is in standard java.util.Properties format and contains the fully qualified name of the implementation class with the key being the system property defined above.
- Use the Services API (as detailed in the JAR specification), if available, to determine the classname. The Services API will look for a classname in the file META-INF/services/javax.xml.soap.SOAPFactory in jars available to the runtime.
- Use the SAAJMetaFactory instance to locate the SOAPFactory implementation class.

**Returns:**
a new instance of a SOAPFactory

**Throws:**
SOAPException - if there was an error creating the default SOAPFactory

**See Also:**
SAAJMetaFactory
public static SOAPFactory newInstance(String protocol) throws SOAPException

SOAPFactory SAAJMetaFactory

SOAPFactory return SOAPFactory

protocol DYNAMIC_SOAP_PROTOCOL DEFAULT_SOAP_PROTOCOL
SOAP_1_1_PROTOCOL SOAP_1_2_PROTOCOL

Throws SOAPException: SOAPFactory

since SAAJ 1.3
See also javax.xml.soap.SAAJMetaFactory

newInstance

public static SOAPFactory newInstance(String protocol) throws SOAPException

Creates a new SOAPFactory object that is an instance of the specified implementation, this method uses the SAAJMetaFactory to locate the implementation class and create the SOAPFactory instance.

Parameters:

protocol - a string constant representing the protocol of the specified SOAP factory implementation. May be either DYNAMIC_SOAP_PROTOCOL, DEFAULT_SOAP_PROTOCOL (which is the same as) SOAP_1_1_PROTOCOL, or SOAP_1_2_PROTOCOL.

Returns:
a new instance of a SOAPFactory

Throws:

SOAPException - if there was an error creating the specified SOAPFactory

Since:
SAAJ 1.3
See Also:
SAAJMetaFactory
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.soap Interface SOAPFault

All Superinterfaces:
   Element, Node, SOAPBodyElement, SOAPElement

public interface SOAPFault
extends SOAPBodyElement

Implements: SOAPBodyElement

SOAPBody / SOAPMessage

SOAPFault SOAP 1.1 SOAP 1.2 MessageFactory

An element in the SOAPBody object that contains error and/or status information. This information may relate to errors in the SOAPMessage object or to problems that are not related to the content in the message itself. Problems not related to the message itself are generally errors in processing, such as the inability to communicate with an upstream server.

Depending on the protocol specified while creating the MessageFactory instance, a SOAPFault has sub-elements as defined in the SOAP 1.1/SOAP 1.2 specification.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
</tbody>
</table>
Fields inherited from interface org.w3c.dom.Node
ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE

Method Summary

Detail
addDetail() Creates an optional Detail object and sets it as the
Detail object for this SOAPFault object.

void
addFaultReasonText(String text, Locale locale) Appends or replaces a Reason Text item containing the
specified text message and an xml:lang derived from locale.

void
appendFaultSubcode(QName subcode) Adds a Subcode to the end of the sequence of Subcodes
contained by this SOAPFault.

Detail
getDetail() Returns the optional detail element for this SOAPFault
object.

String
getFaultActor() Gets the fault actor for this SOAPFault object.

String
getFaultCode() Gets the fault code for this SOAPFault object.

Name
getFaultCodeAsName() Gets the mandatory SOAP 1.1 fault code for this
SOAPFault object as a SAAJ Name object.

QName
getFaultCodeAsQName() Gets the fault code for this SOAPFault object as a QName
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getFaultNode()</code></td>
<td>Returns the optional Node element value for this SOAPFault object.</td>
</tr>
<tr>
<td><code>getFaultReasonLocales()</code></td>
<td>Returns an Iterator over a distinct sequence of Locales for which there are associated Reason Text items.</td>
</tr>
<tr>
<td><code>getFaultReasonText(Locale locale)</code></td>
<td>Returns the Reason Text associated with the given Locale.</td>
</tr>
<tr>
<td><code>getFaultReasonTexts()</code></td>
<td>Returns an Iterator over a sequence of String objects containing all of the Reason Text items for this SOAPFault.</td>
</tr>
<tr>
<td><code>getFaultRole()</code></td>
<td>Returns the optional Role element value for this SOAPFault object.</td>
</tr>
<tr>
<td><code>getFaultString()</code></td>
<td>Gets the fault string for this SOAPFault object.</td>
</tr>
<tr>
<td><code>getFaultStringLocale()</code></td>
<td>Gets the locale of the fault string for this SOAPFault object.</td>
</tr>
<tr>
<td><code>getFaultSubcodes()</code></td>
<td>Gets the Subcodes for this SOAPFault as an iterator over QNames.</td>
</tr>
<tr>
<td><code>hasDetail()</code></td>
<td>Returns true if this SOAPFault has a Detail Subelement and false otherwise.</td>
</tr>
<tr>
<td><code>removeAllFaultSubcodes()</code></td>
<td>Removes any Subcodes that may be contained by this SOAPFault.</td>
</tr>
<tr>
<td><code>setFaultActor(String faultActor)</code></td>
<td>Sets this SOAPFault object with the given fault actor.</td>
</tr>
<tr>
<td><code>setFaultCode(Name faultCodeQName)</code></td>
<td>Sets this SOAPFault object with the given fault code.</td>
</tr>
<tr>
<td><code>setFaultCode(QName faultCodeQName)</code></td>
<td>Sets this SOAPFault object with the given fault code.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>void setFaultCode(String faultCode)</code></td>
<td>Sets this SOAPFault object with the given fault code.</td>
</tr>
<tr>
<td><code>void setFaultNode(String uri)</code></td>
<td>Creates or replaces any existing Node element value for this SOAPFault object.</td>
</tr>
<tr>
<td><code>void setFaultRole(String uri)</code></td>
<td>Creates or replaces any existing Role element value for this SOAPFault object.</td>
</tr>
<tr>
<td><code>void setFaultString(String faultString)</code></td>
<td>Sets the fault string for this SOAPFault object to the given string.</td>
</tr>
<tr>
<td><code>void setFaultString(String faultString, Locale locale)</code></td>
<td>Sets the fault string for this SOAPFault object to the given string and localized to the given locale.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.soap.SOAPElement`:
- `addAttribute`, `addAttribute`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addNamespaceDeclaration`, `addTextNode`, `createQName`, `getAllAttributes`, `getAllAttributesAsQNames`, `getAttributeValue`, `getAttributeValue`, `getChildElements`, `getchildElements`, `getElementQName`, `getEncodingStyle`, `getNamespacePrefixes`, `getNamespaceURI`, `getVisibleNamespacePrefixes`, `removeAttribute`, `removeContents`, `removeNamespaceDeclaration`, `setElementQName`, `setEncodingStyle`

Methods inherited from interface `javax.xml.soap.Node`:
- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

Methods inherited from interface `org.w3c.dom.Node`:
- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildNodes`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getParentNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`
public void setFaultCode(Name faultCodeQName) throws SOAPException

    SOAPFault

SOAP 1.1 (mandatory)

SOAPEnvelope se = ...;
// Create a qualified name in the SOAP namespace with a local
// of "Client". Note that prefix parameter is optional and is
// here which causes the implementation to use an appropriate
Name qname = se.createName("Client", null,
SOAPConstants.URI_NS_SOAP_ENVELOPE);
SOAPFault fault = ...;
fault.setFaultCode(qname);
setFaultCode

void setFaultCode(Name faultCodeQName) throws SOAPException

Sets this SOAPFault object with the given fault code.

Fault codes, which give information about the fault, are defined in the SOAP 1.1 specification. A fault code is mandatory and must be of type Name. This method provides a convenient way to set a fault code. For example,

SOAPEnvelope se = ...;
// Create a qualified name in the SOAP namespace with a localName
// of "Client". Note that prefix parameter is optional and is r
// here which causes the implementation to use an appropriate p
Name qname = se.createName("Client", null,
    SOAPConstants.URI_NS_SOAP_ENVELOPE);

SOAPFault fault = ...;
fault.setFaultCode(qname);

It is preferable to use this method over setFaultCode(String).

Parameters:
    faultCodeQName - a Name object giving the fault code to be set. It
    must be namespace qualified.

Throws:
    SOAPException - if there was an error in adding the faultcode
    element to the underlying XML tree.

Since:
    SAAJ 1.2

See Also:
    getFaultCodeAsName()
public void setFaultCode(javax.xml.namespace.QName faultCodeQName) throws SOAPException

faultCodeQName

Throws

SOAPException: faultcode XML

since

SAAJ 1.3

See also
gFaultCodeAsQName, setFaultCode(Name), getFaultCodeAsQName()

setFaultCode

void setFaultCode(QName faultCodeQName)

throws SOAPException

Sets this SOAPFault object with the given fault code. It is preferable to use this method over setFaultCode(Name).

Parameters:

faultCodeQName - a QName object giving the fault code to be set. It must be namespace qualified.

Throws:

SOAPException - if there was an error in adding the faultcode element to the underlying XML tree.

Since:

SAAJ 1.3

See Also:
gFaultCodeAsQName(), setFaultCode(Name), getFaultCodeAsQName()
**setFaultCode(String)**

Sets this SOAPFault object with the give fault code.

Fault codes, which given information about the fault, are defined in the SOAP 1.1 specification. This element is mandatory in SOAP 1.1. Because the fault code is required to be a QName it is preferable to use the `setFaultCode(Name)` form of this method.

**Parameters:**
- `faultCode` - a String giving the fault code to be set. It must be of the form "prefix:localName" where the prefix has been defined in a namespace declaration.

**Throws:**
- `SOAPException` - if there was an error in adding the `faultCode` to the underlying XML tree.

**See Also:**
- `setFaultCode(Name)`, `getFaultCode()`,
- `SOAPElement.addNamespaceDeclaration(java.lang.String, java.lang.String)`

---

**public Name getFaultCodeAsName()**

SAAJ Name SOAPFault SOAP 1.1 SOAP 1.1 "faultcode" QName SAAJ Name QName

gFaultCode

return Name
getFaultCodeAsName

Name getFaultCodeAsName()

Gets the mandatory SOAP 1.1 fault code for this SOAPFault object as a SAAJ Name object. The SOAP 1.1 specification requires the value of the "faultcode" element to be of type QName. This method returns the content of the element as a QName in the form of a SAAJ Name object. This method should be used instead of the getFaultCode method since it allows applications to easily access the namespace name without additional parsing.

Returns:
   a Name representing the faultcode

Since:
   SAAJ 1.2

See Also:
   setFaultCode(Name)

public javax.xml.namespace.QName getFaultCodeAsQName

Get the fault code for this SOAPFault object as a QName object.
Returns:
   a QName representing the faultcode

Since:
   SAAJ 1.3

See Also:
   setFaultCode(QName)

---

public java.util.Iterator<E> getFaultSubcodes()

QNames    SOAPFault Subcode

   return    QNames    Iterator    Iterator    remove    Subcode

   Subcode

Throws: UnsupportedOperationException: SOAP 1.2 Subcode
   since: SAAJ 1.3

---

getFaultSubcodes

   Iterator getFaultSubcodes()

   Gets the Subcodes for this SOAPFault as an iterator over QNames.

Returns:
   an Iterator that accesses a sequence of QNames. This Iterator should not support the optional remove method. The order in which the Subcodes are returned reflects the hierarchy of Subcodes present in the fault from top to bottom.

Throws:
   UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Subcode.

Since:
   SAAJ 1.3

---

public void removeAllFaultSubcodes()

   SOAPFault Subcode appendFaultSubcode

getFaultSubcodes

   Throws UnsupportedOperationException: SOAP 1.2 Subcode
removeAllFaultSubcodes

void removeAllFaultSubcodes()

Removes any Subcodes that may be contained by this SOAPFault. Subsequent calls to getFaultSubcodes will return an empty iterator until a call to appendFaultSubcode is made.

Throws:
- `UnsupportedOperationException` - if this message does not support the SOAP 1.2 concept of Subcode.

Since:
- SAAJ 1.3

---

public void appendFaultSubcode(javax.xml.namespace.QName subcode) throws SOAPException

Subcode SOAPFault Subcode SOAP 1.2
Subcode SOAP Fault Code
subcode Subcode QName

Throws
- SOAPException: Subcode
Throws `UnsupportedOperationException`: SOAP 1.2 Subcode
since SAAJ 1.3

---

appendFaultSubcode

void appendFaultSubcode(QName subcode)

throws SOAPException

Adds a Subcode to the end of the sequence of Subcodes contained by this SOAPFault. Subcodes, which were introduced in SOAP 1.2, are represented by a recursive sequence of subelements rooted in the mandatory Code subelement of a SOAP Fault.
Parameters:
  subcode - a QName containing the Value of the Subcode.

Throws:
  SOAPException - if there was an error in setting the Subcode
  UnsupportedOperationException - if this message does not
  support the SOAP 1.2 concept of Subcode.

Since:
  SAAJ 1.3

public String getFaultCode()

SOAPFault
return
See also
getFaultCodeAsName, setFaultCode

getFaultCode

String getFaultCode()

Gets the fault code for this SOAPFault object.

Returns:
  a String with the fault code

See Also:
  getFaultCodeAsName(), setFaultCode(javax.xml.soap.Name)

public void setFaultActor(String faultActor) throws
SOAPException (actor)

SOAPFault

SOAP 1.2

#setFaultRole(String)

faultActor

Throws

SOAPException: faultActor XML
setFaultActor

void setFaultActor(String faultActor) throws SOAPException

Sets this SOAPFault object with the given fault actor.

The fault actor is the recipient in the message path who caused the fault to happen.

If this SOAPFault supports SOAP 1.2 then this call is equivalent to setFaultRole(String)

Parameters:
  faultActor - a String identifying the actor that caused this SOAPFault object

Throws:
  SOAPException - if there was an error in adding the faultActor to the underlying XML tree.

See Also:
  getFaultActor()

public String getFaultActor()

SOAPFault

SOAPFault  SOAP 1.2  #getFaultRole()

return

See also

getFaultActor

String getFaultActor()
Gets the fault actor for this SOAPFault object.

If this SOAPFault supports SOAP 1.2 then this call is equivalent to `getFaultRole()`

**Returns:**

- A String giving the actor in the message path that caused this SOAPFault object

**See Also:**
- `setFaultActor(java.lang.String)`

```java
default void setFaultString(String faultString) throws SOAPException
```

**SOAPFault**

`SOAPFault`  **SOAP 1.2**

```
addFaultReasonText(faultString, Locale.getDefault());
```

**faultString** String

**Throws**  
- `SOAPException`: faultString XML

**See also**
- `getFaultString`

**setFaultString**

```java
void setFaultString(String faultString) throws SOAPException
```

Sets the fault string for this SOAPFault object to the given string.

If this SOAPFault is part of a message that supports SOAP 1.2 then this call is equivalent to:

```
addFaultReasonText(faultString, Locale.getDefault());
```
faultString - a String giving an explanation of the fault

Throws:

SOAPException - if there was an error in adding the faultString to the underlying XML tree.

See Also:

getFaultString()

public void setFaultString(String faultString, java.util.Locale locale) throws SOAPException

SOAPFault

SOAPFault

SOAP 1.2

addFaultReasonText(faultString, locale);

setFaultString

void setFaultString(String faultString, Locale locale) throws SOAPException

Sets the fault string for this SOAPFault object to the given string and localized to the given locale.

If this SOAPFault is part of a message that supports SOAP 1.2 then this call is equivalent to:

addFaultReasonText(faultString, locale);
Parameters:
- faultString - a String giving an explanation of the fault
- locale - a Locale object indicating the native language of the faultString

Throws:
- SOAPException - if there was an error in adding the faultString to the underlying XML tree.

Since:
- SAAJ 1.2

See Also:
- getFaultString()

public String getFaultString()

SOAPFault

SOAPFault SOAP 1.2

String reason = null;
try {
    reason = (String) getFaultReasonTexts().next();
} catch (SOAPException e) {}
return reason;

return

See also setFaultString(String), setFaultString(String, Locale)

g FaultString

String getFaultString()

Gets the fault string for this SOAPFault object.

If this SOAPFault is part of a message that supports SOAP 1.2 then this call is equivalent to:

    String reason = null;
    try {

reason = (String) getFaultReasonTexts().next();
} catch (SOAPException e) {}
return reason;

Returns:

a String giving an explanation of the fault

See Also:

setFaultString(String), setFaultString(String, Locale)

public java.util.Locale getFaultStringLocale()

SOAPFault

SOAPFault  SOAP 1.2

Locale locale = null;
try {
locale = (Locale) getFaultReasonLocales().next();
} catch (SOAPException e) {
return locale;

return Locale null
since SAAJ 1.2
See also setFaultString(String, Locale)

getFaultStringLocale

Locale getFaultStringLocale()

Gets the locale of the fault string for this SOAPFault object.

If this SOAPFault is part of a message that supports SOAP 1.2 then this call is equivalent to:

Locale locale = null;
try {
    locale = (Locale) getFaultReasonLocales().next();
} catch (SOAPException e) {}
return locale;

**Returns:**

A `Locale` object indicating the native language of the fault string or `null` if no locale was specified

**Since:**

SAAJ 1.2

**See Also:**

`setFaultString(String, Locale)`

---

**public boolean hasDetail()**

SOAPFault Detail true false

`(getDetail() != null)`

```
return SOAPFault Detail true false
since SAAJ 1.3
```

**hasDetail**

```java
boolean hasDetail()
```

Returns true if this `SOAPFault` has a `Detail` subelement and false otherwise. Equivalent to `(getDetail() != null)`.

**Returns:**

true if this `SOAPFault` has a `Detail` subelement and false otherwise.

**Since:**

SAAJ 1.3

---

**public `Detail` getDetail()**

SOAPFault detail

```
Detail SOAP 1.1 Fault
return
```

SOAPBodyElement Detail null
getDetail

**Detail getDetail()**

Returns the optional detail element for this SOAPFault object.

A `Detail` object carries application-specific error information, the scope of the error information is restricted to faults in the `SOAPBodyElement` objects if this is a SOAP 1.1 Fault.

**Returns:**

a `Detail` object with application-specific error information if present, null otherwise

---

**public** `Detail` **addDetail() throws SOAPException**

**Detail** `SOAPFault` **Detail**

**return**

**Throws** `SOAPException`: `SOAPFault` **Detail**

---

addDetail

**Detail addDetail() throws SOAPException**

Creates an optional `Detail` object and sets it as the `Detail` object for this `SOAPFault` object.

It is illegal to add a detail when the fault already contains a detail. Therefore, this method should be called only after the existing detail has been removed.

**Returns:**
the new Detail object

**Throws:**
- **SOAPException** - if this SOAPFault object already contains a valid Detail object

```java
public java.util.Iterator<E> getFaultReasonLocales() throws SOAPException
```

<table>
<thead>
<tr>
<th>Iterator</th>
<th>Reason Text</th>
<th>Locale</th>
<th>Locale</th>
</tr>
</thead>
<tbody>
<tr>
<td>getFaultReasonText</td>
<td>Reason Text</td>
<td>Locale</td>
<td>Locale</td>
</tr>
<tr>
<td>return</td>
<td>Iterator Reason Text</td>
<td>Locale</td>
<td></td>
</tr>
</tbody>
</table>

**Throws**
- **SOAPException**: Reason
- **UnsupportedOperationException**: SOAP 1.2 Fault Reason

since SAAJ 1.3

**getFaultReasonLocales**

```java
Iterator getFaultReasonLocales() throws SOAPException
```

Returns an `Iterator` over a distinct sequence of `Locale` objects for which there are associated Reason Text items. Any of these `Locale` objects can be used in a call to `getFaultReasonText` in order to obtain a localized version of the Reason Text string.

**Returns:**
- an `Iterator` over a sequence of `Locale` objects for which there are associated Reason Text items.

**Throws:**
- **SOAPException** - if there was an error in retrieving the fault Reason locales.
- **UnsupportedOperationException** - if this message does not support the SOAP 1.2 concept of Fault Reason.

**Since:**
- SAAJ 1.3
public java.util.Iterator<E> getFaultReasonTexts() throws SOAPException

Iterator SOAPFault Reason Text String
return Iterator env:Fault/env:Reason/env:Text
Throws SOAPException: Reason
Throws UnsupportedOperationException: SOAP 1.2 Fault Reason
since SAAJ 1.3

getFaultReasonTexts

Iterator getFaultReasonTexts() throws SOAPException

Returns an Iterator over a sequence of String objects containing all of the Reason Text items for this SOAPFault.

Throws: SOAPException - if there was an error in retrieving the fault Reason texts.
UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Reason.
Since: SAAJ 1.3

public String getFaultReasonText(java.util.Locale locale) throws SOAPException

Locale Reason Text Reason Text Text
Reason Text
locale Reason Text Locale
return locale Reason Text
Throws SOAPException: Reason
UnsupportedOperationException: SOAP 1.2 Fault
getFaultReasonText

String getFaultReasonText(Locale locale)
throws SOAPException

Returns the Reason Text associated with the given Locale. If more than one such Reason Text exists the first matching Text is returned

Parameters:
locale -- the Locale for which a localized Reason Text is desired

Returns:
the Reason Text associated with locale

Throws:
SOAPException - if there was an error in retrieving the fault Reason text for the specified locale.
UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Reason.

Since:
SAAJ 1.3

See Also:
getFaultString()

public void addFaultReasonText(String text, java.util.Locale locale) throws SOAPException

Reason Text locale
Reason Text text locale null

SOAPFault fault = ...;
void addFaultReasonText(String text,
        Locale locale)
    throws SOAPException

Appends or replaces a Reason Text item containing the specified text message and an xml:lang derived from locale. If a Reason Text item with this xml:lang already exists its text value will be replaced with text. The locale parameter should not be null

Code sample:

SOAPFault fault = ...;
fault.addFaultReasonText("Version Mismatch", Locale.ENGLISH);

Parameters:

text - -- reason message string
locale - -- Locale object representing the locale of the message

Throws:

SOAPException - if there was an error in adding the Reason text or the locale passed was null.

UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Reason.

Since:

SAAJ 1.3
public String getFaultNode()

SOAPFault  Node Node  SOAP 1.2

    return String env:Fault/env:Node  null

Throws UnsupportedOperationException:  SOAP 1.2  Fault Node
since  SAAJ 1.3

getFaultNode

String getFaultNode()

    Returns the optional Node element value for this SOAPFault object. The Node element is optional in SOAP 1.2.

    Returns:
    Content of the env:Fault/env:Node element as a String or null if none

    Throws:
    UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Node.

Since:
    SAAJ 1.3

public void setFaultNode(String uri) throws SOAPException

SOAPFault  Node Node  SOAP 1.2

    Throws  SOAPException:  SOAPFault  Node

    Throws UnsupportedOperationException:  SOAP 1.2  Fault Node

since  SAAJ 1.3

setFaultNode

void setFaultNode(String uri)
    throws SOAPException
Creates or replaces any existing Node element value for this SOAPFault object. The Node element is optional in SOAP 1.2.

Throws:

SOAPException - if there was an error in setting the Node for this SOAPFault object.
UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Node.

Since:
SAAJ 1.3

public String getFaultRole()

SOAPFault  Role Role  SOAP 1.2

dereturn String env:Fault/env:Role
null

Throws UnsupportedOperationException: SOAP 1.2  Fault Role

since SAAJ 1.3

getFaultRole

String getFaultRole()

Returns the optional Role element value for this SOAPFault object. The Role element is optional in SOAP 1.2.

Returns:

Content of the env:Fault/env:Role element as a String or null if none

Throws:

UnsupportedOperationException - if this message does not support the SOAP 1.2 concept of Fault Role.

Since:
SAAJ 1.3

public void setFaultRole(String uri) throws SOAPException

SOAPFault  Role Role  SOAP 1.2
setFaultRole

void setFaultRole(String uri)

throws SOAPException

Creates or replaces any existing Role element value for this
SOAPFault object. The Role element is optional in SOAP 1.2.

Parameters:
uri - the URI of the Role

Throws:
SOAPException - if there was an error in setting the Role for this
SOAPFault object.
UnsupportedOperationException - if this message does not
support the SOAP 1.2 concept of Fault Role.

Since:
SAAJ 1.3
javax.xml.soap  Interface SOAPFaultElement

All Superinterfaces:  
   Element, Node, SOAPElement

All Known Subinterfaces:  
   Detail

public interface SOAPFaultElement  
extends SOAPElement

Implements: SOAPElement  
Implemented by: Detail

SOAPFault Detail SOAPFaultElement

   SOAPElement addTextNode SOAPFaultElement

A representation of the contents in a SOAPFault object. The Detail interface is a SOAPFaultElement.

Content is added to a SOAPFaultElement using the SOAPElement method addTextNode.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE,</td>
</tr>
<tr>
<td>ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE,</td>
</tr>
<tr>
<td>NOTATION_NODE,</td>
</tr>
</tbody>
</table>
## Field Summary


<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ATTRIBUTE_NODE</code></td>
</tr>
<tr>
<td><code>CDATA_SECTION_NODE</code></td>
</tr>
<tr>
<td><code>COMMENT_NODE</code></td>
</tr>
<tr>
<td><code>DOCUMENT_FRAGMENT_NODE</code></td>
</tr>
<tr>
<td><code>DOCUMENT_NODE</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_CONTAINED_BY</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_CONTAINS</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_DISCONNECTED</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_FOLLOWING</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC</code></td>
</tr>
<tr>
<td><code>DOCUMENT_POSITION_PRECEDING</code></td>
</tr>
<tr>
<td><code>DOCUMENT_TYPE_NODE</code></td>
</tr>
<tr>
<td><code>ELEMENT_NODE</code></td>
</tr>
<tr>
<td><code>ENTITY_NODE</code></td>
</tr>
<tr>
<td><code>ENTITY_REFERENCE_NODE</code></td>
</tr>
<tr>
<td><code>NOTATION_NODE</code></td>
</tr>
<tr>
<td><code>PROCESSING_INSTRUCTION_NODE</code></td>
</tr>
<tr>
<td><code>TEXT_NODE</code></td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from interface [javax.xml.soap.SOAPElement](https://developer.mozilla.org/en-US/docs/Web/API/SOAPElement)

- `addAttribute`, `addAttribute`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addNamespaceDeclaration`, `addTextNode`, `createQName`, `getAllAttributes`, `getAllAttributesAsQNames`, `getAttributeValue`, `getAttributeValue`, `getChildren`, `getElementQName`, `getEncodingStyle`, `getNamespacePrefixes`, `getNamespaceURI`, `getVisibleNamespacePrefixes`, `removeAttribute`, `removeContents`, `removeNamespaceDeclaration`, `setElementQName`, `setEncodingStyle`


- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

### Methods inherited from interface org.w3c.dom.[**Node**](https://developer.mozilla.org/en-US/docs/Web/API/Node)

- `appendChild`, `cloneNode`, `compareDocumentPosition`, `getAttributes`, `getBaseURI`, `getChildren`, `getFeature`, `getFirstChild`, `getLastChild`, `getLocalName`, `getNamespaceURI`, `getNextSibling`, `getNodeName`, `getNodeType`, `getNodeValue`, `getOwnerDocument`, `getParentNode`, `getPrefix`, `getPreviousSibling`, `getTextContent`, `getUserData`, `hasAttributes`, `hasChildNodes`, `insertBefore`, `isDefaultNamespace`, `isEqualNode`, `isSameNode`, `isSupported`, `lookupNamespaceURI`, `lookupPrefix`, `normalize`, `removeChild`, `replaceChild`, `setNodeValue`, `setPrefix`, `setTextContent`, `setUserData`
## Methods inherited from interface org.w3c.dom.Element

- `getAttribute`
- `getAttributeNode`
- `getAttributeNodeNS`
- `getAttributeNS`
- `getElementsByTagName`
- `getElementsByTagNameNS`
- `getSchemaTypeInfo`
- `getTagName`
- `hasAttribute`
- `hasAttributeNS`
- `removeAttribute`
- `removeAttributeNode`
- `removeAttributeNS`
- `setAttribute`
- `setAttributeNode`
- `setAttributeNodeNS`
- `setAttributeNS`
- `setIdAttribute`
- `setIdAttributeNode`
- `setIdAttributeNS`

## Methods inherited from interface org.w3c.dom.Node

- `appendChild`
- `cloneNode`
- `compareDocumentPosition`
- `getAttributes`
- `getBaseURI`
- `getChildNodes`
- `getFeature`
- `getFirstChild`
- `getLastChild`
- `getLocalName`
- `getNamespaceURI`
- `getNextSibling`
- `getNodeName`
- `getNodeType`
- `getNodeValue`
- `getOwnerDocument`
- `getParentNode`
- `getPrefix`
- `getPreviousSibling`
- `getTextContent`
- `getUserData`
- `hasAttributes`
- `hasChildNodes`
- `insertBefore`
- `isDefaultNamespace`
- `isEqualNode`
- `isSameNode`
- `isSupported`
- `lookupNamespaceURI`
- `lookupPrefix`
- `normalize`
- `removeChild`
- `replaceChild`
- `setNodeValue`
- `setPrefix`
- `setTextContent`
- `setUserData`
javax.xml.rpc.soap  Class SOAPFaultException

java.lang.Object  
  ├ java.lang.Throwable  
  │  └ java.lang.Exception  
  │   └ java.lang.RuntimeException  
  └ javax.xml.rpc.soap.SOAPFaultException

All Implemented Interfaces:  Serializable

public class SOAPFaultException
  extends RuntimeException

Extends:  Throwable > Exception > RuntimeException

SOAPFaultException  SOAP

SOAP  faultdetail  SOAPFaultException  getDetail
  javax.xml.soap.SOAPFactory  createDetail
javax.xml.soap.Detail

faultstring  SOAP  faultcode  SOAP

SOAP  SOAP 1.1  WSDL 1.1
  version  1.0

See also  javax.xml.soap.Detail, createDetail

The SOAPFaultException exception represents a SOAP fault.

The message part in the SOAP fault maps to the contents of faultdetail element accessible through the getDetail method on the SOAPFaultException. The method createDetail on the javax.xml.soap.SOAPFactory creates an instance of the javax.xml.soap.Detail.
The faultstring provides a human-readable description of the SOAP fault. The faultcode element provides an algorithmic mapping of the SOAP fault.

Refer to SOAP 1.1 and WSDL 1.1 specifications for more details of the SOAP faults.

Version:
1.0

Author:
Rahul Sharma

See Also:
Detail, SOAPFactory.createDetail(), Serialized Form

Constructor Summary

| SOAPFaultException( QName faultcode, String faultstring, String faultactor, Detail faultdetail) |
| Constructor for the SOAPFaultException |

Method Summary

| String getFaultActor() |
| Gets the faultactor element. |

| String getFaultString() |
| Gets the faultstring element. |

Methods inherited from class java.lang.Throwable
fillInStackTrace, get Cause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public SOAPFaultException(javax.xml.namespace.QName faultcode, String faultstring, String faultactor, Detail faultdetail)

SOAPFaultException

faultCode SOAP faultcode QName
faultstring SOAP faultstring
faultactor SOAP faultactor
faultdetail SOAP faultdetail

See also createDetail

SOAPFaultException

public SOAPFaultException(QName faultcode, String faultstring, String faultactor, Detail faultdetail)

Constructor for the SOAPFaultException

Parameters:

faultCode - QName for the SOAP faultcode
faultstring - faultstring element of SOAP fault
faultactor - faultactor element of SOAP fault
faultdetail - faultdetail element of SOAP fault

See Also:
SOAPFactory.createDetail()
public javax.xml.namespace.QName getFaultCode()

    faultcode  faultcode SOAP  SOAP  SOAP

    return  faultcode  QName

getAddress

public QName getFaultCode()

    Gets the faultcode element. The faultcode element provides an algorithmic mechanism for identifying the fault. SOAP defines a small set of SOAP fault codes covering basic SOAP faults.

    Returns:
    QName of the faultcode element

public String getFaultString()

    faultstring  faultstring  SOAP
    return  SOAP  faultstring

getAddress

public String getFaultString()

    Gets the faultstring element. The faultstring provides a human-readable description of the SOAP fault and is not intended for algorithmic processing.

    Returns:
    faultstring element of the SOAP fault

public String getFaultActor()

    faultactor  faultactor  SOAP  SOAP
**getFaultActor**

```java
public String getFaultActor()
```

Gets the `faultactor` element. The `faultactor` element provides information about which SOAP node on the SOAP message path caused the fault to happen. It indicates the source of the fault.

**Returns:**
`faultactor` element of the SOAP fault

---

**getDetail**

```java
public Detail getDetail()
```

detail

```java
detail
```

See also

```java
javax.xml.soap.Detail
```

---

**getDetail**

```java
public Detail getDetail()
```

detail
detail

See also

```java
Detail
```

---

Submit a bug or feature
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.ws.soap Class SOAPFaultException

javax.lang.Object
  ▼ java.lang.Throwable
    ▼ java.lang.Exception
      ▼ java.lang.RuntimeException
        ▼ javax.xml.ws.WebServiceException
          ▼ javax.xml.ws.ProtocolException
            ▼ javax.xml.ws.soap.SOAPFaultException

All Implemented Interfaces:
  Serializable

public class SOAPFaultException
extends ProtocolException

Extends: Throwable > Exception > RuntimeException

SOAPFaultException SOAP

SOAP faultdetail SOAPFaultException getDetail
  javax.xml.soap.SOAPFactory createDetail
javax.xml.soap.Detail

faultstring SOAP faultcode SOAP

SOAP SOAP 1.1 WSDL 1.1
  version 1.0
See also javax.xml.soap.Detail, createDetail

The SOAPFaultException exception represents a SOAP 1.1 or 1.2 fault.

A SOAPFaultException wraps a SAAJ SOAPFault that manages the SOAP-specific representation of faults. The createFault method of javax.xml.soap.SOAPFactory may be used to create an instance of javax.xml.soap.SOAPFault for use with the constructor. SOAPBinding
contains an accessor for the SOAPFactory used by the binding instance.

Note that the value of getFault is the only part of the exception used when searializing a SOAP fault.

Refer to the SOAP specification for a complete description of SOAP faults.

Since:
JAX-WS 2.0

See Also:
SOAPFault, SOAPBinding.getSOAPFactory(), ProtocolException, Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>SOAPFaultException(SOAPFault fault)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructor for SOAPFaultException</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>SOAPFault</th>
<th>getFault()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the embedded SOAPFault instance.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Throwable:
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object:
clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Constructor Detail
SOAPFaultException

public SOAPFaultException(SOAPFault fault)

Constructor for SOAPFaultException

Parameters:
  fault - SOAPFault representing the fault

See Also:
  SOAPFactory.createFault(java.lang.String, javax.xml.namespace.QName)

Method Detail

getFault

public SOAPFault getFault()

Gets the embedded SOAPFault instance.

Returns:
  javax.xml.soap.SOAPFault SOAP fault element
javax.xml.ws.handler.soap  Interface SOAPHandler<T extends SOAPMessageContext>

All Superinterfaces:
   Handler<T>

public interface SOAPHandler<T extends SOAPMessageContext>
extends Handler<T>

Implements: "">Handler

SOAPHandler   Handler
   since          JAX-WS 2.0

The SOAPHandler class extends Handler to provide typesafety for the message context parameter and add a method to obtain access to the headers that may be processed by the handler.

Since:
   JAX-WS 2.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set&lt;QName&gt;</td>
</tr>
<tr>
<td>getHeaders()</td>
</tr>
<tr>
<td>Gets the header blocks that can be processed by this Handler instance.</td>
</tr>
</tbody>
</table>

| Methods inherited from interface javax.xml.ws.handler.Handler |
| close, handleFault, handleMessage |

Method Detail
public java.util.Set<E> getHeaders()

    return QName

getHeaders

Set<QName> getHeaders()

Gets the header blocks that can be processed by this Handler instance.

Returns:
    Set of QNames of header blocks processed by this handler instance. QName is the qualified name of the outermost element of the Header block.
javax.xml.soap  Interface SOAPHeader

All Superinterfaces:
   Element, Node, SOAPElement

public interface SOAPHeader
extends SOAPElement

Implements: SOAPElement

SOAP SOAP XML

SOAPEnvelope SOAPHeader SOAPHeader SE
SOAPEnvelope

se.getHeader().detachNode();

SOAPHeader SOAPEnvelope addHeader

se.getHeader().detachNode();
SOAPHeader sh = se.addHeader();

SOAPHeader SOAPHeaderElement addHeaderElement
HeaderElement SOAPHeader addHeaderElement Name
HeaderElement

SOAPHeaderElement shElement = sh.addHeaderElement(name);

See also  javax.xml.soap.SOAPHeaderElement

A representation of the SOAP header element. A SOAP header element consists of XML data that affects the way the application-specific content is processed by the message provider. For example, transaction semantics, authentication information, and so on, can be specified as the content of a SOAPHeader object.
A SOAPEnvelope object contains an empty SOAPHeader object by default. If the SOAPHeader object, which is optional, is not needed, it can be retrieved and deleted with the following line of code. The variable se is a SOAPEnvelope object.

```java
se.getHeader().detachNode();
```

A SOAPHeader object is created with the SOAPEnvelope method `addHeader`. This method, which creates a new header and adds it to the envelope, may be called only after the existing header has been removed.

```java
se.getHeader().detachNode();
SOAPHeader sh = se.addHeader();
```

A SOAPHeader object can have only SOAPHeaderElement objects as its immediate children. The method `addHeaderElement` creates a new HeaderElement object and adds it to the SOAPHeader object. In the following line of code, the argument to the method `addHeaderElement` is a Name object that is the name for the new HeaderElement object.

```java
SOAPHeaderElement shElement = sh.addHeaderElement(name);
```

See Also: [SOAPHeaderElement](#)

### Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addHeaderElement(Name name)</code></td>
<td>Creates a new SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addHeaderElement(QName qname)</code></td>
<td>Creates a new SOAPHeaderElement object initialized with the specified qname and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addNotUnderstoodHeaderElement(QName name)</code></td>
<td>Creates a new NotUnderstood SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addUpgradeHeaderElement(Iterator supportedSOAPURIs)</code></td>
<td>Creates a new Upgrade SOAPHeaderElement object initialized with the specified List of supported SOAP URIs and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addUpgradeHeaderElement(String supportedSoapUri)</code></td>
<td>Creates a new Upgrade SOAPHeaderElement object initialized with the specified supported SOAP URI and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>SOAPHeaderElement</td>
<td><code>addUpgradeHeaderElement(String[] supportedSoapUris)</code></td>
<td>Creates a new Upgrade SOAPHeaderElement object initialized with the specified array of supported SOAP URIs and adds it to this SOAPHeader object.</td>
</tr>
<tr>
<td>Iterator</td>
<td><code>examineAllHeaderElements()</code></td>
<td>Returns an Iterator over all the</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>Iterator examineHeaderElements(String actor)</td>
<td>Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified <code>actor</code>.</td>
<td></td>
</tr>
<tr>
<td>Iterator examineMustUnderstandHeaderElements(String actor)</td>
<td>Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified <code>actor</code> and that have a MustUnderstand attribute whose value is equivalent to true.</td>
<td></td>
</tr>
<tr>
<td>Iterator extractAllHeaderElements()</td>
<td>Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object and detaches them from this SOAPHeader object.</td>
<td></td>
</tr>
<tr>
<td>Iterator extractHeaderElements(String actor)</td>
<td>Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified <code>actor</code> and detaches them from this SOAPHeader object.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.xml.soap.SOAPElement`

- `addAttribute`, `addAttribute`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `addChildElement`, `createQName`, `getAllAttributes`, `getAllAttributesAsQNames`, `getAttributeValue`, `getAttributeValue`, `getAttributes`, `getAttributes`, `getChildElements`, `getChildElements`, `getChildElements`, `getChildren`, `getElementName`, `getElementQName`, `getEncodingStyle`, `getNamespacePrefixes`, `getNamespaceURI`, `getVisibleNamespacePrefixes`, `removeAttribute`, `removeAttribute`, `removeContents`, `removeNamespaceDeclaration`, `setElementQName`, `setEncodingStyle`

Methods inherited from interface `javax.xml.soap.Node`

- `detachNode`, `getParentElement`, `getValue`, `recycleNode`, `setParentElement`, `setValue`

Methods inherited from interface `org.w3c.dom.Node`
Methods inherited from interface org.w3c.dom.Element

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>getAttribute, getAttributeNode, getAttributeNS, getAttributeNS, getElementsByTagName, getElementsByTagNameNS, getSchemaTypeInfo, getTagName, hasAttribute, hasAttributeNS, removeAttribute, removeAttributeNode, removeAttributeNS, setAttribute, setAttributeNode, setAttributeNodeNS, setAttributeNS, setIdAttribute, setIdAttributeNode, setIdAttributeNS</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface org.w3c.dom.Node

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData</td>
<td></td>
</tr>
</tbody>
</table>

Method Detail

public SOAPHeaderElement addHeaderElement(Name name) throws SOAPException

<table>
<thead>
<tr>
<th>name</th>
<th>SOAPHeaderElement</th>
<th>SOAPHeader</th>
</tr>
</thead>
</table>

Throws

SOAPException: SOAP
See also

addHeaderElement(javax.xml.namespace.QName)
addHeaderElement

SOAPHeaderElement addHeaderElement(Name name)
throws SOAPException

Creates a new SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object.

Parameters:
   name - a Name object with the name of the new SOAPHeaderElement object

Returns:
   the new SOAPHeaderElement object that was inserted into this SOAPHeader object

Throws:
   SOAPException - if a SOAP error occurs

See Also:
   addHeaderElement(javax.xml.namespace.QName)

public SOAPHeaderElement addHeaderElement(javax.xml.namespace.QName qname)
throws SOAPException

qname   SOAPHeaderElement    SOAPHeader

   qname    SOAPHeaderElement    qname    QName
return    SOAPHeader    SOAPHeaderElement

Throws:
   SOAPException: SOAP
since    SAAJ 1.3
See also
   addHeaderElement(Name)

addHeaderElement

SOAPHeaderElement addHeaderElement(QName qname)
throws SOAPException
Creates a new SOAPHeaderElement object initialized with the specified qname and adds it to this SOAPHeader object.

Parameters:
  qname - a QName object with the qname of the new SOAPHeaderElement object

Returns:
  the new SOAPHeaderElement object that was inserted into this SOAPHeader object

Throws:
  SOAPException - if a SOAP error occurs

Since:
  SAAJ 1.3

See Also:
  addHeaderElement(Name)

public java.util.Iterator<E>
examineMustUnderstandHeaderElements(String actor)

  SOAPHeader  SOAPHeaderElement  Iterator  actor
  true  MustUnderstand

SOAP 1.2      env:actor  env:role
  actor  String  actor/role  URI
return  Iterator  SOAPHeaderElement  actor / role
since  MustUnderstand  SAAJ 1.2
See  examineHeaderElements, extractHeaderElements,
also  URI_SOAP_ACTOR_NEXT

examineMustUnderstandHeaderElements

Iterator examineMustUnderstandHeaderElements(String actor)

  Returns an Iterator over all the SOAPHeaderElement objects in this
SOAPHeader object that have the specified actor and that have a
MustUnderstand attribute whose value is equivalent to true.

In SOAP 1.2 the `env:actor` attribute is replaced by the `env:role` attribute, but with essentially the same semantics.

**Parameters:**
- `actor` - a String giving the URI of the actor / role for which to search

**Returns:**
- an Iterator object over all the SOAPHeaderElement objects that contain the specified actor / role and are marked as MustUnderstand

**Since:**
- SAAJ 1.2

**See Also:**
- `examineHeaderElements(java.lang.String)`, `extractHeaderElements(java.lang.String)`, `SOAPConstants.URI_SOAP_ACTOR_NEXT`

```java
public java.util.Iterator<E> examineHeaderElements(String actor)
```

```
SOAPHeader actor SOAPHeaderElement Iterator
actor actor actor
actor message
```

**SOAP 1.2**

```
env:actor env:role
actor String actor/role URI
return Iterator actor / role SOAPHeaderElement
See also extractHeaderElements, URI_SOAP_ACTOR_NEXT
```

**examineHeaderElements**

```java
Iterator examineHeaderElements(String actor)
```

Returns an Iterator over all the SOAPHeaderElement objects in this
SOAPHeader object that have the specified `actor`. An `actor` is a global attribute that indicates the intermediate parties that should process a message before it reaches its ultimate receiver. An actor receives the message and processes it before sending it on to the next actor. The default actor is the ultimate intended recipient for the message, so if no `actor` attribute is included in a `SOAPHeader` object, it is sent to the ultimate receiver along with the message body.

In SOAP 1.2 the `env:actor` attribute is replaced by the `env:role` attribute, but with essentially the same semantics.

**Parameters:**
- `actor` - a `String` giving the URI of the `actor`/`role` for which to search

**Returns:**
- an `Iterator` object over all the `SOAPHeaderElement` objects that contain the specified `actor`/`role`

**See Also:**
- `extractHeaderElements(java.lang.String)`, `SOAPConstants.URI_SOAP_ACTOR_NEXT`

```java
public java.util.Iterator<E> extractHeaderElements(String actor)
```

**SOAP 1.2**
```
env:actor    env:role
actor        String    actor/role    URI
return       Iterator    actor/role    SOAPHeaderElement
See also     examineHeaderElements, URI_SOAP_ACTOR_NEXT
```

`extractHeaderElements`
Iterator extractHeaderElements(String actor)

Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified actor and detaches them from this SOAPHeader object.

This method allows an actor to process the parts of the SOAPHeader object that apply to it and to remove them before passing the message on to the next actor.

In SOAP 1.2 the env:actor attribute is replaced by the env:role attribute, but with essentially the same semantics.

Parameters:
actor - a String giving the URI of the actor / role for which to search

Returns:
an Iterator object over all the SOAPHeaderElement objects that contain the specified actor / role

See Also:
examineHeaderElements(java.lang.String), SOAPConstants.URI_SOAP_ACTOR_NEXT

public SOAPHeaderElement addNotUnderstoodHeaderElement(javax.xml.namespace.QName name) throws SOAPException
NotUnderstood
SOAP 1.2

name SOAPHeaderElement QName
return SOAPHeader SOAPHeaderElement

Throws SOAPException: SOAP

Throws UnsupportedOperationException: SOAP 1.1 Header

since SAAJ 1.3
**SOAPHeaderElement addNotUnderstoodHeaderElement(QName name)**

Throws **SOAPException**

Creates a new NotUnderstood SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object. This operation is supported only by SOAP 1.2.

**Parameters:**

name - a QName object with the name of the SOAPHeaderElement object that was not understood.

**Returns:**

the new SOAPHeaderElement object that was inserted into this SOAPHeader object

**Throws:**

**SOAPException** - if a SOAP error occurs.

**UnsupportedOperationException** - if this is a SOAP 1.1 Header.

**Since:**

SAAJ 1.3

---

**public SOAPHeaderElement addUpgradeHeaderElement(java.util.Iterator<E> supportedSOAPURIs) throws SOAPException**

SOAP URI Upgrade

**SOAPHeader SOAP 1.1 SOAP 1.2**

**supportedSOAPURIs** SOAP URI Iterator

**return** SOAPHeader SOAPHeaderElement

**Throws** **SOAPException**: SOAP

**since** SAAJ 1.3

**addUpgradeHeaderElement**

Creates a new Upgrade SOAPHeaderElement object initialized with the specified List of supported SOAP URIs and adds it to this SOAPHeader
object. This operation is supported on both SOAP 1.1 and SOAP 1.2 header.

**Parameters:**

- supportedSOAPURIs - an Iterator object with the URIs of SOAP versions supported.

**Returns:**

- the new SOAPHeaderElement object that was inserted into this SOAPHeader object

**Throws:**

- SOAPException - if a SOAP error occurs.

**Since:**

SAAJ 1.3

```java
public SOAPHeaderElement addUpgradeHeaderElement(String[] supportedSoapUris)
throws SOAPException
```

**SOAP URI** Upgrade

**SOAPHeader** SOAP 1.1 SOAP 1.2

**supportedSoapUris** SOAP URI

**return** SOAPHeader SOAPHeaderElement

**Throws** SOAPException SOAP

**since** SAAJ 1.3

---

**addUpgradeHeaderElement**

```
SOAPHeaderElement addUpgradeHeaderElement(String[] supportedSoapUris)
throws SOAPException
```

Creates a new Upgrade SOAPHeaderElement object initialized with the specified array of supported SOAP URIs and adds it to this SOAPHeader object. This operation is supported on both SOAP 1.1 and SOAP 1.2 header.

**Parameters:**

- supportedSoapUris - an array of the URIs of SOAP versions
public SOAPHeaderElement addUpgradeHeaderElement(String supportedSoapUri) throws SOAPException

SOAP URI Upgrade

SOAPHeader SOAP 1.1 SOAP 1.2

<table>
<thead>
<tr>
<th>supportedSoapUri</th>
<th>SOAP URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>SOAPHeader SOAPHeaderElement</td>
</tr>
<tr>
<td>Throws</td>
<td>SOAPException: SOAP</td>
</tr>
<tr>
<td>since</td>
<td>SAAJ 1.3</td>
</tr>
</tbody>
</table>

addUpgradeHeaderElement

SOAPHeaderElement addUpgradeHeaderElement(String supportedSoapUri) throws SOAPException

Creates a new Upgrade SOAPHeaderElement object initialized with the specified supported SOAP URI and adds it to this SOAPHeader object. This operation is supported on both SOAP 1.1 and SOAP 1.2 header.

Parameters:

- supportedSoapUri - the URI of SOAP the version that is supported.

Returns:

- the new SOAPHeaderElement object that was inserted into this SOAPHeader object

Throws:
**SOAPException** - if a SOAP error occurs.

**Since:**
SAAJ 1.3

---

```java
public java.util.Iterator<E> examineAllHeaderElements()
```

**SOAPHeader**
**SOAPHeaderElement**
**Iterator**

**return**
**Iterator**
**SOAPHeader**
**SOAPHeaderElement**

**since**
SAAJ 1.2

**See also**
extractAllHeaderElements

---

**examineAllHeaderElements**

**Iterator** examineAllHeaderElements()

Returns an **Iterator** over all the **SOAPHeaderElement** objects in this **SOAPHeader** object.

**Returns:**
an **Iterator** object over all the **SOAPHeaderElement** objects contained by this **SOAPHeader**

**Since:**
SAAJ 1.2

**See Also:**
extractAllHeaderElements()

---

```java
public java.util.Iterator<E> extractAllHeaderElements()
```

**SOAPHeader**
**SOAPHeaderElement**
**Iterator**

**return**
**Iterator**
**SOAPHeader**
**SOAPHeaderElement**

**since**
SAAJ 1.2

**See also**
examineAllHeaderElements

---

**extractAllHeaderElements**
Iterator `extractAllHeaderElements()`

Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object and detaches them from this SOAPHeader object.

**Returns:**

an Iterator object over all the SOAPHeaderElement objects contained by this SOAPHeader

**Since:**

SAAJ 1.2

**See Also:**

[examineAllHeaderElements()](#)

---

**Overview**  |  **Package**  |  **Tree**  |  **Deprecated**  |  **Index**  |  **Help**
--- | --- | --- | --- | --- | ---
PREV CLASS | NEXT CLASS | SUMMARY: NESTED | FIELD | CONSTR | METHOD | FRAMES | NO FRAMES | DETAIL | FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.xml.soap Interface SOAPHeaderElement

All Superinterfaces:
   Element, Node, SOAPElement

public interface SOAPHeaderElement
   extends SOAPElement

Implements: SOAPElement

SOAP  SOAP
   SOAPHeader  SOAPHeaderElement
   SOAPHeaderElement  SOAPElement

An object representing the contents in the SOAP header part of the SOAP envelope. The immediate children of a SOAPHeader object can be represented only as SOAPHeaderElement objects.

A SOAPHeaderElement object can have other SOAPElement objects as its children.

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE, DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE, DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS, DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING, DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC, DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE, PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>
**Fields inherited from interface org.w3c.dom.Node**

ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,
DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,
DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,
DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,
DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,
DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,
ENTITY_NODE, ENTITY_REFERENCE_NODE, NOTATION_NODE,
PROCESSING_INSTRUCTION_NODE, TEXT_NODE

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getActor()</strong> String</td>
<td>Returns the uri of the <strong>actor</strong> attribute of this SOAPHeaderElement.</td>
</tr>
<tr>
<td><strong>getMustUnderstand()</strong> boolean</td>
<td>Returns the boolean value of the <strong>mustUnderstand</strong> attribute for this SOAPHeaderElement.</td>
</tr>
<tr>
<td><strong>getRelay()</strong> boolean</td>
<td>Returns the boolean value of the <strong>relay</strong> attribute for this SOAPHeaderElement.</td>
</tr>
<tr>
<td><strong>getRole()</strong> String</td>
<td>Returns the value of the <strong>Role</strong> attribute of this SOAPHeaderElement.</td>
</tr>
<tr>
<td><strong>setActor(String actorURI)</strong> void</td>
<td>Sets the actor associated with this SOAPHeaderElement object to the specified actor.</td>
</tr>
<tr>
<td><strong>setMustUnderstand(boolean mustUnderstand)</strong> void</td>
<td>Sets the <strong>mustUnderstand</strong> attribute for this SOAPHeaderElement object to be either true or false.</td>
</tr>
<tr>
<td><strong>setRelay(boolean relay)</strong> void</td>
<td>Sets the <strong>relay</strong> attribute for this SOAPHeaderElement to be either true or false.</td>
</tr>
<tr>
<td><strong>setRole(String uri)</strong> void</td>
<td>Sets the <strong>Role</strong> associated with this SOAPHeaderElement object to the specified <strong>Role</strong>.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface javax.xml.soap.SOAPElement**
addAttribute, addAttribute, addChildElement, addChildElement, addChildElement, addChildElement, createQName, getAllAttributes, getAllAttributesAsQNames, getAttributeValue, getAttributeValue, getChildElements, getChildElements, getChildElements, getElementName, getElementQName, getEncodingStyle, getNamespacePrefixes, getNamespaceURI, getVisibleNamespacePrefixes, removeAttribute, removeContents, removeNamespaceDeclaration, setElementQName, setEncodingStyle

Methods inherited from interface javax.xml.soap.Node
detachNode, getParentElement, getValue, recycleNode, setParentElement, setValue

Methods inherited from interface org.w3c.dom.Node
appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild, replaceChild, setNodeValue, setPrefix, setTextContent, setUserData

Methods inherited from interface org.w3c.dom.Element
getAttribute, getAttributeNode, getAttributeNodeNS, getAttributeName, getElementsByTagName, getElementsByTagNameNS, getSchemaTypeInfo, getTagName, hasAttribute, hasAttributeNS, removeAttribute, removeAttributeNode, removeAttributeNS, setAttribute, setAttributeNode, setAttributeNodeNS, setAttributeNS, setIdAttribute, setIdAttributeNode, setIdAttributeNS

Methods inherited from interface org.w3c.dom.Node
appendChild, cloneNode, compareDocumentPosition, getAttributes, getBaseURI, getChildNodes, getFeature, getFirstChild, getLastChild, getLocalName, getNamespaceURI, getNextSibling, getNodeName, getNodeType, getNodeValue, getOwnerDocument, getParentNode, getPrefix, getPreviousSibling, getTextContent, getUserData, hasAttributes, hasChildNodes, insertBefore, isDefaultNamespace, isEqualNode, isSameNode, isSupported, lookupNamespaceURI, lookupPrefix, normalize, removeChild,
public void setActor(String actorURI)

SOAPHeaderElement actor actor
SOAPConstants.URI_SOAP_ACTOR_NEXT

SOAPHeaderElement actorURI actor URI
Throws IllegalArgumentException: actor
See also getActor

setActor

void setActor(String actorURI)

Sets the actor associated with this SOAPHeaderElement object to the specified actor. The default value of an actor is:
SOAPConstants.URI_SOAP_ACTOR_NEXT

If this SOAPHeaderElement supports SOAP 1.2 then this call is equivalent to setRole(String)

Parameters:
actorURI - a String giving the URI of the actor to set

Throws:
IllegalArgumentException - if there is a problem in setting the actor.

See Also:
getActor()

public void setRole(String uri) throws SOAPException
setRole

```java
void setRole(String uri)
    throws SOAPException
```

Sets the Role associated with this SOAPHeaderElement object to the specified Role.

**Parameters:**
- `uri` -- the URI of the Role

**Throws:**
- `SOAPException` - if there is an error in setting the role
- `UnsupportedOperationException` - if this message does not support the SOAP 1.2 concept of Fault Role.

**Since:**
- SAAJ 1.3

---

getActor

```java
public String getActor()
```

**See also**
- `setActor`
Returns the uri of the *actor* attribute of this SOAPHeaderElement.

If this SOAPHeaderElement supports SOAP 1.2 then this call is equivalent to *getRole(*)

**Returns:**
a String giving the URI of the actor

**See Also:**
*setActor(java.lang.String)*

### public String getRole()

**SOAPHeaderElement** *Role*

**return**

Role URI  String

**Throws** UnsupportedOperation Exception: SOAP 1.2 Fault Role

**since**  SAAJ 1.3

getRole

**String** getRole()

Returns the value of the *Role* attribute of this SOAPHeaderElement.

**Returns:**
a String giving the URI of the Role

**Throws:**
  *UnsupportedOperationException* - if this message does not support the SOAP 1.2 concept of Fault Role.

**Since:**
  SAAJ 1.3

### public void setMustUnderstand(boolean mustUnderstand)

**SOAPHeaderElement** *mustUnderstand*  true  false

**mustUnderstand**

**SOAPHeaderElement**
public boolean getMustUnderstand()

SOAPHeaderElement mustUnderstand boolean

return SOAPHeaderElement mustUnderstand true false

getMustUnderstand

boolean getMustUnderstand()
Returns the boolean value of the mustUnderstand attribute for this SOAPHeaderElement.

Returns:
true if the mustUnderstand attribute of this SOAPHeaderElement object is turned on; false otherwise

```java
public void setRelay(boolean relay) throws SOAPException

SOAPHeaderElement  relay  true  false
SOAP relay  true  SOAP mustUnderstand  false

relay  relay
Throws  SOAPException: relay
Throws  UnsupportedOperationException: SOAP 1.2  Relay
since  SAAJ 1.3
See also  setMustUnderstand, getRelay
```

**setRelay**

```java
void setRelay(boolean relay)
throws SOAPException
```

Sets the relay attribute for this SOAPHeaderElement to be either true or false.

The SOAP relay attribute is set to true to indicate that the SOAP header block must be relayed by any node that is targeted by the header block but not actually process it. This attribute is ignored on header blocks whose mustUnderstand attribute is set to true or that are targeted at the ultimate reciever (which is the default). The default value of this attribute is false.

**Parameters:**
relay - the new value of the relay attribute
Throws:

- SOAPException - if there is a problem in setting the relay attribute.
- UnsupportedOperationExceptio

Since:
SAAJ 1.3

See Also:

setMustUnderstand(boolean), getRelay()
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.soap  Class SOAPMessage

  java.lang.Object
   javax.xml.soap.SOAPMessage

  public abstract class SOAPMessage
  extends Object

  SOAP  SOAP "" XML   XML/SOAP  MIME

  SOAPMessage  SOAP  SOAPMessage  SOAP  SOAPPart
  SOAP  XML

  SOAPMessage

  •  SOAPPart
  •  SOAPEnvelope
  •  SOAPBody
  •  SOAPHeader

  SOAPMessage.getSOAPPart()  SOAP  SOAPEnvelope  SOA
  SOAPEnvelope  SOAPBody  SOAPHeader

  SOAPPart sp = message.getSOAPPart();
  SOAPEnvelope se = sp.getEnvelope();
  SOAPBody sb = se.getBody();
  SOAPHeader sh = se.getHeader();

  SOAPMessage  SOAP  SOAPMessage  AttachmentPart  AttachmentPart
  SOAPMessage  AttachmentPart  SOAPMessage

  SOAP  XML  XML

  MessageFactory  SAAJ  SOAPMessage  MessageFactory
  ebXML  SOAPMessage  MessageFactory  ebXML
The root class for all SOAP messages. As transmitted on the "wire", a SOAP message is an XML document or a MIME message whose first body part is an XML/SOAP document.

A SOAPMessage object consists of a SOAP part and optionally one or more attachment parts. The SOAP part for a SOAPMessage object is a SOAPPart object, which contains information used for message routing and identification, and which can contain application-specific content. All data in the SOAP Part of a message must be in XML format.

A new SOAPMessage object contains the following by default:

- A SOAPPart object
- A SOAPEnvelope object
- A SOAPBody object
- A SOAPHeader object

The SOAP part of a message can be retrieved by calling the method SOAPMessage.getSOAPPart(). The SOAPEnvelope object is retrieved from the SOAPPart object, and the SOAPEnvelope object is used to retrieve the SOAPBody and SOAPHeader objects.

```java
SOAPPart sp = message.getSOAPPart();
SOAPEnvelope se = sp.getEnvelope();
SOAPBody sb = se.getBody();
SOAPHeader sh = se.getHeader();
```

In addition to the mandatory SOAPPart object, a SOAPMessage object may contain zero or more AttachmentPart objects, each of which contains application-specific data. The SOAPMessage interface provides methods for
creating `AttachmentPart` objects and also for adding them to a `SOAPMessage` object. A party that has received a `SOAPMessage` object can examine its contents by retrieving individual attachment parts.

Unlike the rest of a SOAP message, an attachment is not required to be in XML format and can therefore be anything from simple text to an image file. Consequently, any message content that is not in XML format must be in an `AttachmentPart` object.

A `MessageFactory` object may create `SOAPMessage` objects with behavior that is specialized to a particular implementation or application of SAAJ. For instance, a `MessageFactory` object may produce `SOAPMessage` objects that conform to a particular Profile such as ebXML. In this case a `MessageFactory` object might produce `SOAPMessage` objects that are initialized with ebXML headers.

In order to ensure backward source compatibility, methods that are added to this class after version 1.1 of the SAAJ specification are all concrete instead of abstract and they all have default implementations. Unless otherwise noted in the JavaDocs for those methods the default implementations simply throw an `UnsupportedOperationException` and the SAAJ implementation code must override them with methods that provide the specified behavior. Legacy client code does not have this restriction, however, so long as there is no claim made that it conforms to some later version of the specification than it was originally written for. A legacy class that extends the `SOAPMessage` class can be compiled and/or run against succeeding versions of the SAAJ API without modification. If such a class was correctly implemented then it will continue to behave correctly relative to the version of the specification against which it was written.

**See Also:**

`MessageFactory`, `AttachmentPart`

---

### Field Summary

| static String | `CHARACTER_SET_ENCODING` | Specifies the character type encoding for the SOAP |
| static String WRITE_XML_DECLARATION | Specifies whether the SOAP Message will contain an XML declaration when it is sent. |

**Constructor Summary**

**SOAPMessage()**

**Method Summary**

**abstract void addAttachmentPart(AttachmentPart AttachmentPart)**

Adds the given AttachmentPart object to this SOAPMessage object.

**abstract int countAttachments()**

Gets a count of the number of attachments in this message.

**abstract AttachmentPart createAttachmentPart()**

Creates a new empty AttachmentPart object.

**AttachmentPart createAttachmentPart(DataHandler dataHandler)**

Creates an AttachmentPart object and populates it using the given DataHandler object.

**AttachmentPart createAttachmentPart(Object content, String contentType)**

Creates an AttachmentPart object and populates it with the specified data of the specified content type.

**abstract AttachmentPart getAttachment(SOAPElement element)**

Returns an AttachmentPart object that is associated with an attachment that is referenced by this SOAPElement or null if no such attachment exists.

**abstract Iterator getAttachments()**

Retrieves all the AttachmentPart objects that are part of this SOAPMessage object.

**abstract Iterator getAttachments(MimeHeaders headers)**

Retrieves all the AttachmentPart objects that have
header entries that match the specified headers.

abstract String getContentDescription()
       Retrieves a description of this SOAPMessage object's content.

abstract MimeHeaders getMimeHeaders()
       Returns all the transport-specific MIME headers for this SOAPMessage object in a transport-independent fashion.

Object getProperty(String property)
       Retrieves value of the specified property.

SOAPBody getSOAPBody()
       Gets the SOAP Body contained in this SOAPMessage object.

SOAPHeader getSOAPHeader()
       Gets the SOAP Header contained in this SOAPMessage object.

abstract SOAPPart getSOAPPart()
       Gets the SOAP part of this SOAPMessage object.

abstract void removeAllAttachments()
       Removes all AttachmentPart objects that have been added to this SOAPMessage object.

abstract void removeAttachments(MimeHeaders headers)
       Removes all the AttachmentPart objects that have header entries that match the specified headers.

abstract void saveChanges()
       Updates this SOAPMessage object with all the changes that have been made to it.

abstract boolean saveRequired()
       Indicates whether this SOAPMessage object needs to have the method saveChanges called on it.

abstract void setContentDescription(String description)
       Sets the description of this SOAPMessage object's content with the given description.

void setProperty(String property, Object value)
       Associates the specified value with the specified property.
abstract void writeTo(OutputStream out)

Writes this SOAPMessage object to the given output stream.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

CHARACTER_SET_ENCODING

public static final String CHARACTER_SET_ENCODING

Specifies the character type encoding for the SOAP Message. Valid values include "utf-8" and "utf-16". See vendor documentation for additional supported values. The default is "utf-8".

Since:
SAAJ 1.2

See Also:
SOAPMessage.setProperty, Constant Field Values

WRITE_XML_DECLARATION

public static final String WRITE_XML_DECLARATION

Specifies whether the SOAP Message will contain an XML declaration when it is sent. The only valid values are "true" and "false". The default is "false".

Since:
public SOAPMessage()

SOAPMessage

public SOAPMessage()

Method Detail

setContentDescription

public abstract void setContentDescription(String description)

Sets the description of this SOAPMessage object's content with the given description.

Parameters:

description - a String describing the content of this message

See Also:

g getContentDescription()

getContentDescription

public abstract String getContentDescription()

Retrieves a description of this SOAPMessage object's content.
Returns:
a String describing the content of this message or null if no description has been set

See Also:
setContentDescription(java.lang.String)

getSOAPPart

public abstract SOAPPart getSOAPPart()

Gets the SOAP part of this SOAPMessage object.

SOAPMessage object contains one or more attachments, the SOAP Part must be the first MIME body part in the message.

Returns:
the SOAPPart object for this SOAPMessage object

getSOAPBody

public SOAPBody getSOAPBody()

throws SOAPException

Gets the SOAP Body contained in this SOAPMessage object.

Returns:
the SOAPBody object contained by this SOAPMessage object

Throws:
SOAPException - if the SOAP Body does not exist or cannot be retrieved

Since:
SAAJ 1.2
getSOAPHeader

public SOAPHeader getSOAPHeader() throws SOAPException

Gets the SOAP Header contained in this SOAPMessage object.

**Returns:**
the SOAPHeader object contained by this SOAPMessage object

**Throws:**
SOAPException - if the SOAP Header does not exist or cannot be retrieved

**Since:**
SAAJ 1.2

---

abstract public void removeAllAttachments()

SOAPMessage AttachmentPart

SOAP

removeAllAttachments

public abstract void removeAllAttachments()

Removes all AttachmentPart objects that have been added to this SOAPMessage object.

This method does not touch the SOAP part.

---

abstract public int countAttachments()

SOAP

return SOAPMessage AttachmentPart
**countAttachments**

```java
public abstract int countAttachments()
```

Gets a count of the number of attachments in this message. This count does not include the SOAP part.

**Returns:**
- the number of AttachmentPart objects that are part of this SOAPMessage object

---

**getAttachments**

```java
public abstract Iterator getAttachments()
```

Retrieves all the AttachmentPart objects that are part of this SOAPMessage object.

**Returns:**
- an iterator over all the attachments in this message

---

**getAttachments**

```java
public abstract Iterator getAttachments(MimeHeaders headers)
```

Retrieves all the AttachmentPart objects that have header entries that match the specified headers. Note that a returned attachment could have headers in addition to those specified.

**Parameters:**
- headers - a MimeHeaders object containing the MIME headers for which to search
Returns:
an iterator over all attachments that have a header that matches
one of the given headers

abstract public void removeAttachments(MimeHeaders headers)

Parameters:
headers - a MimeHeaders object containing the MIME headers for
which to search

Since:
SAAJ 1.3

abstract public AttachmentPart getAttachment(SOAPElemen...
getAttachment

public abstract AttachmentPart getAttachment(SOAPElement element) throws SOAPException

Returns an AttachmentPart object that is associated with an attachment that is referenced by this SOAPElement or null if no such attachment exists. References can be made via an href attribute as described in SOAP Messages with Attachments, or via a single Text child node containing a URI as described in the WS-I Attachments Profile 1.0 for elements of schema type ref:swaRef(ref:swaRef). These two mechanisms must be supported. The support for references via href attribute also implies that this method should also be supported on an element that is an xop:Include element (XOP). other reference mechanisms may be supported by individual implementations of this standard. Contact your vendor for details.

Parameters:

    element - The SOAPElement containing the reference to an Attachment

Returns:

    the referenced AttachmentPart or null if no such AttachmentPart exists or no reference can be found in this SOAPElement.
abstract public void addAttachmentPart(AttachmentPart AttachmentPart)

    attachment - if there is an error in the attempt to access the attachment

Since: SAAJ 1.3

abstract public AttachmentPart createAttachmentPart()

    return SOAPMessage AttachmentPart

createAttachmentPart

addAttachmentPart

public abstract void addAttachmentPart(AttachmentPart AttachmentPart)

Adds the given AttachmentPart object to this SOAPMessage object. An AttachmentPart object must be created before it can be added to a message.

Parameters:
    AttachmentPart - an AttachmentPart object that is to become part of this SOAPMessage object

Throws:
    IllegalArgumentException

public abstract AttachmentPart createAttachmentPart()

Creates a new empty AttachmentPart object. Note that the method addAttachmentPart must be called with this new AttachmentPart object as the parameter in order for it to become an attachment to this SOAPMessage object.

Returns:

a new AttachmentPart object that can be populated and added to this SOAPMessage object

public AttachmentPart createAttachmentPart(DataHandler dataHandler)

createAttachmentPart

public AttachmentPart createAttachmentPart(DataHandler dataHandler)

Creates an AttachmentPart object and populates it using the given DataHandler object.

Parameters:

dataHandler - the javax.activation.DataHandler object that will generate the content for this SOAPMessage object

Returns:

a new AttachmentPart object that contains data generated by the given DataHandler object

Throws:

IllegalArgumentException - if there was a problem with the specified DataHandler object

See also

javax.activation.DataHandler,
javax.activation.DataContentHandler
See Also:
DataHandler, DataContentHandler

getMimeHeaders

```java
public abstract MimeHeaders getMimeHeaders()
```

Returns all the transport-specific MIME headers for this SOAPMessage object in a transport-independent fashion.

Returns:

a MimeHeaders object containing the MimeHeader objects

```java
public AttachmentPart createAttachmentPart(Object content, String contentType)
```

Creates an AttachmentPart object and populates it with the specified data of the specified content type. The type of the Object should correspond to the value given for the Content-Type.

Parameters:

Throws

IllegalArgumentException: contentType

See also

javax.activation.DataHandler, javax.activation.DataContentHandler

createAttachmentPart

```java
public AttachmentPart createAttachmentPart(Object content, String contentType)
```

Creates an AttachmentPart object and populates it with the specified data of the specified content type. The type of the Object should correspond to the value given for the Content-Type.

Parameters:
content - an Object containing the content for the AttachmentPart object to be created
contentType - a String object giving the type of content; examples are "text/xml", "text/plain", and "image/jpeg"

Returns:
a new AttachmentPart object that contains the given data

Throws:
IllegalArgumentException - may be thrown if the contentType does not match the type of the content object, or if there was no DataContentHandler object for the given content object

See Also:
DataHandler, DataContentHandler

abstract public void saveChanges() throws SOAPException

saveChanges
saveChanges id
Messageld
saveChanges MIME
AttachmentPart

Throws
SOAPException:

saveChanges

public abstract void saveChanges() throws SOAPException

Updates this SOAPMessage object with all the changes that have been made to it. This method is called automatically when writeTo(OutputStream) is called. However, if changes are made to a message that was received or to one that has already been sent, the method saveChanges needs to be called explicitly in order to save the changes. The method saveChanges also generates any changes that can be read back (for example, a Messageld in profiles that support
a message id). All MIME headers in a message that is created for sending purposes are guaranteed to have valid values only after saveChanges has been called.

In addition, this method marks the point at which the data from all constituent AttachmentPart objects are pulled into the message.

**Throws:**

- SOAPException - if there was a problem saving changes to this message.

```java
abstract public boolean saveRequired()

public abstract boolean saveRequired()

Indicates whether this SOAPMessage object needs to have the method saveChanges called on it.

Returns:

- true if saveChanges needs to be called; false otherwise.
```

```java
abstract public void writeTo(java.io.OutputStream out)

throws SOAPException, java.io.IOException

XML MIME

MIME Header

Throws java.io.IOException: I/O *
writeTo

public abstract void writeTo(OutputStream out)
   throws SOAPException, IOException

Writes this SOAPMessage object to the given output stream. The externalization format is as defined by the SOAP 1.1 with Attachments specification.

If there are no attachments, just an XML stream is written out. For those messages that have attachments, writeTo writes a MIME-encoded byte stream.

Note that this method does not write the transport-specific MIME Headers of the Message

Parameters:
   out - the OutputStream object to which this SOAPMessage object will be written

Throws:
   IOException - if an I/O error occurs
   SOAPException - if there was a problem in externalizing this SOAP message

setProperty

public void setProperty(String property, Object value)
   throws SOAPException

Associates the specified value with the specified property. If there was already a value associated with this property, the old value is
The valid property names include `WRITE_XML_DECLARATION` and `CHARACTER_SET_ENCODING`. All of these standard SAAJ properties are prefixed by "javax.xml.soap". Vendors may also add implementation specific properties. These properties must be prefixed with package names that are unique to the vendor.

Setting the property `WRITE_XML_DECLARATION` to "true" will cause an XML Declaration to be written out at the start of the SOAP message. The default value of "false" suppresses this declaration.

The property `CHARACTER_SET_ENCODING` defaults to the value "utf-8" which causes the SOAP message to be encoded using UTF-8. Setting `CHARACTER_SET_ENCODING` to "utf-16" causes the SOAP message to be encoded using UTF-16.

Some implementations may allow encodings in addition to UTF-8 and UTF-16. Refer to your vendor's documentation for details.

**Parameters:**
- `property` - the property with which the specified value is to be associated.
- `value` - the value to be associated with the specified property

**Throws:**
- `SOAPException` - if the property name is not recognized.

**Since:**
- SAAJ 1.2

---

getProperty

```java
public Object getProperty(String property)
    throws SOAPException
```
property - the name of the property to retrieve

**Returns:**
the value associated with the named property or `null` if no such property exists.

**Throws:**
`SOAPException` - if the property name is not recognized.

**Since:**
SAAJ 1.2
javax.xml.rpc.handler.soap  

**Interface SOAPMessageContext**

**All Superinterfaces:**  
MessageContext

```
public interface SOAPMessageContext
extends MessageContext
```

**Implements:** MessageContext

```java
javax.xml.rpc.soap.SOAPMessageContext  
javax.xml.soap.SOAPMessage
```

**RPC** SOAP  
SOAP 1.1 Java API

version 1.0

See also javax.xml.soap.SOAPMessage

The interface `javax.xml.rpc.soap.SOAPMessageContext` provides access to the SOAP message for either RPC request or response. The `javax.xml.soap.SOAPMessage` specifies the standard Java API for the representation of a SOAP 1.1 message with attachments.

**Version:**  
1.0

**Author:**  
Rahul Sharma

See Also:  
SOAPMessage

---

**Method Summary**

<table>
<thead>
<tr>
<th>SOAPMessage</th>
<th>getMessage()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the SOAPMessage from this message context</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String[]</th>
<th>getRoles()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the SOAP actor roles associated with an execution of the HandlerChain and its contained Handler</td>
</tr>
</tbody>
</table>
instances.

```java
void setMessage(SOAPMessage message)
    Sets the SOAPMessage in this message context
```

Methods inherited from interface `javax.xml.rpc.handler.MessageContext`
- `containsProperty`, `getProperty`, `getPropertyNames`, `removeProperty`, `setProperty`

## Method Detail

### public SOAPMessage getMessage()

```java
SOAPMessage getMessage()
```

Returns the SOAPMessage from this message context.

**Returns:**
- Returns the SOAPMessage; returns null if no SOAPMessage is present in this message context.

### public void setMessage(SOAPMessage message)

```java
setMessage(SOAPMessage message)
```

**Throws:**
- `JAXRPCException`: SOAPMessage
- `UnsupportedOperationException`
void setMessage(SOAPMessage message)

Sets the SOAPMessage in this message context

**Parameters:**

message - SOAP message

**Throws:**

JAXRPCException - If any error during the setting of the SOAPMessage in this message context

UnsupportedOperationException - If this operation is not supported

public String[] getRoles()

getRoles

String[] getRoles()

Gets the SOAP actor roles associated with an execution of the HandlerChain and its contained Handler instances. Note that SOAP actor roles apply to the SOAP node and are managed using HandlerChain.setRoles and HandlerChain.getRoles. Handler instances in the HandlerChain use this information about the SOAP actor roles to process the SOAP header blocks. Note that the SOAP actor roles are invariant during the processing of SOAP message through the HandlerChain.

**Returns:**

Array of URIs for SOAP actor roles

See Also:
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
Interface SOAPMessageContext

All Superinterfaces: Map< String, Object >, MessageContext

public interface SOAPMessageContext
extends MessageContext

Implements: MessageContext

javax.xml.rpc.soap.SOAPMessageContext RPC SOAP
javax.xml.soap.SOAPMessage SOAP 1.1 Java API

version 1.0
See also javax.xml.soap.SOAPMessage

The interface SOAPMessageContext provides access to the SOAP message for either RPC request or response. The javax.xml.soap.SOAPMessage specifies the standard Java API for the representation of a SOAP 1.1 message with attachments.

Since: JAX-WS 2.0
See Also: SOAPMessage

Nested Class Summary

| Nested classes/interfaces inherited from interface | javax.xml.ws.handler.MessageContext |
| MessageContext.Scope |

| Nested classes/interfaces inherited from interface java.util.Map |
Field Summary

Fields inherited from interface javax.xml.ws.handler.MessageContext
HTTP_REQUEST_HEADERS, HTTP_REQUEST_METHOD, HTTP_RESPONSE_CODE, HTTP_RESPONSE_HEADERS, INBOUND_MESSAGE_ATTACHMENTS, MESSAGE_OUTBOUND_PROPERTY, OUTBOUND_MESSAGE_ATTACHMENTS, PATH_INFO, QUERY_STRING, SERVLET_CONTEXT, SERVLET_REQUEST, SERVLET_RESPONSE, WSDL_DESCRIPTION, WSDL_INTERFACE, WSDL_OPERATION, WSDL_PORT, WSDL_SERVICE

Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object[]</td>
<td>getHeaders(QName header, JAXBContext context, boolean allRoles)</td>
<td>Gets headers that have a particular qualified name from the message in the message context.</td>
</tr>
<tr>
<td>SOAPMessage</td>
<td>getMessage()</td>
<td>Gets the SOAPMessage from this message context.</td>
</tr>
<tr>
<td>Set&lt;String&gt;</td>
<td>getRoles()</td>
<td>Gets the SOAP actor roles associated with an execution of the handler chain.</td>
</tr>
<tr>
<td>void</td>
<td>setMessage(SOAPMessage message)</td>
<td>Sets the SOAPMessage in this message context.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.ws.handler.MessageContext
getScope, setScope

Methods inherited from interface java.util.Map
clear, containsKey, containsValue, entrySet, equals, get, hashCode, isEmpty, keySet, put, putAll, remove, size, values
### Method Detail

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public <code>SOAPMessage</code> <code>getMessage()</code></td>
<td><code>SOAPMessage</code> return <code>SOAPMessage</code> <code>null</code></td>
</tr>
<tr>
<td><code>getMessage</code></td>
<td><code>getSOAPMessage()</code></td>
</tr>
<tr>
<td><code>SOAPMessage</code></td>
<td>Gets the SOAPMessage from this message context. Modifications to the returned SOAPMessage change the message in-place, there is no need to subsequently call <code>setMessage</code>.</td>
</tr>
<tr>
<td><strong>Returns:</strong></td>
<td><strong>Returns</strong> the SOAPMessage; returns null if no SOAPMessage is present in this message context</td>
</tr>
<tr>
<td>public <code>void</code> <code>setMessage(SOAPMessage</code> <code>message)</code></td>
<td><code>SOAPMessage</code> message <code>SOAP</code> <code>JAXRPCException</code></td>
</tr>
<tr>
<td><code>SOAPMessage</code></td>
<td>Sets the SOAPMessage in this message context</td>
</tr>
<tr>
<td><strong>Parameters:</strong></td>
<td><strong>Parameters:</strong></td>
</tr>
<tr>
<td><code>message</code> - SOAP message</td>
<td><code>message</code> - SOAP message</td>
</tr>
<tr>
<td><strong>Throws:</strong></td>
<td><strong>Throws:</strong></td>
</tr>
<tr>
<td><code>WebServiceException</code> - If any error during the setting of the</td>
<td><code>WebServiceException</code> - If any error during the setting of the</td>
</tr>
</tbody>
</table>
SOAPMessage in this message context

UnsupportedOperationException - If this operation is not supported

getHeaders

Object[] getHeaders(QName header,
                     JAXBContext context,
                     boolean allRoles)

Gets headers that have a particular qualified name from the message in the message context. Note that a SOAP message can contain multiple headers with the same qualified name.

Parameters:
header - The XML qualified name of the SOAP header(s).
context - The JAXBContext that should be used to unmarshall the header
allRoles - If true then returns headers for all SOAP roles, if false then only returns headers targetted at the roles currently being played by this SOAP node, see getRoles.

Returns:
An array of unmarshalled headers; returns an empty array if no message is present in this message context or no headers match the supplied qualified name.

Throws:
WebServiceException - If an error occurs when using the supplied JAXBContext to unmarshall. The cause of the WebServiceException is the original JAXBException.

public String[] getRoles()
getRoles

Set<String> getRoles()

Gets the SOAP actor roles associated with an execution of the handler chain. Note that SOAP actor roles apply to the SOAP node and are managed using SOAPBinding.setRoles and SOAPBinding.getRoles. Handler instances in the handler chain use this information about the SOAP actor roles to process the SOAP header blocks. Note that the SOAP actor roles are invariant during the processing of SOAP message through the handler chain.

Returns:
Array of String for SOAP actor roles
javax.jws.soap  Annotation Type SOAPMessageHandler

**Deprecated.** As of JSR-181 2.0 with no replacement.

**Implements:** Annotation

**Deprecated.**

SOAP

deprecated  

@Deprecated
public @interface SOAPMessageHandler

A single SOAP message handler

**Author:**
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>className</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Name of the handler class.</td>
</tr>
</tbody>
</table>

### Optional Element Summary

<table>
<thead>
<tr>
<th>String[]</th>
<th>headers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Deprecated.</strong> List of SOAP headers processed by the handler.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>InitParam[]</th>
<th>initParams</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Array of name/value pairs that should be passed to the handler during initialization.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>Deprecated.</strong> Name of the handler.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>String[]</td>
<td><strong>Deprecated.</strong> List of SOAP roles/actors implemented by the handler</td>
</tr>
</tbody>
</table>

## Element Detail

abstract public String `className()`

className

deprecated public abstract String `className`

**Deprecation.**
Name of the handler class.

abstract public String `name()`

name

deprecated public abstract String `name`

**Deprecation.**
Name of the handler. Defaults to the name of the handler class.

**Default:**
```
"
``` 实现了 SOAP 角色/角色的列表。

abstract public `InitParam`[] `initParams()`

```
**initParams**

```java
public abstract InitParam[] initParams
```

*Deprecated.*
Array of name/value pairs that should be passed to the handler during initialization.

**Default:**

```java
{}
```

---

**abstract public String[] roles()**

*SOAP /*

**roles**

```java
public abstract String[] roles
```

*Deprecated.*
List of SOAP roles/actors implemented by the handler

**Default:**

```java
{}
```

---

**abstract public String[] headers()**

*SOAP QName QName QName QName javax.xml.namespace.QName javax.xml.namespace.QName.*

```java
javax.xml.namespace.QName.valueOf(String qNameAsString)
```

**headers**

```java
public abstract String[] headers
```

*Deprecated.*
List of SOAP headers processed by the handler. Each element in this array contains a QName which defines the header element processed by the handler. The QNames are specified using the string notation described in the documentation for javax.xml.namespace.QName.valueOf(String qNameAsString)

**Default:**

```java
{}
```

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: REQUIRED</th>
<th>OPTIONAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>DETAIL: ELEMENT</td>
<td></td>
</tr>
</tbody>
</table>
javax.jws.soap Annotation Type
SOAPMessageHandlers

Deprecated. As of JSR-181 2.0 with no replacement.

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=TYPE)

Deprecated.
SOAP Web Service SOAP

@SOAPMessageHandlers SOAPMessageHandler
deprecated JSR-181 2.0

@Retention(value=RUNTIME)
@Target(value=TYPE)
@Deprecated
public @interface SOAPMessageHandlers

Specifies a list of SOAP protocol handlers that run before and after business methods on the Web Service. These handlers are called in response to SOAP messages targeting the service.

The @SOAPMessageHandlers annotation is an array of SOAPMessageHandler types. The handlers are run in the order in which they appear in the annotation, starting with the first handler in the array.

Author:
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

<table>
<thead>
<tr>
<th>Required Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPMessageHandler[]</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
abstract public SOAPMessageHandler[] value()

value

public abstract SOAPMessageHandler[] value

Deprecated.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.soap  Class SOAPPart

java.lang.Object  
  javax.xml.soap.SOAPPart

All Implemented Interfaces:
  Document, Node

public abstract class SOAPPart
  extends Object
  implements Document, Node

Implements: org.w3c.dom.Document, Node

SOAPMessage  SOAP  SOAP
SOAPMessage  SOAP  SOAPPart

SOAPPart  MIME  MIME  Content-Id  Content-Location
Content-Type  Content-Type  "text/xml"
SOAP  Mime  Content-Type  "text/xml"  SOAP  XML
"text/xml"  "text/xml"

SOAP  MIME  Content-Type  "text/xml"  SOAP
"text/xml"  MIME  Content-Type

SOAPMessage.getSOAPPart  SOAPMessage  SOAPPart
message  SOAPMessage  SOAP

SOAPPart soapPart = message.getSOAPPart();

SOAPPart  SOAPEnvelope  SOAPEnvelope  SOAPBody
SOAPHeader  SOAPPart  getEnvelope  SOAPEnvelope

The container for the SOAP-specific portion of a SOAPMessage object. All messages are required to have a SOAP part, so when a SOAPMessage
object is created, it will automatically have a SOAPPart object.

A SOAPPart object is a MIME part and has the MIME headers Content-Id, Content-Location, and Content-Type. Because the value of Content-Type must be "text/xml", a SOAPPart object automatically has a MIME header of Content-Type with its value set to "text/xml". The value must be "text/xml" because content in the SOAP part of a message must be in XML format. Content that is not of type "text/xml" must be in an AttachmentPart object rather than in the SOAPPart object.

When a message is sent, its SOAP part must have the MIME header Content-Type set to "text/xml". Or, from the other perspective, the SOAP part of any message that is received must have the MIME header Content-Type with a value of "text/xml".

A client can access the SOAPPart object of a SOAPMessage object by calling the method SOAPMessage.getSOAPPart. The following line of code, in which message is a SOAPMessage object, retrieves the SOAP part of a message.

```java
SOAPPart soapPart = message.getSOAPPart();
```

A SOAPPart object contains a SOAPEnvelope object, which in turn contains a SOAPBody object and a SOAPHeader object. The SOAPPart method getEnvelope can be used to retrieve the SOAPEnvelope object.

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE_NODE, CDATA_SECTION_NODE, COMMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_FRAGMENT_NODE, DOCUMENT_NODE,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_CONTAINED_BY, DOCUMENT_POSITION_CONTAINS,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_DISCONNECTED, DOCUMENT_POSITION_FOLLOWING,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_IMPLEMENTATION_SPECIFIC,</td>
</tr>
<tr>
<td>DOCUMENT_POSITION_PRECEDING, DOCUMENT_TYPE_NODE, ELEMENT_NODE,</td>
</tr>
<tr>
<td>ENTITY_NODE, ENTITY REFERENCE_NODE, NOTATION_NODE,</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION_NODE, TEXT_NODE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface org.w3c.dom.Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>Node</td>
</tr>
</tbody>
</table>
### Constructor Summary

**SOAPPart()**

### Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract void addMimeHeader(String name, String value)</td>
<td>Creates a MimeHeader object with the specified name and value and adds it to this SOAPPart object.</td>
</tr>
<tr>
<td>abstract Iterator getAllMimeHeaders()</td>
<td>Retrieves all the headers for this SOAPPart object as an iterator over the MimeHeader objects.</td>
</tr>
<tr>
<td>abstract Source getContent()</td>
<td>Returns the content of the SOAPEnvelope as a JAXP Source object.</td>
</tr>
<tr>
<td>String getContentId()</td>
<td>Retrieves the value of the MIME header whose name is &quot;Content-Id&quot;.</td>
</tr>
<tr>
<td>String getContentLocation()</td>
<td>Retrieves the value of the MIME header whose name is &quot;Content-Location&quot;.</td>
</tr>
<tr>
<td>abstract SOAPEnvelope getEnvelope()</td>
<td>Gets the SOAPEnvelope object associated with this SOAPPart object.</td>
</tr>
<tr>
<td>abstract Iterator getMatchingMimeHeaders(String[] names)</td>
<td>Retrieves all MimeHeader objects that match a name in the given array.</td>
</tr>
<tr>
<td>String getMimeHeader(String name)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>abstract String[]</td>
<td>Gets all the values of the MimeHeader object in this SOAPPart object that is identified by the given String.</td>
</tr>
<tr>
<td>abstract Iterator getNonMatchingMimeHeaders(String[] names)</td>
<td>Retrieves all MimeHeader objects whose name does not match a name in the given array.</td>
</tr>
<tr>
<td>abstract void removeAllMimeHeaders()</td>
<td>Removes all the MimeHeader objects for this SOAPEnvelope object.</td>
</tr>
<tr>
<td>abstract void removeMimeHeader(String header)</td>
<td>Removes all MIME headers that match the given name.</td>
</tr>
<tr>
<td>abstract void setContent(Source source)</td>
<td>Sets the content of the SOAPEnvelope object with the data from the given Source object.</td>
</tr>
<tr>
<td>void setContentId(String contentId)</td>
<td>Sets the value of the MIME header named &quot;Content-Id&quot; to the given String.</td>
</tr>
<tr>
<td>void setContentLocation(String contentLocation)</td>
<td>Sets the value of the MIME header &quot;Content-Location&quot; to the given String.</td>
</tr>
<tr>
<td>abstract void setMimeHeader(String name, String value)</td>
<td>Changes the first header entry that matches the given header name so that its value is the given value, adding a new header with the given name and value if no existing header is a match.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Methods inherited from interface org.w3c.dom.Document:
- adoptNode, createAttribute, createAttributeNS, createCDATASection, createComment, createDocumentFragment, createElement, createElementNS, createEntityReference, createProcessingInstruction, createTextNode, getDoctype, getDocumentElement, getDocumentURI, getDomConfig, getElementById, getElementsByTagName, getElementsByNameNS, getImplementation,
Methods inherited from interface org.w3c.dom.\texttt{Node}

- \texttt{appendChild}, \texttt{cloneNode}, \texttt{compareDocumentPosition}, \texttt{getAttributes}, \texttt{getBaseURI}, \texttt{getChildNodes}, \texttt{getFeature}, \texttt{getFirstChild}, \texttt{getLastChild}, \texttt{getLocalName}, \texttt{getNamespaceURI}, \texttt{getNextSibling}, \texttt{getNodeName}, \texttt{getNodeType}, \texttt{getNodeValue}, \texttt{getOwnerDocument}, \texttt{getParentNode}, \texttt{getPrefix}, \texttt{getPreviousSibling}, \texttt{getTextContent}, \texttt{getUserData}, \texttt{hasAttributes}, \texttt{hasChildNodes}, \texttt{insertBefore}, \texttt{isDefaultNamespace}, \texttt{isEqualNode}, \texttt{isSameNode}, \texttt{isSupported}, \texttt{lookupNamespaceURI}, \texttt{lookupPrefix}, \texttt{normalize}, \texttt{removeChild}, \texttt{replaceChild}, \texttt{setNodeValue}, \texttt{setPrefix}, \texttt{setTextContent}, \texttt{setUserData}

Methods inherited from interface javax.xml.soap.\texttt{Node}

- \texttt{detachNode}, \texttt{getParentElement}, \texttt{getValue}, \texttt{recycleNode}, \texttt{setParentElement}, \texttt{setValue}

Methods inherited from interface org.w3c.dom.\texttt{Node}

- \texttt{appendChild}, \texttt{cloneNode}, \texttt{compareDocumentPosition}, \texttt{getAttributes}, \texttt{getBaseURI}, \texttt{getChildNodes}, \texttt{getFeature}, \texttt{getFirstChild}, \texttt{getLastChild}, \texttt{getLocalName}, \texttt{getNamespaceURI}, \texttt{getNextSibling}, \texttt{getNodeName}, \texttt{getNodeType}, \texttt{getNodeValue}, \texttt{getOwnerDocument}, \texttt{getParentNode}, \texttt{getPrefix}, \texttt{getPreviousSibling}, \texttt{getTextContent},\texttt{getUserData}, \texttt{hasAttributes}, \texttt{hasChildNodes}, \texttt{insertBefore}, \texttt{isDefaultNamespace}, \texttt{isEqualNode}, \texttt{isSameNode}, \texttt{isSupported}, \texttt{lookupNamespaceURI}, \texttt{lookupPrefix}, \texttt{normalize}, \texttt{removeChild}, \texttt{replaceChild}, \texttt{setNodeValue}, \texttt{setPrefix}, \texttt{setTextContent}, \texttt{setUserData}

Constructor Detail

public \texttt{SOAPPart()}

\texttt{SOAPPart}

public \texttt{SOAPPart()}

\texttt{SOAPPart}
public abstract SOAPEnvelope getEnvelope() throws SOAPException

SOAPPart  SOAP
return  SOAPPart  SOAPEnvelope
Throws  SOAPException:  SOAP

getEnvelope

public abstract SOAPEnvelope getEnvelope() throws SOAPException

SOAPPart  SOAP
return  SOAPPart  SOAPEnvelope
Throws  SOAPException:  SOAP

getContentId

public String getContentId()
"Content-Id"  MIME
return  String "Content-Id"  MIME
See also  setContentId

getContentId

public String getContentId()

Retrieves the value of the MIME header whose name is "Content-Id".

Returns:
public String getContentLocation()
"Content-Location" MIME

    return String "Content-Location" MIME

See also
setContentLocation

getContentLocation

public String getContentLocation()

    Retrieves the value of the MIME header whose name is "Content-Location".

    Returns:
    a String giving the value of the MIME header whose name is "Content-Location"

    See Also:
    setContentLocation

setContentId

public void setContentId(String contentId)
"Content-Id" MIME String
    contentId String MIME "Content-Id"

    Throws
    IllegalArgumentException: id

    See also
    getContentId

setContentId

public void setContentId(String contentId)

    Sets the value of the MIME header named "Content-Id" to the given
String.

**Parameters:**
- `contentId` - a String giving the value of the MIME header "Content-Id"

**Throws:**
- `IllegalArgumentException` - if there is a problem in setting the content id

**See Also:**
- `getContentId()`

```java
public void setContentLocation(String contentLocation)
MIME "Content-Location" String
contentLocation String MIME "Content-Location"
Throws IllegalArgumentException: getContentLocation
See also
```

**setContentLocation**

```java
public void setContentLocation(String contentLocation)
```

Sets the value of the MIME header "Content-Location" to the given String.

**Parameters:**
- `contentLocation` - a String giving the value of the MIME header "Content-Location"

**Throws:**
- `IllegalArgumentException` - if there is a problem in setting the content location.

**See Also:**
- `getContentLocation()`

```java
abstract public void removeMimeHeader(String header)
MIME
```
removeMimeHeader

public abstract void removeMimeHeader(String header)

Removes all MIME headers that match the given name.

Parameters:
  header - a String giving the name of the MIME header(s) to be removed

abstract public void removeAllMimeHeaders()

SOAPEnvelope MimeHeader

removeAllMimeHeaders

public abstract void removeAllMimeHeaders()

Removes all the MimeHeader objects for this SOAPEnvelope object.

abstract public String[] getMimeHeader(String name)

SOAPPart String MimeHeader
  name "Content-Type"
  return String
  See also setMimeHeader

getMimeHeader

public abstract String[] getMimeHeader(String name)

Gets all the values of the MimeHeader object in this SOAPPart object that is identified by the given String.
Parameters:

- name - the name of the header; example: "Content-Type"

Returns:

- a String array giving all the values for the specified header

See Also:

- setMimeHeader(java.lang.String, java.lang.String)

---

abstract public void setMimeHeader(String name, String value)

RFC822 US-ASCII

<table>
<thead>
<tr>
<th>name</th>
<th>String</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>String</td>
</tr>
</tbody>
</table>

| Throws             | IllegalArgumentException: MIME |

See also

- getMimeHeader

---

setMimeHeader

public abstract void setMimeHeader(String name, String value)

Changes the first header entry that matches the given header name so that its value is the given value, adding a new header with the given name and value if no existing header is a match. If there is a match, this method clears all existing values for the first header that matches and sets the given value instead. If more than one header has the given name, this method removes all of the matching headers after the first one.

Note that RFC822 headers can contain only US-ASCII characters.

Parameters:

- name - a String giving the header name for which to search
- value - a String giving the value to be set. This value will be
substituted for the current value(s) of the first header that is a match if there is one. If there is no match, this value will be the value for a new MimeHeader object.

Throws:

IllegalArgumentException - if there was a problem with the specified mime header name or value

See Also:

getMimeHeader(java.lang.String)

abstract public void addMimeHeader(String name, String value)

MimeHeader SOAPPart MimeHeader

RFC822 US-ASCII

name String
value String

Throws IllegalArgumentException: MIME

addMimeHeader

public abstract void addMimeHeader(String name, String value)

Creates a MimeHeader object with the specified name and value and adds it to this SOAPPart object. If a MimeHeader with the specified name already exists, this method adds the specified value to the already existing value(s).

Note that RFC822 headers can contain only US-ASCII characters.

Parameters:

name - a String giving the header name
value - a String giving the value to be set or added

Throws:

IllegalArgumentException - if there was a problem with the
abstract public java.util.Iterator<E> getAllMimeHeaders()

MimeHeader SOAPPart
return Iterator SOAPPart Mime

g GetAllMimeHeaders

public abstract Iterator getAllMimeHeaders()

Retrieves all the headers for this SOAPPart object as an iterator over the MimeHeader objects.

Returns:
- an Iterator object with all of the Mime headers for this SOAPPart object

abstract public java.util.Iterator<E> getMatchingMimeHeaders(String[] names)

MimeHeader
names String Mime
return Iterator MIME

g getMatchingMimeHeaders

public abstract Iterator getMatchingMimeHeaders(String[] names)

Retrieves all MimeHeader objects that match a name in the given array.

Parameters:
- names - a String array with the name(s) of the MIME headers to be returned

Returns:
abstract public java.util.Iterator<E>
getNonMatchingMimeHeaders(String[] names)

Retrieves all MimeHeader objects whose name does not match a name in the given array.

Parameters:
names - a String array with the name(s) of the MIME headers not to be returned

Returns:
all of the MIME headers in this SOAPPart object except those that match one of the names in the given array. The nonmatching MIME headers are returned as an Iterator object.

abstract public void
setContent(javax.xml.transform.Source source) throws SOAPException

Throws:
SOAPException:
getContent
public abstract void setContent(Source source) throws SOAPException

Sets the content of the SOAPEnvelope object with the data from the given Source object. This Source must contain a valid SOAP document.

Parameters:
source - the javax.xml.transform.Source object with the data to be set

Throws:
SOAPException - if there is a problem in setting the source

See Also:
getContent()
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.registry.infomodel  Interface SpecificationLink

All Superinterfaces:
   ExtensibleObject, RegistryObject

public interface SpecificationLink
extends RegistryObject

Implements: RegistryObject

SpecificationLink  ServiceBinding  ServiceBinding
ServiceBinding  SpecificationLink  WSDL  CORBA  IDL
UDDI  tModelInstanceInfo  instanceDetails

See also  javax.xml.registry.infomodel.Concept

A SpecificationLink provides the linkage between a ServiceBinding and one of its technical specifications that describes how to use the service using the ServiceBinding. For example, a ServiceBinding may have a SpecificationLink instance that describes how to access the service using a technical specification in the form of a WSDL document or a CORBA IDL document. It serves the same purpose as the union of the tModelInstanceInfo and instanceDetails structures in UDDI.

Author:
   Farrukh S. Najmi

See Also:
   Concept

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getServiceBinding()</td>
<td>Gets the parent ServiceBinding for this SpecificationLink.</td>
</tr>
<tr>
<td>getSpecificationObject()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td>RegistryObject</td>
<td>Gets the specification object for this object.</td>
</tr>
<tr>
<td>InternationalString</td>
<td>getUsageDescription() Gets the description of usage parameters.</td>
</tr>
<tr>
<td>Collection</td>
<td>getUsageParameters() Gets any usage parameters.</td>
</tr>
<tr>
<td>void</td>
<td>setSpecificationObject(RegistryObject obj) Sets the specification object for this object.</td>
</tr>
<tr>
<td>void</td>
<td>setUsageDescription(InternationalString usageDescription) Sets the description of usage parameters.</td>
</tr>
<tr>
<td>void</td>
<td>setUsageParameters(Collection usageParameters) Sets any usage parameters.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.registry.infomodel.RegistryObject

addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getAttribute, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML

Methods inherited from interface javax.xml.registry.infomodel.ExtensibleObject

addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

Method Detail

public RegistryObject getSpecificationObject() throws JAXRException
getSpecificationObject

*RegistryObject* getSpecificationObject() throws *JAXRException*

Gets the specification object for this object.

**Capability Level:** 0

**Returns:**
the RegistryObject that is the specification object. For a UDDI provider the specification object must be a Concept with no parent. For an ebXML provider it is likely to be an ExtrinsicObject.

**Throws:**
*JAXRException* - If the JAXR provider encounters an internal error

public void setSpecificationObject(*RegistryObject* obj) throws *JAXRException*

*obj* RegistryObject UDDI Concept ebXML ExtrinsicObject

**Throws**
*JAXRException* - JAXR
**setSpecificationObject**

```java
void setSpecificationObject(RegistryObject obj)
    throws JAXRException
```

Sets the specification object for this object.

**Capability Level: 0**

**Parameters:**
- `obj` - the RegistryObject that is the specification object. For a UDDI provider the specification object must be a Concept with no parent. For an ebXML provider it is likely to be an ExtrinsicObject.

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public InternationalString getUsageDescription() throws JAXRException**

```java
0

    return null

Throws `JAXRException`: JAXR
```

**getUsageDescription**

```java
InternationalString getUsageDescription()
    throws JAXRException
```

Gets the description of usage parameters. Default is an empty String.
Returns:
the usage description for this object, which must not be null

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void setUsageDescription(InternationalString usageDescription) throws JAXRException

Sets the description of usage parameters.

Parameters:
  usageDescription - the description of usage parameters for this object

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getUsageParameters() throws JAXRException
getUsageParameters

`Collection getUsageParameters() throws JAXRException`

Gets any usage parameters. A usage parameter is an arbitrary String that describes how to use the technical specification accessed via this SpecificationLink. Each parameter is a String.

**Capability Level: 0**

**Returns:**
the Collection of String instances. The Collection may be empty but not null.

**Throws:**
`JAXRException` - If the JAXR provider encounters an internal error

---

`public void setUsageParameters(java.util.Collection<E> usageParameters) throws JAXRException`
setUsageParameters

void setUsageParameters(Collection usageParameters)
    throws JAXRException

Sets any usage parameters. Each parameter is a String

Capability Level: 0

Parameters:
    usageParameters - the Collection of usage parameter Strings

Throws:
    JAXRException - If the JAXR provider encounters an internal error

public ServiceBinding getServiceBinding() throws JAXRException

SpecificationLink  ServiceBinding 0

    return ServiceBinding

Throws JAXRException: JAXR

getServiceBinding

ServiceBinding  getServiceBinding() throws JAXRException

Gets the parent ServiceBinding for this SpecificationLink.

Capability Level: 0

Returns:
    the parent ServiceBinding within which this object is composed

Throws:
    JAXRException - If the JAXR provider encounters an internal
javax.persistence  Annotation Type SqlResultSetMapping

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface SqlResultSetMapping

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

**SQL**

```java
Query q = em.createNativeQuery(
    "SELECT o.id AS order_id, " +
    "o.quantity AS order_quantity, " +
    "o.item AS order_item, " +
    "i.name AS item_name, " +
    "FROM Order o, Item i " +
    "WHERE (order_quantity > 25) AND (order_item = i.id)",
    "OrderResults");

@SqlResultSetMapping(name="OrderResults",
    entities={
        @EntityResult(entityClass=com.acme.Order.class, fields={
            @FieldResult(name="id", column="order_id"),
            @FieldResult(name="quantity", column="order_quantity",
            @FieldResult(name="item", column="order_item"))},
        columns={
            @ColumnResult(name="item_name")}
    )
)
```

**since** Java Persistence 1.0

This annotation is used to specify the mapping of the result of a native SQL query.

Example:

```java
Query q = em.createNativeQuery(
```
"SELECT o.id AS order_id, " +  
"o.quantity AS order_quantity, " +  
"o.item AS order_item, " +  
"i.name AS item_name, " +  
"FROM Order o, Item i " +  
"WHERE (order_quantity > 25) AND (order_item = i.id)",  
"OrderResults");  

@SqlResultSetMapping(name="OrderResults",  
   entities={  
      @EntityResult(entityClass=com.acme.Order.class, fields={  
         @FieldResult(name="id", column="order_id"),  
         @FieldResult(name="quantity", column="order_quantity"  
         @FieldResult(name="item", column="order_item"}}),  
   columns={  
      @ColumnResult(name="item_name")}  
})  

Since:  
Java Persistence 1.0

---

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>The name given to the result set mapping, and used to refer to it in the methods of the Query API.</td>
</tr>
</tbody>
</table>

### Optional Element Summary

<table>
<thead>
<tr>
<th>ColumnResult[]</th>
<th>columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>ColumnResult[]</td>
<td>Specifies the result set mapping to scalar values.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EntityResult[]</th>
<th>entities</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityResult[]</td>
<td>Specifies the result set mapping to entities.</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String name()
Query API

name

```java
public abstract String name
```

The name given to the result set mapping, and used to refer to it in the methods of the Query API.

abstract public `EntityResult[]` entities()

entities

```java
public abstract EntityResult[] entities
```

Specifies the result set mapping to entities.

**Default:**

```java
{}
```

abstract public `ColumnResult[]` columns()

columns

```java
public abstract ColumnResult[] columns
```

Specifies the result set mapping to scalar values.

**Default:**

```java
{}
```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.persistence Annotation Type
SqlResultSetMappings

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface SqlResultSetMappings

Implements: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

SqlResultSetMapping

since Java Persistence 1.0

This annotation is used to define one or more SqlResultSetMapping.

Since: Java Persistence 1.0

### Required Element Summary

| SqlResultSetMapping[] value | One or more SqlResultSetMapping |

### Element Detail

abstract public SqlResultSetMapping[] value()

value
public abstract SqlResultSetMapping[] value

One or more SqlResultSetMapping.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.stream.events Interface StartDocument

All Superinterfaces:
   XMLEvent, XMLStreamConstants

public interface StartDocument
   extends XMLEvent

Implements: XMLEvent

version 1.0

An interface for the start document event

Version:
   1.0

Author:
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

Fields inherited from interface
javax.xml.stream.XMLStreamConstants
ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT,
END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE,
NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE,
START_DOCUMENT, START_ELEMENT

Method Summary

   boolean encodingSet()
      Returns true if CharacterEncodingScheme was set in the
encoding declaration of the document

| String `getCharacterEncodingScheme()` | Returns the encoding style of the XML data |
| String `getSystemId()` | Returns the system ID of the XML data |
| String `getVersion()` | Returns the version of XML of this XML stream |
| boolean `isStandalone()` | Returns if this XML is standalone |
| boolean `standaloneSet()` | Returns true if the standalone attribute was set in the encoding declaration of the document. |

Methods inherited from interface
javax.xml.stream.events.XMLEvent
asCharacters, asEndElement, asStartElement, getEventType, getLocation, getSchemaType, isAttribute, isCharacters, isEndDocument, isEndElement, isEntityReference, isNamespace, isProcessingInstruction, isStartDocument, isStartElement, writeAsEncodedUnicode

Method Detail

public String `getSystemId()`
XML ID

    return ID ""

getSystemId

`String getSystemId()`

Returns the system ID of the XML data

Returns:
the system ID, defaults to ""

public String getCharacterEncodingScheme()
XML
        return "UTF-8"

getCharacterEncodingScheme

String getCharacterEncodingScheme()

        Returns the encoding style of the XML data

        Returns:
        the character encoding, defaults to "UTF-8"

public boolean encodingSet()
CharacterEncodingScheme true

encodingSet

boolean encodingSet()

        Returns true if CharacterEncodingScheme was set in the encoding declaration of the document

public boolean isStandalone()
XML true
        return XML "no"

isStandalone
boolean isStandalone()

Returns if this XML is standalone

Returns:
the standalone state of XML, defaults to "no"

public boolean standaloneSet()

ture

standaloneSet

boolean standaloneSet()

Returns true if the standalone attribute was set in the encoding declaration of the document.

public String getVersion()

XML XML

return XML "1.0"

getVersion

String getVersion()

Returns the version of XML of this XML stream

Returns:
the version of XML, defaults to "1.0"
javax.xml.stream.events Interface StartElement

All Superinterfaces:  
   XMLEvent, XMLStreamConstants

public interface StartElement
extends XMLEvent

Implements: XMLEvent

StartElement StartElement
   version 1.0

The StartElement interface provides access to information about start elements. A StartElement is reported for each Start Tag in the document.

Version:  
   1.0
Author:  
   Copyright (c) 2003 by BEA Systems. All Rights Reserved.

Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants
ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, 
END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, 
NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, 
START_DOCUMENT, START_ELEMENT

Method Summary

getAttributeByName(QName name)
### Method Detail

```java
public javax.xml.namespace.QName getName()

    return

getName
```

```java
QName getName()
```
Get the name of this event

**Returns:**
the qualified name of this event

```java
public java.util.Iterator<E> getAttributes()
```

**getAttributes**

**Iterator** getAttributes()

Returns an Iterator of non-namespace declared attributes declared on this `START_ELEMENT`, returns an empty iterator if there are no attributes. The iterator must contain only implementations of the `javax.xml.stream.Attribute` interface. Attributes are fundamentally unordered and may not be reported in any order.

**Returns:**
a readonly Iterator over `Attribute` interfaces, or an empty iterator

```java
public java.util.Iterator<E> getNamespaces()
```

**getNamespaces**

```java
Iterator getNamespaceContext()
```

**javax.xml.stream.Namespace**

```java
return Iterator
```
getNamespaces

Iterator getNamespaces()

Returns an Iterator of namespaces declared on this element. This Iterator does not contain previously declared namespaces unless they appear on the current START_ELEMENT. Therefore this list may contain redeclared namespaces and duplicate namespace declarations. Use the getNamespaceContext() method to get the current context of namespace declarations.

The iterator must contain only implementations of the javax.xml.stream.Namespace interface.

A Namespace is a Attribute. One can iterate over a list of namespaces as a list of attributes. However this method returns only the list of namespaces declared on this START_ELEMENT and does not include the attributes declared on this START_ELEMENT. Returns an empty iterator if there are no namespaces.

Returns:
a readonly Iterator over Namespace interfaces, or an empty iterator

public Attribute getAttributeByQName(javax.xml.namespace.QName name)

name qname
return null

getAttributeByQName

Attribute getAttributeByQName(QName name)

Returns the attribute referred to by this name

Parameters:
name - the qname of the desired name

Returns:
the attribute corresponding to the name value or null

public javax.xml.namespace.NamespaceContext
getNamespaceContext()
NamespaceContext StartElement
return

getNamespaceContext

NamespaceContext getNamespaceContext()

Gets a read-only namespace context. If no context is available this method will return an empty namespace context. The NamespaceContext contains information about all namespaces in scope for this StartElement.

Returns:
the current namespace context

public String getNamespaceURI(String prefix)
null

prefix
return URI null

getNamespaceURI

String getNamespaceURI(String prefix)

Gets the value that the prefix is bound to in the context of this element. Returns null if the prefix is not bound in this context

Parameters:
prefix - the prefix to lookup

Returns:
the uri bound to the prefix or null
javax.ejb  Annotation Type Stateful

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Stateful

**Implements**: Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean

Component-defining annotation for a stateful session bean.

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><strong>description</strong></td>
</tr>
<tr>
<td>String</td>
<td><strong>mappedName</strong></td>
</tr>
<tr>
<td>String</td>
<td>A product specific name(e.g. global JNDI name) that this session bean should be mapped to.</td>
</tr>
<tr>
<td>String</td>
<td><strong>name</strong></td>
</tr>
<tr>
<td></td>
<td>ejb-name for this bean.</td>
</tr>
</tbody>
</table>

abstract public String name()

Bean  ejb-name

**name**

public abstract String name

    ejb-name for this bean.
abstract public String mappedName()
Bean  JNDI

mappedName

public abstract String mappedName

A product specific name (e.g. global JNDI name) that this session bean should be mapped to. Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

Default: 

abstract public String description()

description

public abstract String description

Default: 

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
javax.management.j2ee.statistics Interface StatefulSessionBeanStats

All Superinterfaces:
   EJBStats, SessionBeanStats, Stats

public interface StatefulSessionBeanStats
extends SessionBeanStats

Implements: SessionBeanStats

Bean

Specifies the statistics provided by a stateful session bean.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RangeStatistic</td>
<td>getPassiveCount()</td>
</tr>
<tr>
<td></td>
<td>Number of beans that are in the passive state.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.management.j2ee.statistics.SessionBeanStats
getMethodReadyCount

Methods inherited from interface javax.management.j2ee.statistics.EJBStats
getCreateCount, getRemoveCount

Methods inherited from interface javax.management.j2ee.statistics.Stats
getStatistic, getStatisticNames, getStatistics
public RangeStatistic getPassiveCount()
Bean

getPassiveCount

RangeStatistic getPassiveCount()

Number of beans that are in the passive state.
javax.faces.component Interface StateHolder

All Known Implementing Classes:

- DateTimeConverter
- DoubleRangeValidator
- EnumConverter
- HtmlColumn
- HtmlCommandButton
- HtmlCommandLink
- HtmlDataTable
- HtmlForm
- HtmlGraphicImage
- HtmlInputHidden
- HtmlInputSecret
- HtmlInputText
- HtmlInputTextarea
- HtmlMessage
- HtmlMessages
- HtmlOutputFormat
- HtmlOutputLabel
- HtmlOutputLink
- HtmlOutputText
- HtmlPanelGrid
- HtmlPanelGroup
- HtmlSelectBooleanCheckbox
- HtmlSelectManyCheckbox
- HtmlSelectManyListbox
- HtmlSelectManyMenu
- HtmlSelectOneListbox
- HtmlSelectOneMenu
- HtmlSelectOneRadio
- LengthValidator
- LongRangeValidator
- MethodExpressionActionListener
- MethodExpressionValidator
- MethodExpressionValueChangeListener
- NumberConverter
- UIColumn
- UICommand
- UIComponent
- UIComponentBase
- UIData
- UIForm
- UIGraphic
- UIInput
- UIMessage
- UIMessages
- UINamingContainer
- UIOutput
- UIPanel
- UIParameter
- UISelectBoolean
- UISelectItem
- UISelectItems
- UISelectMany
- UISelectOne
- UIViewRoot

---

public interface StateHolder

Implemented by: DateTimeConverter, DoubleRangeValidator, EnumConverter, LengthValidator, LongRangeValidator, MethodExpressionActionListener, MethodExpressionValidator, MethodExpressionValueChangeListener, NumberConverter, UIComponent

#saveState  #restoreState  #saveState  #restoreState
This interface is implemented by classes that need to save their state between requests.

An implementor **must** implement both
`saveState(javax.faces.context.FacesContext)` and
`restoreState(javax.faces.context.FacesContext, java.lang.Object)` methods in this class, since these two methods have a tightly coupled contract between themselves. In other words, if there is an inheritance hierarchy, it is not permissible to have the
`saveState(javax.faces.context.FacesContext)` and
`restoreState(javax.faces.context.FacesContext, java.lang.Object)` methods reside at different levels of the hierarchy.

An implementor must have a public no-args constructor.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td><strong>isTransient()</strong></td>
</tr>
<tr>
<td></td>
<td>If true, the Object implementing this interface must not participate in state saving or restoring.</td>
</tr>
<tr>
<td>void</td>
<td><strong>restoreState(FacesContext context, Object state)</strong></td>
</tr>
<tr>
<td></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td><strong>saveState(FacesContext context)</strong></td>
</tr>
<tr>
<td></td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td><strong>setTransient(boolean newTransientValue)</strong></td>
</tr>
<tr>
<td></td>
<td>Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.</td>
</tr>
</tbody>
</table>

---

### Method Detail

**public Object saveState(FacesContext context)**
Serializable

**StateHolder**

UIComponent  

#saveState

facet  

[javax.faces.application.StateManager](https://docs.oracle.com/javaee/spec_9/javax/faces/application/StateManager.html)

Object state = component.saveState(facesContext);

component

Serializable

**Throws**

NullPointeException: context null

**saveState**

Object saveState([FacesContext](https://docs.oracle.com/javaee/spec_9/javax/faces/context/FacesContext.html) context)

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the saveState([javax.faces.context.FacesContext](https://docs.oracle.com/javaee/spec_9/javax/faces/context/FacesContext.html)) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the [StateManager](https://docs.oracle.com/javaee/spec_9/javax/faces/application/StateManager.html)

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable
public void restoreState(FacesContext context, Object state)

Object

StateHolder

UIComponent #restoreState

Throws: NullPointerException - if context is null

restoreState

void restoreState(FacesContext context, Object state)

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Throws: NullPointerException - if either context or state are null

public boolean isTransient()

true Object
isTransient

boolean isTransient()

If true, the Object implementing this interface must not participate in state saving or restoring.

public void setTransient(boolean newTransientValue)

Object

newTransientValue Object boolean true

setTransient

void setTransient(boolean newTransientValue)

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

Parameters:
newTransientValue - boolean pass true if this Object will participate in state saving or restoring, otherwise pass false.
javax.ejb  **Annotation Type Stateless**

```java
@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface Stateless
```

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

Bean

Component-defining annotation for a stateless session bean.

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><strong>description</strong></td>
</tr>
<tr>
<td>String</td>
<td><strong>mappedName</strong></td>
</tr>
<tr>
<td>String</td>
<td><strong>name</strong></td>
</tr>
</tbody>
</table>

A product specific name(e.g. global JNDI name) that this session bean should be mapped to.

ejb-name for this bean.

### abstract public String name()

**Bean**  
ejb-name

### name

```java
public abstract String name
```

ejb-name for this bean.
abstract public String mappedName()
Bean  JNDI

mappedName

class

public abstract String mappedName

A product specific name(e.g. global JNDI name) that this session bean should be mapped to. Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

Default:

""

abstract public String description()

description

class

public abstract String description

Default:

""
to license terms.

PS:
javax.management.j2ee.statistics Interface
StatelessSessionBeanStats

All Superinterfaces:
   EJBStats, SessionBeanStats, Stats

public interface StatelessSessionBeanStats
extends SessionBeanStats

Implements: SessionBeanStats

Bean

Specifies the statistics provided by a stateless session bean.

Method Summary

| Methods inherited from interface javax.management.j2ee.statistics.SessionBeanStats |
| getMethodReadyCount |

| Methods inherited from interface javax.management.j2ee.statistics.EJBStats |
| getCountCreate, getCountRemove |

| Methods inherited from interface javax.management.j2ee.statistics.Stats |
| getStatistic, getStatisticNames, getStatistics |
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.application ClassStateManager

java.lang.Object
   - javax.faces.application.StateManager

Direct Known Subclasses:
   StateManagerWrapper

public abstract class StateManager
   extends Object

Inner classes: StateManager.SerializedView
Extended by: StateManagerWrapper

StateManager StateManager Application RenderKit
  StateManager ( ResponseStateManager ) RenderKit

StateManager directs the process of saving and restoring the view between requests. The StateManager instance for an application is retrieved from the Application instance, and thus cannot know any details of the markup language created by the RenderKit being used to render a view. The StateManager utilizes a helper object (ResponseStateManager), that is provided by the RenderKit implementation and is therefore aware of the markup language details.

<table>
<thead>
<tr>
<th>Nested Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>StateManager.SerializedView</td>
</tr>
<tr>
<td>Deprecated. This class was not marked Serializable in the 1.0 version of the spec. It was also not a static inner class, so it can't be made to be Serializable. Therefore, it is being deprecated in version 1.2 of the spec. The replacement is to use an implementation dependent object.</td>
</tr>
</tbody>
</table>
Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String STATE_SAVING_METHOD_CLIENT</td>
<td>Constant value for the initialization parameter named by the STATE_SAVING_METHOD_PARAM_NAME that indicates state saving should take place on the client.</td>
</tr>
<tr>
<td>static String STATE_SAVING_METHOD_PARAM_NAME</td>
<td>The ServletContext init parameter consulted by theStateManager to tell where the state should be saved.</td>
</tr>
<tr>
<td>static String STATE_SAVING_METHOD_SERVER</td>
<td>Constant value for the initialization parameter named by the STATE_SAVING_METHOD_PARAM_NAME that indicates state saving should take place on the server.</td>
</tr>
</tbody>
</table>

Constructor Summary

StateManager()  

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected Object getComponentStateToSave(FacesContext context)</td>
<td>Deprecated. the distinction between tree structure and component state is now an implementation detail. implementation returns null.</td>
</tr>
<tr>
<td>protected Object getTreeStructureToSave(FacesContext context)</td>
<td>Deprecated. the distinction between tree structure and component state is now an implementation detail. implementation returns null.</td>
</tr>
<tr>
<td>boolean isSavingStateInClient(FacesContext context)</td>
<td></td>
</tr>
<tr>
<td>protected void RestoreComponentState(FacesContext context, UIViewRoot viewRoot, String renderKitId)</td>
<td>Deprecated. the distinction between tree structure and component state is now an implementation detail. implementation does nothing.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>protected UIViewRoot</strong> restoreTreeStructure(FacesContext context, String renderKitId)**</td>
<td>Deprecated. The distinction between tree structure and component state is now an implementation detail. Implementation returns null.</td>
</tr>
<tr>
<td><strong>abstract UIViewRoot</strong> restoreView(FacesContext context, String viewId, String renderKitId)**</td>
<td>Restore the tree structure and the component state for the specified viewId, in an implementation dependent manner, and return the restored UIViewRoot.</td>
</tr>
<tr>
<td><strong>StateManager.SerializedView saveSerializedView(FacesContext context)</strong></td>
<td>Deprecated. This has been replaced by saveView(javax.faces.context.FacesContext). The implementation returns null.</td>
</tr>
<tr>
<td><strong>Object saveView(FacesContext context)</strong></td>
<td>Return an opaque object containing sufficient information for this same instance to restore the state of the current UIViewRoot on a subsequent request.</td>
</tr>
<tr>
<td><strong>void writeState(FacesContext context, Object state)</strong></td>
<td>Save the state represented in the specified object instance, in an implementation dependent manner.</td>
</tr>
<tr>
<td><strong>void writeState(FacesContext context, StateManager.SerializedView state)</strong></td>
<td>Deprecated. This method has been replaced by writeState(javax.faces.context.FacesContext,javax.faces.context.FacesContext,StateManager.SerializedView). The default implementation of this method does nothing.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**: clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Field Detail**
The `ServletContext` init parameter consulted by the `StateManager` to tell where the state should be saved. Valid values are given as the values of the constants: `STATE_SAVING_METHOD_CLIENT` or `STATE_SAVING_METHOD_SERVER`.

If this parameter is not specified, the default value is the value of the constant `STATE_SAVING_METHOD_CLIENT`.

**See Also:**
- [Constant Field Values](#)

---

**STATE_SAVING_METHOD_CLIENT**

Constant value for the initialization parameter named by the `STATE_SAVING_METHOD_PARAM_NAME` that indicates state saving should take place on the client.

**See Also:**
- [Constant Field Values](#)

---

**STATE_SAVING_METHOD_SERVER**

Constant value for the initialization parameter named by the `STATE_SAVING_METHOD_PARAM_NAME` that indicates state saving should take place on the server.
public StateManager()

StateManager

public StateManager()

Method Detail

public StateManager.SerializedView saveSerializedView(FacesContext context)

null

transient true facet

null id NamingContainer

context

Throws IllegalStateException: NamingContainer facet

deprecated

saveSerializedView

public StateManager.SerializedView saveSerializedView(FacesContext context)

Deprecated. this has been replaced by
saveView( javax.faces.context.FacesContext ). The default implementation returns null.

Return the tree structure and component state information for the view contained in the specified FacesContext instance as an object of type StateManager.SerializedView. If there is no state information to be saved, return null instead.

Components may opt out of being included in the serialized view by setting their transient property to true. This must cause the component itself, as well as all of that component's children and facets, to be omitted from the saved tree structure and component state information.

This method must also enforce the rule that, for components with non-null ids, all components that are descendants of the same nearest NamingContainer must have unique identifiers.

Parameters:
context - FacesContext for the current request

Throws:
IllegalStateException - if more than one component or facet within the same NamingContainer in this view has the same non-null component id

public Object saveView( FacesContext context )

Object UIViewRoot java.io.Serializable null

transient true facet

null id NamingContainer

StateManager #saveSerializedView Object 0
structure 1SerializedView state

custom FacesContext
saveView

public Object saveView(FacesContext context)

Return an opaque object containing sufficient information for this same instance to restore the state of the current UIViewRoot on a subsequent request. The returned object must implement java.io.Serializable. If there is no state information to be saved, return null instead.

Components may opt out of being included in the serialized view by setting their transient property to true. This must cause the component itself, as well as all of that component’s children and facets, to be omitted from the saved tree structure and component state information.

This method must also enforce the rule that, for components with non-null ids, all components that are descendants of the same nearest NamingContainer must have unique identifiers.

For backwards compatibility with existing StateManager implementations, the default implementation of this method calls saveSerializedView(javax.faces.context.FacesContext) and creates and returns a two element object array with element zero containing the structure property and element one containing the state property of the SerializedView.

Parameters:
context - FacesContext for the current request

Throws:
IllegalStateException - if more than one component or facet within the same NamingContainer in this view has the same non-null component id

Since:
protected Object getTreeStructureToSave(FacesContext context)

    saveSerializedView()   Serializable facet

    transient   true facet

        context   FacesContext
        deprecated

getTreeStructureToSave

protected Object getTreeStructureToSave(FacesContext context)

    Deprecated. the distinction between tree structure and component state is now an implementation detail. The default implementation returns null.

    Convenience method, which must be called by saveSerializedView(), to construct and return a Serializable object that represents the structure of the entire component tree (including children and facets) of this view.

    Components may opt-out of being included in the tree structure by setting their transient property to true. This must cause the component itself, as well as all of that component's children and facets, to be omitted from the saved tree structure information.

    Parameters:

        context - FacesContext for the current request

protected Object getComponentStateToSave(FacesContext context)
saveSerializedView()   Serializable facet

transient   true facet

context
FacesContext
deprecated

gGetComponentStateToSave

protected Object getComponentStateToSave(FacesContext context)

Deprecated. the distinction between tree structure and component state is now an implementation detail. The default implementation returns null.

Convenience method, which must be called by saveSerializedView(), to construct and return a Serializable object that represents the state of all component properties, attributes, and attached objects, for the entire component tree (including children and facets) of this view.

Components may opt-out of being included in the component state by setting their transient property to true. This must cause the component itself, as well as all of that component's children and facets, to be omitted from the saved component state information.

Parameters:
context - FacesContext for the current request

public void writeState(FacesContext context, Object state) throws java.io.IOException

Object

ResponseStateManager   writeState()   RenderKit
ResponseWriter
writeState

public void writeState(FacesContext context, Object state)
    throws IOException

Save the state represented in the specified state Object instance, in an implementation dependent manner.

This method will typically simply delegate the actual writing to the writeState() method of the ResponseStateManager instance provided by the RenderKit being used to render this view. This method assumes that the caller has positioned the ResponseWriter at the correct position for the saved state to be written.

For backwards compatibility with existing StateManager implementations, the default implementation of this method checks if the argument is an instance of Object [] of length greater than or equal to two. If so, it creates a SerializedView instance with the tree structure coming from element zero and the component state coming from element one and calls through to writeState(javax.faces.context.FacesContext, javax.faces.application.StateManager.SerializedView) If not, does nothing.

Parameters:

context - FacesContext for the current request
state - the Serializable state to be written, as returned by saveSerializedView(javax.faces.context.FacesContext)

Throws:
public void writeState(FacesContext context,
StateManager.SerializedView state) throws java.io.IOException

SerializedView

StateManager.STATE_SAVING_METHOD_PARAMETER_NAME

init
ResponseStateManager writeState()
RenderKit ResponseWriter

context FacesContext
state

deprecated #writeState(javax.faces.context.FacesContext,java.lang.Object)

writeState

public void writeState(FacesContext context,
StateManager.SerializedView state)
throws IOException

Deprecated. This method has been replaced by
The default implementation of this method does nothing.

Save the state represented in the specified SerializedView instance,
in an implementation dependent manner.

This method must consult the context initialization parameter named
by the symbolic constant
StateManager.STATE_SAVING_METHOD_PARAMETER_NAME to determine
whether state should be saved on the client or the server. If not
present, client side state saving is assumed.

If the init parameter indicates that client side state saving should be
used, this method must delegate the actual writing to the
writeState() method of the ResponseStateManager instance provided
by the RenderKit being used to render this view. This method
assumes that the caller has positioned the ResponseWriter at the
correct position for the saved state to be written.

Parameters:
   context - FacesContext for the current request
   state - the serialized state to be written

Throws:
    IOException

abstract public UIViewRoot restoreView(FacesContext context, String viewId, String renderKitId)

    viewId      UIViewRoot  viewId      null
    StateManager.STATE_SAVING_METHOD_PARAMETER_NAME

    init      ResponseStateManager
    getTreeStructureToRestore()  null
    getComponentStateToRestore()   RenderKit

    context       FacesContext
    viewId
    renderKitId   renderKitId
    null

    Throws    IllegalArgumentException:  renderKitId  null
public abstract UIViewRoot restoreView(FacesContext context, String viewId, String renderKitId)

Restore the tree structure and the component state of the view for the specified viewId, in an implementation dependent manner, and return the restored UIViewRoot. If there is no saved state information available for this viewId, return null instead.

This method must consult the context initialization parameter named by the symbolic constant StateManager.STATE_SAVING_METHOD_PARAMETER_NAME to determine whether state should be saved on the client or the server. If not present, client side state saving is assumed.

If the init parameter indicates that client side state saving should be used, this method must call the getTreeStructureToRestore() and (if the previous method call returned a non-null value) getComponentStateToRestore() methods of the ResponseStateManager instance provided by the RenderKit responsible for this view.

Parameters:
context - FacesContext for the current request
viewId - View identifier of the view to be restored
renderKitId - the renderKitId used to render this response. Must not be null.

Throws:
IllegalArgumentException - if renderKitId IS null.

protected UIViewRoot restoreTreeStructure(FacesContext context, String viewId, String renderKitId)

restoreView() UIViewRoot facet null

context FacesContext
restoreTreeStructure

protected `UIViewRoot restoreTreeStructure(FacesContext context, String viewId, String renderKitId)`

**Deprecated.** *the distinction between tree structure and component state is now an implementation detail. The default implementation returns null.*

Convenience method, which must be called by `restoreView()`, to construct and return a `UIViewRoot` instance (populated with children and facets) representing the tree structure of the component tree being restored. If no saved state information is available, return `null` instead.

**Parameters:**
- `context` - `FacesContext` for the current request
- `viewId` - View identifier of the view to be restored
- `renderKitId` - the renderKitId used to render this response. Must not be `null`.

**Throws:**
- `IllegalArgumentException` - if `renderKitId` is `null`.

protected void restoreComponentState(FacesContext context, UIViewRoot viewRoot, String renderKitId)

`restoreView()`
restoreComponentState

protected void restoreComponentState(FacesContext context,
                                      UIViewRoot viewRoot,
                                      String renderKitId)

Deprecated. the distinction between tree structure and component state is now an implementation detail. The default implementation does nothing.

Convenience method, which must be called by restoreView(), to restore the attributes, properties, and attached objects of all components in the restored component tree.

Parameters:
- context - FacesContext for the current request
- viewRoot - UIViewRoot returned by a previous call to restoreTreeStructure()
- renderKitId - the renderKitId used to render this response. Must not be null.

Throws:
- IllegalArgumentException - if renderKitId is null.

public boolean isSavingStateInClient(FacesContext context)

return #STATE_SAVING_METHOD_PARAM_NAME ServletContext init
#STATE_SAVING_METHOD_CLIENT true false

Throws
- NullPointerException: context null

isSavingStateInClient

public boolean isSavingStateInClient(FacesContext context)
Returns:

true if and only if the value of the ServletContext init parameter
named by the value of the constant
STATE_SAVING_METHOD_PARAM_NAME is equal to the value of the
constant STATE_SAVING_METHOD_CLIENT. false otherwise.

Throws:

NullPointerException - if context is null.
javax.faces.application  **Class StateManagerWrapper**

**java.lang.Object**
- javax.faces.application.StateManager
  - javax.faces.application.StateManagerWrapper

public abstract class **StateManagerWrapper**

extends **StateManager**

**Extends:** **StateManager**

StateManager  StateManager  StateManager

#**getWrapped**

since 1.2

Provides a simple implementation of **StateManager** that can be subclassed by developers wishing to provide specialized behavior to an existing **StateManager** instance. The default implementation of all methods is to call through to the wrapped **StateManager**.

Usage: extend this class and override **getWrapped()** to return the instance we are wrapping.

**Since:**

1.2

---

**Nested Class Summary**

**Nested classes/interfaces inherited from class javax.faces.application.StateManager**

<table>
<thead>
<tr>
<th><strong>Nested classes/interfaces</strong></th>
<th><strong>StateManager.SerializedView</strong></th>
</tr>
</thead>
</table>

Field Summary

Fields inherited from class javax.faces.application.StateManager
STATE_SAVING_METHOD_CLIENT, STATE_SAVING_METHOD_PARAM_NAME, STATE_SAVING_METHOD_SERVER

Constructor Summary

StateManagerWrapper()
StateManager.restoreTreeStructure(javax.faces.context.FacesContext, String, String) on the wrapped StateManager object.

restoreView(FacesContext context, String viewId)

The default behavior of this method is to call StateManager.restoreView(javax.faces.context.FacesContext, String) on the wrapped StateManager object.

UIViewRoot

StateManager.restoreView(FacesContext context, String viewId)

The default behavior of this method is to call StateManager.restoreView(javax.faces.context.FacesContext, String) on the wrapped StateManager object.

StateManager.SerializedView

StateManager.saveSerializedView(FacesContext context)

The default behavior of this method is to call StateManager.saveSerializedView(javax.faces.context.FacesContext) on the wrapped StateManager object.

Object

StateManager.saveView(FacesContext context)

The default behavior of this method is to call StateManager.saveView(javax.faces.context.FacesContext) on the wrapped StateManager object.

void

StateManager.writeState(FacesContext context, Object state)

The default behavior of this method is to call StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object) on the wrapped StateManager object.

void

StateManager.writeState(FacesContext context, StateManager.SerializedView state)

The default behavior of this method is to call StateManager.writeState(javax.faces.context.FacesContext, javax.faces.application.StateManager.SerializedView) on the wrapped StateManager object.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public StateManagerWrapperWrapper()

StateManagerWrapper
public StateManagerWrapper()

### Method Detail

abstract protected **StateManager** getWrapped()

```java
return getWrapped()
```

getWrapped

protected abstract **StateManager** getWrapped()

`Returns:`

the instance that we are wrapping.

---

public **StateManager.SerializedView** saveSerializedView(FacesContext context)

```java
StateManager saveSerializedView(javax.faces.context.FacesContext)
```

since 1.2

See also **saveSerializedView(javax.faces.context.FacesContext)**

--

saveSerializedView

public **StateManager.SerializedView** saveSerializedView(FacesContext context)

The default behavior of this method is to call **StateManager.saveSerializedView(javax.faces.context.FacesContext)** on the wrapped **StateManager** object.

** Overrides:**

**saveSerializedView** in class **StateManager**

**Parameters:**
public Object saveView(FacesContext context)

StateManager
saveView(javax.faces.context.FacesContext)
since 1.2
See also saveView(javax.faces.context.FacesContext)

saveView

public Object saveView(FacesContext context)

The default behavior of this method is to call
StateManager.saveView(javax.faces.context.FacesContext) on the
wrapped StateManager object.

Overrides:
saveView in class StateManager

Parameters:
context - FacesContext for the current request

Since:
1.2
See Also:
StateManager.saveView(javax.faces.context.FacesContext)

protected Object getTreeStructureToSave(FacesContext context)

StateManager
getTreeStructureToSave

protected Object getTreeStructureToSave(FacesContext context)

The default behavior of this method is to call
StateManager.getTreeStructureToSave(javax.faces.context.FacesContext)
on the wrapped StateManager object.

Overrides:
getTreeStructureToSave in class StateManager

Parameters:
context - FacesContext for the current request

Since:
1.2

See Also:
StateManager.getTreeStructureToSave(javax.faces.context.FacesContext)

getComponentStateToSave

protected Object getComponentStateToSave(FacesContext context)

StateManager

GetComponentStateToSave(javax.faces.context.FacesContext)

since
1.2

See also GetComponentStateToSave(javax.faces.context.FacesContext)

getComponentStateToSave

protected Object getComponentStateToSave(FacesContext context)
The default behavior of this method is to call `StateManager.getComponentStateToSave(javax.faces.context.FacesContext)` on the wrapped `StateManager` object.

**Overrides:**
- `StateManager.getComponentStateToSave` in class `StateManager`

**Parameters:**
- `context` - `FacesContext` for the current request

**Since:**
1.2

**See Also:**
- `StateManager.getComponentStateToSave(javax.faces.context.FacesContext)`

---

```java
public void writeState(FacesContext context, Object state)
throws java.io.IOException
```

- `StateManager`
- `writeState(javax.faces.context.FacesContext, java.lang.Object)`

**since**
1.2

**See also**
- `writeState(javax.faces.context.FacesContext, java.lang.Object)`

---

**writeState**

```java
public void writeState(FacesContext context, Object state)
throws IOException
```

The default behavior of this method is to call `StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object)` on the wrapped `StateManager` object.

**Overrides:**
- `writeState` in class `StateManager`

**Parameters:**
- `context` - `FacesContext` for the current request
public void writeState(FacesContext context, StateManager.SerializedView state) throws java.io.IOException

StateManager
writeState(javax.faces.context.FacesContext, javax.faces.application.StateManager.SerializedView)

since 1.2
See also writeState(javax.faces.context.FacesContext, javax.faces.application.StateManager.SerializedView)

writeState

public void writeState(FacesContext context, StateManager.SerializedView state) throws IOException

The default behavior of this method is to call StateManager.writeState(javax.faces.context.FacesContext, javax.faces.application.StateManager.SerializedView) on the wrapped StateManager object.

Overrides:
writeState in class StateManager

Parameters:
context - FacesContext for the current request
state - the serialized state to be written

Throws:
   IOException

Since:
   1.2

See Also:
   StateManager.writeState(javax.faces.context.FacesContext,
                           javax.faces.application.StateManager.SerializedView)

public UIViewRoot restoreView(FacesContext context, String viewId, String renderKitId)

StateManager
restoreView(javax.faces.context.FacesContext, String, String)

since 1.2

See also  restoreView(javax.faces.context.FacesContext, String, String)

restoreView

public UIViewRoot restoreView(FacesContext context,
                              String viewId,
                              String renderKitId)

The default behavior of this method is to call
StateManager.restoreView(javax.faces.context.FacesContext, String, String) on the wrapped StateManager object.

Specified by:
   restoreView in class StateManager

Parameters:
context - FacesContext for the current request
viewId - View identifier of the view to be restored
renderKitId - the renderKitId used to render this response. Must not be null.

Since:
protected `UIViewRoot` `restoreTreeStructure(FacesContext` context, String viewId, String renderKitId)

StateManager
`restoreTreeStructure(javax.faces.context.FacesContext, String, String)`

since 1.2
See also `restoreTreeStructure(javax.faces.context.FacesContext, String, String)`

The default behavior of this method is to call
StateManager.restoreTreeStructure(javax.faces.context.FacesContext, String, String) on the wrapped StateManager object.

Overrides:
`restoreTreeStructure` in class StateManager

Parameters:
context - FacesContext for the current request
viewId - View identifier of the view to be restored
renderKitId - the renderKitId used to render this response. Must not be null.

Since:
1.2

See Also:
StateManager.restoreTreeStructure(javax.faces.context.FacesContext, String, String)
protected void restoreComponentState(FacesContext context, UIViewRoot viewRoot, String renderKitId)

StateManager
restoreComponentState(javax.faces.context.FacesContext,
javax.faces.component.UIViewRoot, String)

since 1.2
See also
StateManager.restoreComponentState(javax.faces.context.FacesContext,
javax.faces.component.UIViewRoot, String)

restoreComponentState

protected void restoreComponentState(FacesContext context,
UIViewRoot viewRoot, String renderKitId)

The default behavior of this method is to call
StateManager.restoreComponentState(javax.faces.context.FacesCont
javax.faces.component.UIViewRoot, String) on the wrapped
StateManager object.

Overrides:
restoreComponentState in class StateManager

Parameters:
context - FacesContext for the current request
viewRoot - UIViewRoot returned by a previous call to
restoreTreeStructure()
renderKitId - the renderKitId used to render this response. Must
not be null.

Since:
1.2
See Also:
StateManager.restoreComponentState(javax.faces.context.FacesCont
javax.faces.component.UIViewRoot, String)
public boolean isSavingStateInClient(FacesContext context)

StateManager

isSavingStateInClient(javax.faces.context.FacesContext)

since 1.2

See also isSavingStateInClient(javax.faces.context.FacesContext)

isSavingStateInClient

public boolean isSavingStateInClient(FacesContext context)

The default behavior of this method is to call StateManager.isSavingStateInClient(javax.faces.context.FacesContext) on the wrapped StateManager object.

Overrides:

isSavingStateInClient in class StateManager

Returns:

true if and only if the value of the ServletContext init parameter named by the value of the constant StateManager.STATE_SAVING_METHOD_PARAM_NAME is equal to the value of the constant StateManager.STATE_SAVING_METHOD_CLIENT. false otherwise.

Since:

1.2

See Also:

StateManager.isSavingStateInClient(javax.faces.context.FacesContext)
PS:
javax.enterprise.deploy.shared  **Class StateType**

```java
java.lang.Object
    ↓ javax.enterprise.deploy.shared.StateType

public class StateType
    extends Object

StateType  DeploymentStatus

Class StateTypes defines enumeration values for the DeploymentStatus object.

Author:
    rsearls
```

### Field Summary

<table>
<thead>
<tr>
<th>static StateType</th>
<th>COMPLETED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The action operation has completed normally.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static StateType</th>
<th>FAILED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The action operation has failed.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static StateType</th>
<th>RELEASED</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The DeploymentManager is running in disconnected mode.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static StateType</th>
<th>RUNNING</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The action operation is running normally.</td>
</tr>
</tbody>
</table>

### Constructor Summary

<table>
<thead>
<tr>
<th>protected StateType(int value)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construct a new enumeration value with the given integer value.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected StateType[] getEnumValueTable()</td>
<td>Returns the enumeration value table for class StateType</td>
</tr>
<tr>
<td>protected int getOffset()</td>
<td>Returns the lowest integer value used by this enumeration value's enumeration class.</td>
</tr>
<tr>
<td>static StateType getStateType(int value)</td>
<td>Return an object of the specified value.</td>
</tr>
<tr>
<td>protected String[] getStringTable()</td>
<td>Returns the string table for class StateType</td>
</tr>
<tr>
<td>int getValue()</td>
<td>Returns this enumeration value's integer value.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Return the string name of this StateType or the integer value if outside the bounds of the table</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

### Field Detail

**RUNNING**

public static final StateType RUNNING

The action operation is running normally.
public static final StateType COMPLETED

    The action operation has completed normally.

---

public static final StateType FAILED

    The action operation has failed.

---

public static final StateType RELEASED

    The DeploymentManager is running in disconnected mode.

### Constructor Detail

**protected StateType(int value)**

    value

**StateType**

**protected StateType(int value)**

    Construct a new enumeration value with the given integer value.
Parameters:
   value - Integer value.

Method Detail

public int getValue()

   return

getValue

public int getValue()

   Returns this enumeration value's integer value.

   Returns:
       the value

protected String[] getStringTable()

getStateType

protected String[] getStringTable()

   Returns the string table for class StateType

protected StateType[] getEnumValueTable()

getStateEnumValueTable
protected `StateType[]` `getEnumValueTable()`

    Returns the enumeration value table for class `StateType`

---

**public static `StateType` `getStateType(int value)`**

    `value`

**getStateType**

**public static `StateType` `getStateType(int value)`**

    Return an object of the specified value.

    **Parameters:**
    `value` - a designator for the object.

---

**public String `toString()`**

**`StateType`**

**toString**

**public `String` `toString()`**

    Return the string name of this `StateType` or the integer value if outside the bounds of the table

    **Overrides:**
    `toString` in class `Object`

---

**protected `int` `getOffset()`**
protected int getOffset()

Returns the lowest integer value used by this enumeration value's enumeration class.

The default implementation returns 0.

Returns:
   the offset of the lowest enumeration value.
javax.management.j2ee.statistics Interface Statistic

All Known Subinterfaces:
  BoundaryStatistic, BoundedRangeStatistic, CountStatistic, RangeStatistic, TimeStatistic

public interface Statistic

Implemented by: BoundaryStatistic, CountStatistic, RangeStatistic, TimeStatistic

Statistic Stats

The Statistic model and its sub-models specify the data models which are required to be used to provide the performance data described by the specific attributes in the Stats models.

Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getDescription()</td>
<td>A human-readable description of the Statistic.</td>
</tr>
<tr>
<td>long</td>
<td>getLastSampleTime()</td>
<td>The time of the last measurement represented as a long, whose value is the number of milliseconds since January 1, 1970, 00:00:00.</td>
</tr>
<tr>
<td>String</td>
<td>getName()</td>
<td>The name of this Statistic.</td>
</tr>
<tr>
<td>long</td>
<td>getStartTime()</td>
<td>The time of the first measurement represented as a long, whose value is the number of milliseconds since January 1, 1970, 00:00:00.</td>
</tr>
<tr>
<td>String</td>
<td>getUnit()</td>
<td>The unit of measurement for this Statistic.</td>
</tr>
</tbody>
</table>
public String getName()
Statistic

getName

String getName()
The name of this Statistic.

public String getUnit()
Statistic TimeStatistic
"HOUR""MINUTE""SECOND""MILLISECOND""MICROSECOND""NANOSECOND"

getUnit

String getUnit()
The unit of measurement for this Statistic. Valid values for TimeStatistic measurements are "HOUR", "MINUTE", "SECOND", "MILLISECOND", "MICROSECOND" and "NANOSECOND".

public String getDescription()
Statistic

description

String getDescription()
String getDescription()

A human-readable description of the Statistic.

public long getStartTime()
long 1970 1 1 00:00:00

getStartTime

long getStartTime()

The time of the first measurement represented as a long, whose
value is the number of milliseconds since January 1, 1970, 00:00:00.

public long getLastSampleTime()
long 1970 1 1 00:00:00

getLastSampleTime

long getLastSampleTime()

The time of the last measurement represented as a long, whose
value is the number of milliseconds since January 1, 1970, 00:00:00.
PS:
javax.management.j2ee.statistics Interface Stats

All Known Subinterfaces:

EJBStats, EntityBeanStats, JCACollectionPoolStats,
JCAConnectionStats, JCAStats, JDBCConnectionPoolStats,
JDBCConnectionStats, JDBCStats, JMSCollectionStats,
JMSCollectionStats, JMSEndpointStats, JMSProducerStats,
JMSSessionStats, JMSStats, JTAStats, JVMStats,
MessageDrivenBeanStats, ServletStats, SessionBeanStats,
StatefulSessionBeanStats, StatelessSessionBeanStats, URLStats

public interface Stats

Implemented by: EJBStats, JCACollectionStats, JCAStats,
JDBCConnectionStats, JDBCStats, JMSCollectionStats,
JMSCollectionStats, JMSEndpointStats, JMSSessionStats, JMSStats, JTAStats, JVMStats,
ServletStats, URLStats

Stats

The Stats model and its submodels specify performance data attributes
for each of the specific managed object types.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Statistic getStatistic(String statisticName)</strong></td>
</tr>
<tr>
<td><strong>String[] getStatisticNames()</strong></td>
</tr>
<tr>
<td><strong>Statistic[] getStatistics()</strong></td>
</tr>
</tbody>
</table>
Method Detail

public Statistic getStatistic(String statisticName)

getStatistic

Statistic getStatistic(String statisticName)

Get a Statistic by name.

public String[] getStatisticNames()

String Stats Statistic
statisticNames Stats null statisticNames
Statistic

getStatisticNames

String[] getStatisticNames()

Returns an array of Strings which are the names of the attributes from the specific Stats submodel that this object supports. Attributes named in the list must correspond to attributes that will return a Statistic object of the appropriate type which contains valid performance data. The return value of attributes in the Stats submodel that are not included in the statisticNames list must be null. For each name in the statisticNames list there must be one Statistic with the same name in the statistics list.

public Statistic[] getStatistics()
getStatistics

\texttt{Statistic[]} \texttt{getStatistics()}

Returns an array containing all of the Statistic objects supported by this Stats object.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.transaction  **Interface Status**

```java
public interface Status
```

Status

The Status interface defines static variables used for transaction status codes.

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int STATUS_ACTIVE</td>
<td>A transaction is associated with the target object and it is in the active state.</td>
</tr>
<tr>
<td>static int STATUS_COMMITTED</td>
<td>A transaction is associated with the target object and it has been committed.</td>
</tr>
<tr>
<td>static int STATUS_COMMITTING</td>
<td>A transaction is associated with the target object and it is in the process of committing.</td>
</tr>
<tr>
<td>static int STATUS_MARKED_ROLLBACK</td>
<td>A transaction is associated with the target object and it has been marked for rollback, perhaps as a result of a setRollbackOnly operation.</td>
</tr>
<tr>
<td>static int STATUS_NO_TRANSACTION</td>
<td>No transaction is currently associated with the target object.</td>
</tr>
<tr>
<td>static int STATUS_PREPARED</td>
<td>A transaction is associated with the target object and it has been prepared.</td>
</tr>
<tr>
<td>static int STATUS_PREPARING</td>
<td>A transaction is associated with the target object and it is</td>
</tr>
</tbody>
</table>
in the process of preparing.

<table>
<thead>
<tr>
<th>static int</th>
<th>STATUS_ROLLED_BACK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A transaction is associated with the target object and the outcome has been determined to be rollback.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>STATUS_ROLLING_BACK</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A transaction is associated with the target object and it is in the process of rolling back.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static int</th>
<th>STATUS_UNKNOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A transaction is associated with the target object but its current status cannot be determined.</td>
</tr>
</tbody>
</table>

**Field Detail**

**STATUS_ACTIVE**

static final int **STATUS_ACTIVE**

A transaction is associated with the target object and it is in the active state. An implementation returns this status after a transaction has been started and prior to a Coordinator issuing any prepares, unless the transaction has been marked for rollback.

**See Also:**

Constant Field Values

**STATUS_MARKED_ROLLBACK**

static final int **STATUS_MARKED_ROLLBACK**

A transaction is associated with the target object and it has been marked for rollback, perhaps as a result of a setRollbackOnly
operation.

See Also:

Constant Field Values

STATUS_PREPARED

static final int STATUS_PREPARED

    A transaction is associated with the target object and it has been prepared. That is, all subordinates have agreed to commit. The target object may be waiting for instructions from a superior as to how to proceed.

See Also:

Constant Field Values

STATUS_COMMITTED

static final int STATUS_COMMITTED

    A transaction is associated with the target object and it has been committed. It is likely that heuristics exist; otherwise, the transaction would have been destroyed and NoTransaction returned.

See Also:

Constant Field Values

STATUS_ROLLED_BACK

static final int STATUS_ROLLED_BACK
A transaction is associated with the target object and the outcome has been determined to be rollback. It is likely that heuristics exist; otherwise, the transaction would have been destroyed and NoTransaction returned.

See Also:
Constant Field Values

STATUS_UNKNOWN

static final int STATUS_UNKNOWN

A transaction is associated with the target object but its current status cannot be determined. This is a transient condition and a subsequent invocation will ultimately return a different status.

See Also:
Constant Field Values

STATUS_NO_TRANSACTION

static final int STATUS_NO_TRANSACTION

No transaction is currently associated with the target object. This will occur after a transaction has completed.

See Also:
Constant Field Values

STATUS_PREPARING
static final int STATUS_PREPARING

A transaction is associated with the target object and it is in the process of preparing. An implementation returns this status if it has started preparing, but has not yet completed the process. The likely reason for this is that the implementation is probably waiting for responses to prepare from one or more Resources.

See Also:
Constant Field Values

_________________________________________________________________

STATUS_COMMITTING

static final int STATUS_COMMITTING

A transaction is associated with the target object and it is in the process of committing. An implementation returns this status if it has decided to commit but has not yet completed the committing process. This occurs because the implementation is probably waiting for responses from one or more Resources.

See Also:
Constant Field Values

_________________________________________________________________

STATUS_ROLLING_BACK

static final int STATUS_ROLLING_BACK

A transaction is associated with the target object and it is in the process of rolling back. An implementation returns this status if it has decided to rollback but has not yet completed the process. The implementation is probably waiting for responses from one or more Resources.
See Also:

Constant Field Values

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
An abstract class that models a message store and its access protocol, for storing and retrieving messages. Subclasses provide actual implementations.

Note that Store extends the Service class, which provides many common methods for naming stores, connecting to stores, and listening to connection events.

Version:
1.28, 07/05/04

Author:
John Mani, Bill Shannon

See Also:
Service, ConnectionEvent, StoreEvent
### Field Summary

Fields inherited from class javax.mail.Service
debug, session, url

### Constructor Summary

protected Store(Session session, URLName url)
Constructor.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>addFolderListener(FolderListener l)</td>
<td>Add a listener for Folder events on any Folder object obtained from this Store.</td>
</tr>
<tr>
<td>void</td>
<td>addStoreListener(StoreListener l)</td>
<td>Add a listener for StoreEvents on this Store.</td>
</tr>
<tr>
<td>abstract Folder</td>
<td>getDefaultFolder()</td>
<td>Returns a Folder object that represents the 'root' of the default namespace presented to the user by the Store.</td>
</tr>
<tr>
<td>abstract Folder</td>
<td>getFolder(String name)</td>
<td>Return the Folder object corresponding to the given name.</td>
</tr>
<tr>
<td>abstract Folder</td>
<td>getFolder(URLName url)</td>
<td>Return a closed Folder object, corresponding to the given URLName.</td>
</tr>
<tr>
<td>Folder[]</td>
<td>getPersonalNamespaces()</td>
<td>Return a set of folders representing the personal namespaces for the current user.</td>
</tr>
<tr>
<td>Folder[]</td>
<td>getSharedNamespaces()</td>
<td>Return a set of folders representing the shared namespaces.</td>
</tr>
<tr>
<td>Folder[]</td>
<td>getUserNamespaces(String user)</td>
<td>Return a set of folders representing the namespaces for user.</td>
</tr>
</tbody>
</table>
protected void notifyFolderListeners(int type, Folder folder)
Notify all FolderListeners.

protected void notifyFolderRenamedListeners(Folder oldF, Folder newF)
Notify all FolderListeners about the renaming of a folder.

protected void notifyStoreListeners(int type, String message)
Notify all StoreListeners.

void removeFolderListener(FolderListener l)
Remove a listener for Folder events.

void removeStoreListener(StoreListener l)
Remove a listener for Store events.

Methods inherited from class javax.mail.Service
addConnectionListener, close, connect, connect, connect, connect, finalize, getURLName, isConnected, notifyConnectionListeners, protocolConnect, queueEvent, removeConnectionListener, setConnected, setURLName, toString

Methods inherited from class java.lang.Object
clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

protected Store(Session session, URLName urlName)

session
Store Session
urlName
Store URLName

Store

protected Store(Session session, URLName urlName)

Constructor.
Parameters:
- session - Session object for this Store.
- urlName - URLName object to be used for this Store.

Method Detail

abstract public Folder getDefaultFolder() throws MessagingException

Stores

public abstract Folder getDefaultFolder()

Returns a Folder object that represents the 'root' of the default namespace presented to the user by the Store.

Returns:
the root Folder

Throws:
- IllegalStateException - if this Store is not connected.
- MessagingException

abstract public Folder getFolder(String name) throws MessagingException

Folder Store Folder Folder

name Folder Store ""

return Folder
getFolder

public abstract Folder getFolder(String name) throws MessagingException

Return the Folder object corresponding to the given name. Note that a Folder object is returned even if the named folder does not physically exist on the Store. The exists() method on the folder object indicates whether this folder really exists.

Folder objects are not cached by the Store, so invoking this method on the same name multiple times will return that many distinct Folder objects.

Parameters:
  name - The name of the Folder. In some Stores, name can be an absolute path if it starts with the hierarchy delimiter. Else it is interpreted relative to the 'root' of this namespace.

Returns:
  Folder object

Throws:
  IllegalStateException - if this Store is not connected. MessagingException

See Also:
  Folder.exists(), Folder.create(int)
getFolder

public abstract Folder getFolder(URLName url) throws MessagingException

Return a closed Folder object, corresponding to the given URLName. The store specified in the given URLName should refer to this Store object.

Implementations of this method may obtain the name of the actual folder using the getFile() method on URLName, and use that name to create the folder.

Parameters:
url - URLName that denotes a folder

Returns:
Folder object

Throws:
IllegalStateException - if this Store is not connected.
MessagingException

See Also:
URLName

public Folder[] getPersonalNamespaces() throws MessagingException

INBOX Store

getDefaultFolder

Throws
IllegalStateException: Store
Folder
getPersonalNamespaces

public Folder[] getPersonalNamespaces() throws MessagingException

Return a set of folders representing the personal namespaces for the current user. A personal namespace is a set of names that is considered within the personal scope of the authenticated user. Typically, only the authenticated user has access to mail folders in their personal namespace. If an INBOX exists for a user, it must appear within the user's personal namespace. In the typical case, there should be only one personal namespace for each user in each Store.

This implementation returns an array with a single entry containing the return value of the getDefaultFolder method. Subclasses should override this method to return appropriate information.

Returns:
array of Folder objects

Throws:
IllegalStateException - if this Store is not connected.
MessagingException

Since:
JavaMail 1.2

getUserNamespaces

public Folder[] getUserNamespaces(String user) throws MessagingException

user

Throws
IllegalStateException: Store
return Folder
getUserNamespaces

public Folder[] getUserNamespaces(String user) throws MessagingException

Return a set of folders representing the namespaces for user. The namespaces returned represent the personal namespaces for the user. To access mail folders in the other user's namespace, the currently authenticated user must be explicitly granted access rights. For example, it is common for a manager to grant to their secretary access rights to their mail folders.

This implementation returns an empty array. Subclasses should override this method to return appropriate information.

Returns:
array of Folder objects

Throws:
IllegalStateException - if this Store is not connected.
MessagingException

Since:
JavaMail 1.2

public Folder[] getSharedNamespaces() throws MessagingException

Throws
IllegalStateException: Store Folder return
since
JavaMail 1.2
getSharedNamespaces

public Folder[] getSharedNamespaces()

throws MessagingException

Return a set of folders representing the shared namespaces. A shared namespace is a namespace that consists of mail folders that are intended to be shared amongst users and do not exist within a user's personal namespace.

This implementation returns an empty array. Subclasses should override this method to return appropriate information.

Returns:
array of Folder objects

Throws:
IllegalStateException - if this Store is not connected.
MessagingException

Since:
JavaMail 1.2

public void addStoreListener(StoreListener l)

Store StoreEvent

StoreListener
/

See also javax.mail.event.StoreEvent

addStoreListener

public void addStoreListener(StoreListener l)

Add a listener for StoreEvents on this Store.

The default implementation provided here adds this listener to an internal list of StoreListeners.
public void removeStoreListener(StoreListener l)
Store

StoreListener

/ 

See also addStoreListener

removeStoreListener

public void removeStoreListener(StoreListener l)

Remove a listener for Store events.

The default implementation provided here removes this listener from the internal list of StoreListeners.

Parameters:
1 - the listener

See Also: addStoreListener(javax.mail.event.StoreListener)

protected void notifyStoreListeners(int type, String message)
StoreListenerStore StoreEvent

StoreListener
notifyStoreListeners

protected void notifyStoreListeners(int type, String message)

Notify all StoreListeners. Store implementations are expected to use this method to broadcast StoreEvents.

The provided default implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered StoreListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

public void addFolderListener(FolderListener l)

Add a listener for Folder events on any Folder object obtained from this Store. FolderEvents are delivered to FolderListeners on the affected Folder as well as to FolderListeners on the containing Store.

The default implementation provided here adds this listener to an internal list of FolderListeners.

Parameters:
public void removeFolderListener(FolderListener l)
Folder
FolderListener
/
See also addFolderListener

removeFolderListener

public void removeFolderListener(FolderListener l)

Remove a listener for Folder events.

The default implementation provided here removes this listener from the internal list of FolderListeners.

Parameters:
1 - the listener
See Also:
addFolderListener(javax.mail.event.FolderListener)

protected void notifyFolderListeners(int type, Folder folder)
FolderListenerStore  Folder
FolderListener

See also notifyFolderRenamedListeners
notifyFolderListeners

protected void notifyFolderListeners(int type, Folder folder)

Notify all FolderListeners. Store implementations are expected to use this method to broadcast Folder events.

The provided default implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered FolderListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.

Parameters:
- type - type of FolderEvent
- folder - affected Folder

See Also:
notifyFolderRenamedListeners(javax.mail.Folder, javax.mail.Folder)

protected void notifyFolderRenamedListeners(Folder oldF, Folder newF)

FolderListener Store Folder

FolderListener
- oldF
- newF

since JavaMail 1.1

notifyFolderRenamedListeners

protected void notifyFolderRenamedListeners(Folder oldF, Folder newF)

Notify all FolderListeners about the renaming of a folder. Store
implementations are expected to use this method to broadcast
Folder events indicating the renaming of folders.

The provided default implementation queues the event into an
internal event queue. An event dispatcher thread dequeues events
from the queue anddispatches them to the registered
FolderListeners. Note that the event dispatching occurs in a separate
thread, thus avoiding potential deadlock problems.

**Parameters:**
- oldF - the folder being renamed
- newF - the folder representing the new name.

**Since:**
JavaMail 1.1
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>

| SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |

| PREV CLASS | NEXT CLASS | FRAMES | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD | DETAIL: FIELD | CONSTR | METHOD |
javax.mail  **Class StoreClosedException**

java.lang.Object  
  └ java.lang.Throwable  
    └ java.lang.Exception  
      └ javax.mail.MessagingException  
        └ javax.mail.StoreClosedException

All Implemented Interfaces:
  Serializable

---

public class StoreClosedException

extends MessagingException

**Extends:** Throwable > Exception > MessagingException

Messaging  Store  Store  Messaging

Store  connect

getMessage()

This exception is thrown when a method is invoked on a Messaging object and the Store that owns that object has died due to some reason. This exception should be treated as a fatal error; in particular any messaging object belonging to that Store must be considered invalid.

The connect method may be invoked on the dead Store object to revive it.

The getMessage() method returns more detailed information about the error that caused this exception.

**Author:**
  John Mani
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>StoreClosedException(Store store)</td>
<td>Constructor</td>
</tr>
<tr>
<td>StoreClosedException(Store store, String message)</td>
<td>Constructor</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Store getStore()</td>
<td>Returns the dead Store object</td>
</tr>
</tbody>
</table>

#### Methods inherited from class javax.mail.MessagingException
- `getCause`, `getNextException`, `setNextException`, `toString`

#### Methods inherited from class java.lang.Throwable
- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`

#### Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Constructor Detail

```java
public StoreClosedException(Store store) Constructor
```

`store` Store
StoreClosedException

public StoreClosedException(Store store)

Constructor

Parameters:
store - The dead Store object

public StoreClosedException(Store store, String message)
Constructor

store
message

StoreClosedException

public StoreClosedException(Store store, String message)

Constructor

Parameters:
store - The dead Store object
message - The detailed error message

Method Detail

public Store.getStore()

getStore

public Store getStore()
Returns the dead Store object

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.event Class StoreEvent

java.lang.Object
   java.util.EventObject
      javax.mail.event.MailEvent
         javax.mail.event.StoreEvent

All Implemented Interfaces:
   Serializable

public class StoreEvent
   extends MailEvent

   Extends: java.util.EventObject > MailEvent

   Store ALERT NOTICEALERT

This class models notifications from the Store connection. These notifications can be ALERTS or NOTICES. ALERTS must be presented to the user in a fashion that calls the user's attention to the message.

Author:
   John Mani

See Also:
   Serialized Form

Field Summary

<p>| static int | ALERT          | Indicates that this message is an ALERT. |
| static int | NOTICE         | Indicates that this message is a NOTICE.  |</p>
<table>
<thead>
<tr>
<th>Field Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ALERT</strong></td>
</tr>
</tbody>
</table>

public static final int **ALERT**
Indicates that this message is an ALERT.

See Also:
  Constant Field Values

---

NOTICE

```java
public static final int NOTICE
```

Indicates that this message is a NOTICE.

See Also:
  Constant Field Values

---

```java
protected int type
```

The event type.

---

```java
protected String message
```

The message text to be presented to the user.

## Constructor Detail

```java
public StoreEvent(Store store, int type, String message)
```
StoreEvent

public StoreEvent(Store store, int type, String message)

Constructor.

Parameters:
store - The source Store

Method Detail

public int getMessageType()

    return
    See also ALERT, NOTICE

getMessageType

public int getMessageType()

    Return the type of this event.

    Returns:
    type
    See Also: ALERT, NOTICE

public String getMessage()
getMessage

public String getMessage()

Get the message from the Store.

Returns:
message from the Store

public void dispatch(Object listener)
StoreListener

dispatch

public void dispatch(Object listener)

Invokes the appropriate StoreListener method.

Specified by:
dispatch in class MailEvent
javax.mail.event Interface StoreListener

All Superinterfaces:
   EventListener

public interface StoreListener
extends EventListener

Implements: java.util.EventListener

Store Listener

This is the Listener interface for Store Notifications.

Author:  
   John Mani

Method Summary

<table>
<thead>
<tr>
<th>void notification(StoreEvent e)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invoked when the Store generates a notification event.</td>
</tr>
</tbody>
</table>

Method Detail

public void notification(StoreEvent e)
Store

See also ALERT, NOTICE

notification
void notification(StoreEvent e)

Invoked when the Store generates a notification event.

See Also:
    StoreEvent.ALERT, StoreEvent.NOTICE
public interface Streamable

Streamable Record Record
Streamable Record
Streamable Record
Streamable Record

since 0.8
See also javax.resource.cci.Record

Streamable interface enables a resource adapter to extract data from an input Record or set data into an output Record as a stream of bytes.

The Streamable interface provides a resource adapter's view of the data that has been set in a Record instance by a component.

The Streamable interface is not directly used by a component. It is used by a resource adapter implementation. A component uses Record or any derived interfaces to manage records.

Since: 0.8

Author: Rahul Sharma

See Also: Record

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>read(InputStream istream)</td>
<td>void</td>
</tr>
</tbody>
</table>
void write(OutputStream ostream)
Write fields of a Streamable object to an OutputStream

Method Detail

public void read(java.io.InputStream istream) throws java.io.IOException
InputStream  Streamable
  istream    Streamable  InputStream

read

void read(InputStream istream)
  throws IOException

Read data from an InputStream and initialize fields of a Streamable object.

Parameters:
  istream - InputStream that represents a resource adapter specific internal representation of fields of a Streamable object

Throws:
  IOException

public void write(java.io.OutputStream ostream) throws java.io.IOException
Streamable  OutputStream
  ostream    Streamable  OutputStream

write
void write(OutputStream ostream) throws IOException

Write fields of a Streamable object to an OutputStream

Parameters:
ostream - OutputStream that holds value of a Streamable object

Throws:
IOException

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
This interface declares a simple filter interface that one can create to filter XMLStreamReaders.
boolean accept(XMLStreamReader reader)

Tests whether the current state is part of this stream. This method will return true if this filter accepts this event and false otherwise. The method should not change the state of the reader when accepting a state.

**Parameters:**
reader - the event to test

**Returns:**
true if this filter accepts this event, false otherwise
javax.jms Interface StreamMessage

All Superinterfaces:
   Message

public interface StreamMessage extends Message

Implements: Message

StreamMessage Java Message
java.io.DataInputStream  java.io.DataOutputStream

   StreamMessage.writeInt(6)
StreamMessage.writeObject(new Integer(6))

   clearBody          reset          reset

   clearBody

   MessageNotReadableException

   MessageNotWriteableException

StreamMessage JMException  valueOf()  String
String

|                | boolean  byte  short  char  int  long  float  double  String  byte |
|----------------|----------------|-------------------|-----------------|-----------------|----------------|----------------|----------------|----------------|----------------|
|boolean         | X              |                   |                 |                 |                 |                 |                 |                 |                 |
|byte            |                 | X                 | X               | X               | X               |                 |                 |                 |                 |
|short           |                 |                   |                 |                 | X               | X               |                 |                 |                 |
|char            |                 |                   |                 |                 |                 | X               |                 |                 |                 |
|int             |                 |                   |                 |                 | X               | X               |                 |                 |                 |
|long            |                 |                   |                 |                 |                 |                 | X               |                 |                 |
|float           |                 |                   |                 |                 |                 |                 |                 | X               | X               |
|double          |                 |                   |                 |                 |                 |                 |                 |                 |                 |
A `StreamMessage` object is used to send a stream of primitive types in the Java programming language. It is filled and read sequentially. It inherits from the `Message` interface and adds a stream message body. Its methods are based largely on those found in `java.io.DataInputStream` and `java.io.DataOutputStream`.

The primitive types can be read or written explicitly using methods for each type. They may also be read or written generically as objects. For instance, a call to `StreamMessage.writeInt(6)` is equivalent to `StreamMessage.writeObject(new Integer(6))`. Both forms are provided, because the explicit form is convenient for static programming, and the object form is needed when types are not known at compile time.

When the message is first created, and when `clearBody` is called, the body of the message is in write-only mode. After the first call to `reset` has been made, the message body is in read-only mode. After a message has been sent, the client that sent it can retain and modify it without affecting the message that has been sent. The same message object can be sent multiple times. When a message has been received, the provider has called `reset` so that the message body is in read-only mode for the client.

If `clearBody` is called on a message in read-only mode, the message body is cleared and the message body is in write-only mode.

If a client attempts to read a message in write-only mode, a
MessageNotReadableException is thrown.

If a client attempts to write a message in read-only mode, a MessageNotWriteableException is thrown.

StreamMessage objects support the following conversion table. The marked cases must be supported. The unmarked cases must throw a JMSException. The String-to-primitive conversions may throw a runtime exception if the primitive’s `valueOf()` method does not accept it as a valid String representation of the primitive.

A value written as the row type can be read as the column type.

<table>
<thead>
<tr>
<th></th>
<th>boolean</th>
<th>byte</th>
<th>short</th>
<th>char</th>
<th>int</th>
<th>long</th>
<th>float</th>
<th>double</th>
<th>String</th>
<th>byte[]</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte</td>
<td></td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>short</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>char</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>int</td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>long</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>float</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>double</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>String</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>byte[]</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Attempting to read a null value as a primitive type must be treated as calling the primitive's corresponding `valueOf(String)` conversion method with a null value. Since `char` does not support a `String` conversion, attempting to read a null value as a `char` must throw a `NullPointerException`.

**Version:**
1.0 - 6 August 1998

**Author:**
Mark Hapner, Rich Burridge

**See Also:**
Session.createStreamMessage(), BytesMessage, MapMessage, Message, ObjectMessage, TextMessage
## Field Summary

Fields inherited from interface javax.jms.Message

<table>
<thead>
<tr>
<th>Field</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_DELIVERY_MODE</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_PRIORITY</td>
<td></td>
</tr>
<tr>
<td>DEFAULT_TIME_TO_LIVE</td>
<td></td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>readBoolean()</td>
<td>Reads a boolean from the stream message.</td>
</tr>
<tr>
<td>byte</td>
<td>readByte()</td>
<td>Reads a byte value from the stream message.</td>
</tr>
<tr>
<td>byte[]</td>
<td>readBytes(byte[] value)</td>
<td>Reads a byte array field from the stream message into the specified byte[] object (the read buffer).</td>
</tr>
<tr>
<td>char</td>
<td>readChar()</td>
<td>Reads a Unicode character value from the stream message.</td>
</tr>
<tr>
<td>double</td>
<td>readDouble()</td>
<td>Reads a double from the stream message.</td>
</tr>
<tr>
<td>float</td>
<td>readFloat()</td>
<td>Reads a float from the stream message.</td>
</tr>
<tr>
<td>int</td>
<td>readInt()</td>
<td>Reads a 32-bit integer from the stream message.</td>
</tr>
<tr>
<td>long</td>
<td>readLong()</td>
<td>Reads a 64-bit integer from the stream message.</td>
</tr>
<tr>
<td>Object</td>
<td>readObject()</td>
<td>Reads an object from the stream message.</td>
</tr>
<tr>
<td>short</td>
<td>readShort()</td>
<td>Reads a 16-bit integer from the stream message.</td>
</tr>
<tr>
<td>String</td>
<td>readString()</td>
<td>Reads a String from the stream message.</td>
</tr>
<tr>
<td>void</td>
<td>reset()</td>
<td>Puts the message body in read-only mode and repositions the stream to the beginning.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>writeBoolean(boolean value)</td>
<td>Writes a boolean to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeByte(byte value)</td>
<td>Writes a byte to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeBytes(byte[] value)</td>
<td>Writes a byte array field to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeBytes(byte[] value, int offset, int length)</td>
<td>Writes a portion of a byte array as a byte array field to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeChar(char value)</td>
<td>Writes a char to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeDouble(double value)</td>
<td>Writes a double to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeFloat(float value)</td>
<td>Writes a float to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeInt(int value)</td>
<td>Writes an int to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeLong(long value)</td>
<td>Writes a long to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeObject(Object value)</td>
<td>Writes an object to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeShort(short value)</td>
<td>Writes a short to the stream message.</td>
<td></td>
</tr>
<tr>
<td>writeString(String value)</td>
<td>Writes a String to the stream message.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.Message: 
- acknowledge, clearBody, clearProperties, getBooleanProperty, 
  getByteProperty, getDoubleProperty, getFloatProperty, 
  getIntProperty, getJMSCorrelationID, getJMSCorrelationIDAsBytes, 
  getJMSDeliveryMode, getJMSDestination, getJMSExpiration, 
  getJMSMessageID, getJMSPriority, getJMSRedelivered, getJMSReplyTo, 
  getJMSTimestamp, getJMSType, getLongProperty, getJsonObjectProperty, 
  getPropertyNames, getShortProperty, getStringProperty, 
  propertyExists, setBooleanProperty, setByteProperty, 
  setDoubleProperty, setFloatProperty, setIntProperty, 
  setJMSCorrelationID, setJMSCorrelationIDAsBytes,
Method Detail

public boolean readBoolean() throws JMSException

    boolean return

    Throws JMSException: JMS
    Throws MessageEOFException:
    Throws MessageFormatException:
    Throws MessageNotReadableException:

readBoolean

boolean readBoolean() throws JMSException

    Reads a boolean from the stream message.

Returns:
    the boolean value read

Throws:
    JMSException - if the JMS provider fails to read the message due to some internal error.
    MessageEOFException - if unexpected end of message stream has been reached.
    MessageFormatException - if this type conversion is invalid.
    MessageNotReadableException - if the message is in write-only mode.

public byte readByte() throws JMSException
byte

return 8

Throws JMSException: JMS

Throws MessageEOFException:

Throws MessageFormatException:

Throws MessageNotReadableException:

readByte

byte readByte()

throws JMSException

Reads a byte value from the stream message.

Returns:
the next byte from the stream message as a 8-bit byte

Throws:
JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of message stream has been reached.
MessageFormatException - if this type conversion is invalid.
MessageNotReadableException - if the message is in write-only mode.

-----------

public short readShort() throws JMSException

16

return 16

Throws JMSException: JMS

Throws MessageEOFException:

Throws MessageFormatException:

Throws MessageNotReadableException:
short readShort() throws JMSException

Reads a 16-bit integer from the stream message.

Returns:  
a 16-bit integer from the stream message

Throws:  
  JMSException - if the JMS provider fails to read the message due to some internal error.
  MessageEOFException - if unexpected end of message stream has been reached.
  MessageFormatException - if this type conversion is invalid.
  MessageNotReadableException - if the message is in write-only mode.

public char readChar() throws JMSException

Unicode

<table>
<thead>
<tr>
<th>return</th>
<th>Unicode</th>
</tr>
</thead>
</table>

Throws  
  JMSException: JMS
  MessageEOFException:  
  MessageFormatException:  
  MessageNotReadableException: 

readChar

char readChar() throws JMSException

Reads a Unicode character value from the stream message.

Returns:  
a Unicode character from the stream message

Throws:  
  JMSException - if the JMS provider fails to read the message due to some internal error.
  MessageEOFException - if unexpected end of message stream
public int readInt() throws JMSException

32

return 32

int

Throws: JMSException: JMS

Throws: MessageEOFException:

Throws: MessageFormatException:

Throws: MessageNotReadableException:

readInt

int readInt() throws JMSException

Reads a 32-bit integer from the stream message.

Returns:

a 32-bit integer value from the stream message, interpreted as an int

Throws:

JMSException - if the JMS provider fails to read the message due to some internal error.

MessageEOFException - if unexpected end of message stream has been reached.

MessageFormatException - if this type conversion is invalid.

MessageNotReadableException - if the message is in write-only mode.

public long readLong() throws JMSException

64

return 64

long
Throws JMSException: JMS
Throws MessageEOFException:
Throws MessageFormatException:
Throws MessageNotReadableException:

**readLong**

long readLong() throws JMSException

Reads a 64-bit integer from the stream message.

**Returns:**
- a 64-bit integer value from the stream message, interpreted as a long

**Throws:**
- JMSException - if the JMS provider fails to read the message due to some internal error.
- MessageEOFException - if unexpected end of message stream has been reached.
- MessageFormatException - if this type conversion is invalid.
- MessageNotReadableException - if the message is in write-only mode.

**public float readFloat() throws JMSException**

float return

float

**Throws**
- JMSException: JMS
- MessageEOFException:
- MessageFormatException:
- MessageNotReadableException:

**readFloat**

float readFloat()
public double readDouble() throws JMSException

double return double

Throws

JMSException: JMS
MessageEOFException:
MessageFormatException:
MessageNotReadableException:

readDouble

double readDouble() throws JMSException

Reads a double from the stream message.

Returns:

a double value from the stream message

Throws:

JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of message stream has been reached.
MessageFormatException - if this type conversion is invalid.
MessageNotReadableException - if the message is in write-only mode.
public String readString() throws JMSException

String return Unicode

Throws JMSException: JMS
Throws MessageEOFException:
Throws MessageFormatException:
Throws MessageNotReadableException:

readString

String readString() throws JMSException

Reads a String from the stream message.

Returns: a Unicode string from the stream message

Throws: JMSException - if the JMS provider fails to read the message due to some internal error.
MessageEOFException - if unexpected end of message stream has been reached.
MessageFormatException - if this type conversion is invalid.
MessageNotReadableException - if the message is in write-only mode.

public int readBytes(byte[] value) throws JMSException

byte readBytes
readBytes readBytes -1

byte null readBytes -1

byte readBytes 0

byte[] readBytes

MessageFormatException

byte byte[] readObject

return byte -1

Throws JMSException: JMS

Throws MessageEOFException:

Throws MessageFormatException:

Throws MessageNotReadableException:

See also readObject()
If the byte array field value is empty, `readBytes` returns 0.

Once the first `readBytes` call on a `byte[]` field value has been made, the full value of the field must be read before it is valid to read the next field. An attempt to read the next field before that has been done will throw a `MessageFormatException`.

To read the byte field value into a new `byte[]` object, use the `readObject` method.

**Parameters:**
- `value` - the buffer into which the data is read

**Returns:**
the total number of bytes read into the buffer, or -1 if there is no more data because the end of the byte field has been reached

**Throws:**
- `JMSException` - if the JMS provider fails to read the message due to some internal error.
- `MessageEOFException` - if unexpected end of message stream has been reached.
- `MessageFormatException` - if this type conversion is invalid.
- `MessageNotReadableException` - if the message is in write-only mode.

**See Also:**
- `readObject()`

---

```java
public Object readObject() throws JMSException
```

```
Java “Java ”
writeObject  writetype

byte    byte[]    Byte[]
byte    readObject byte    byte[]
MessageFormatException
return    Java
```

```java
int    Intege```
Throws  

**JMSException**: JMS

Throws  

**MessageEOFException**: 

Throws  

**MessageFormatException**: 

Throws  

**MessageNotReadableException**: 

See also  

readBytes(byte[] value)

---

**readObject**

```
Object readObject()  
throws JMSException
```

Reads an object from the stream message.

This method can be used to return, in objectified format, an object in the Java programming language ("Java object") that has been written to the stream with the equivalent `writeObject` method call, or its equivalent primitive `write` method.

Note that byte values are returned as `byte[]`, not `Byte[]`.

An attempt to call `readObject` to read a byte field value into a new `byte[]` object before the full value of the byte field has been read will throw a `MessageFormatException`.

**Returns:**

a Java object from the stream message, in objectified format (for example, if the object was written as an `int`, an `Integer` is returned)

**Throws:**

- **JMSException** - if the JMS provider fails to read the message due to some internal error.
- **MessageEOFException** - if unexpected end of message stream has been reached.
- **MessageFormatException** - if this type conversion is invalid.
- **MessageNotReadableException** - if the message is in write-only mode.

**See Also:**
public void writeBoolean(boolean value) throws JMSException

boolean true (byte)1 false (byte)0

value boolean

Throws JMSException: JMS

Throws MessageNotWriteableException:

writeBoolean

void writeBoolean(boolean value)

throws JMSException

Writes a boolean to the stream message. The value true is written as the value (byte)1; the value false is written as the value (byte)0.

Parameters:
value - the boolean value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeByte(byte value) throws JMSException

byte

value byte

Throws JMSException: JMS

Throws MessageNotWriteableException:

writeByte
void writeByte(byte value) throws JMSException

Writes a byte to the stream message.

Parameters:
value - the byte value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeShort(short value) throws JMSException

short
value
short

Throws
JMSException: JMS
MessageNotWriteableException:

writeShort

void writeShort(short value) throws JMSException

Writes a short to the stream message.

Parameters:
value - the short value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public void writeChar(char value) throws JMSException

char
```java
void writeChar(char value)
  throws JMSException

Writes a char to the stream message.

Parameters:
  value - the char value to be written

Throws:
  JMSException - if the JMS provider fails to write the message due to some internal error.
  MessageNotWriteableException - if the message is in read-only mode.
```

```java
public void writeInt(int value) throws JMSException
int

void writeInt(int value)
  throws JMSException

Writes an int to the stream message.

Parameters:
  value - the int value to be written

Throws:
```

JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

```java
public void writeLong(long value) throws JMSException

void writeLong(long value)
    throws JMSException

writeLong

Parameters:
value - the long value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.
```

```java
public void writeFloat(float value) throws JMSException

void writeFloat(float value)
    throws JMSException

writeFloat
```
void `writeFloat`(float value) throws `JMSException`

Writes a float to the stream message.

**Parameters:**
value - the float value to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```
public void writeDouble(double value) throws `JMSException`

double
value double

throws `JMSException`: JMS
Throws `MessageNotWriteableException`:
```

writeDouble

void `writeDouble`(double value) throws `JMSException`

Writes a double to the stream message.

**Parameters:**
value - the double value to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```
public void writeString(String value) throws `JMSException`
```

```
public void writeBytes(byte[] value) throws JMSException
byte

void writeBytes(byte[] value)
throws JMSException

void writeString(String value)
throws JMSException

writeString

throws JMSException

void writeString(String value)
throws JMSException

public void writeBytes(byte[] value) throws JMSException

Write a byte array field to the stream message.

void writeBytes(byte[] value)
throws JMSException

Write a String to the stream message.

Parameters:
value - the String value to be written

Throws:
JMSException - if the JMS provider fails to write the message due to some internal error.
messageNotWriteableException - if the message is in read-only mode.

Throws:
JMSException: JMS

Throws:
MessageNotWriteableException: JMS
The byte array `value` is written to the message as a byte array field. Consecutively written byte array fields are treated as two distinct fields when the fields are read.

**Parameters:**
- `value` - the byte array value to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```java
public void writeBytes(byte[] value, int offset, int length)
    throws JMSException
```

Writes a portion of a byte array as a byte array field to the stream message.

The a portion of the byte array `value` is written to the message as a byte array field. Consecutively written byte array fields are treated as
two distinct fields when the fields are read.

**Parameters:**
- value - the byte array value to be written
- offset - the initial offset within the byte array
- length - the number of bytes to use

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due to some internal error.
- `MessageNotWriteableException` - if the message is in read-only mode.

```java
public void writeObject(Object value) throws JMSException
```

<table>
<thead>
<tr>
<th>Integer</th>
<th>Double</th>
<th>Long</th>
<th>String</th>
<th>byte</th>
</tr>
</thead>
</table>

Java

**Throws**
- `JMSException`: JMS
- `MessageFormatException`: 
- `MessageNotWriteableException`: 

**writeObject**

```java
void writeObject(Object value)
``` throws `JMSException`

Writes an object to the stream message.

This method works only for the objectified primitive object types (Integer, Double, Long ...), String objects, and byte arrays.

**Parameters:**
- value - the Java object to be written

**Throws:**
- `JMSException` - if the JMS provider fails to write the message due
to some internal error.

MessageFormatException - if the object is invalid.

MessageNotWriteableException - if the message is in read-only mode.

---

public void reset() throws JMSException

Throws JMSException: JMS

Throws MessageFormatException:

reset

void reset() throws JMSException

Puts the message body in read-only mode and repositions the stream to the beginning.

Throws:

JMSException - if the JMS provider fails to reset the message due to some internal error.

MessageFormatException - if the message has an invalid format.
javax.xml.stream.util  **Class StreamReaderDelegate**

java.lang.Object  
  javax.xml.stream.util.StreamReaderDelegate

**All Implemented Interfaces:**
  XMLStreamConstants, XMLStreamReader

```
public class StreamReaderDelegate
extends Object
implements XMLStreamReader

Implements: XMLStreamReader
```

XMLStreamReader  
XMLStreamReader

version  
1.0

See  
javax.xml.stream.XMLStreamReader,
also  
javax.xml.stream.util.EventReaderDelegate

This is the base class for deriving an XMLStreamReader filter This class is designed to sit between an XMLStreamReader and an application's XMLStreamReader. By default each method does nothing but call the corresponding method on the parent interface.

**Version:**  
1.0

**Author:**  
Copyright (c) 2003 by BEA Systems. All Rights Reserved.

**See Also:**
  XMLHttpRequest, EventReaderDelegate

---

**Field Summary**
### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>StreamReaderDelegate</strong>()</td>
<td>Construct an empty filter with no parent.</td>
</tr>
<tr>
<td><strong>StreamReaderDelegate</strong>(XMLStreamReader reader)</td>
<td>Construct an filter with the specified parent.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>close</strong>()</td>
<td>Frees any resources associated with this Reader.</td>
</tr>
<tr>
<td><strong>getAttributeCount</strong>()</td>
<td>Returns the count of attributes on this START_ELEMENT, this method is only valid on a START_ELEMENT or ATTRIBUTE.</td>
</tr>
<tr>
<td><strong>getAttributeLocalName</strong>(int index)</td>
<td>Returns the localName of the attribute at the provided index</td>
</tr>
<tr>
<td><strong>getAttributeName</strong>(int index)</td>
<td>Returns the qname of the attribute at the provided index</td>
</tr>
<tr>
<td><strong>getAttributeNamespace</strong>(int index)</td>
<td>Returns the namespace of the attribute at the provided index</td>
</tr>
<tr>
<td><strong>getAttributePrefix</strong>(int index)</td>
<td>Returns the prefix of this attribute at the provided index</td>
</tr>
<tr>
<td><strong>getAttributeType</strong>(int index)</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>String</code> Returns the XML type of the attribute at the provided index.</td>
<td></td>
</tr>
<tr>
<td><code>String </code> <strong>getAttributeValue</strong>(int index)</td>
<td>Returns the value of the attribute at the index.</td>
</tr>
<tr>
<td><code>String </code> <strong>getAttributeValue</strong>(String namespaceUri, String localName)</td>
<td>Returns the normalized attribute value of the attribute with the namespace and localName. If the namespaceURI is null the namespace is not checked for equality.</td>
</tr>
<tr>
<td><code>String </code> <strong>getCharacterEncodingScheme</strong>()</td>
<td>Returns the character encoding declared on the xml declaration. Returns null if none was declared.</td>
</tr>
<tr>
<td><code>String </code> <strong>getElementText</strong>()</td>
<td>Reads the content of a text-only element, an exception is thrown if this is not a text-only element.</td>
</tr>
<tr>
<td><code>String </code> <strong>getEncoding</strong>()</td>
<td>Return input encoding if known or null if unknown.</td>
</tr>
<tr>
<td><code>int </code> <strong>getEventType</strong>()</td>
<td>Returns an integer code that indicates the type of the event the cursor is pointing to.</td>
</tr>
<tr>
<td><code>String </code> <strong>getLocalName</strong>()</td>
<td>Returns the (local) name of the current event.</td>
</tr>
<tr>
<td><code>Location</code> <strong>getLocation</strong>()</td>
<td>Return the current location of the processor.</td>
</tr>
<tr>
<td><code>QName</code> <strong>getName</strong>()</td>
<td>Returns a QName for the current START_ELEMENT or END_ELEMENT event.</td>
</tr>
<tr>
<td><code>NamespaceContext</code> <strong>getNamespaceContext</strong>()</td>
<td>Returns a read only namespace context for the current position.</td>
</tr>
<tr>
<td><code>int</code> <strong>getNamespaceCount</strong>()</td>
<td>Returns the count of namespaces declared on this START_ELEMENT or END_ELEMENT, this method is only valid on a START_ELEMENT, END_ELEMENT or NAMESPACE.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>getNamespacePrefix(int index)</code></td>
<td>Returns the prefix for the namespace declared at the index.</td>
</tr>
<tr>
<td><code>getNamespaceURI()</code></td>
<td>If the current event is a START_ELEMENT or END_ELEMENT this method returns the URI of the prefix or the default namespace.</td>
</tr>
<tr>
<td><code>getNamespaceURI(int index)</code></td>
<td>Returns the uri for the namespace declared at the index.</td>
</tr>
<tr>
<td><code>getNamespaceURI(String prefix)</code></td>
<td>Return the uri for the given prefix.</td>
</tr>
<tr>
<td><code>getParent()</code></td>
<td>Get the parent of this instance.</td>
</tr>
<tr>
<td><code>getPIData()</code></td>
<td>Get the data section of a processing instruction</td>
</tr>
<tr>
<td><code>getPITarget()</code></td>
<td>Get the target of a processing instruction</td>
</tr>
<tr>
<td><code>getPrefix()</code></td>
<td>Returns the prefix of the current event or null if the event does not have a prefix.</td>
</tr>
<tr>
<td><code>getProperty(String name)</code></td>
<td>Get the value of a feature/property from the underlying implementation.</td>
</tr>
<tr>
<td><code>getText()</code></td>
<td>Returns the current value of the parse event as a string, this returns the string value of a CHARACTERS event, returns the value of a COMMENT, the replacement value for an ENTITY_REFERENCE, the string value of a CDATA section, the string value for a SPACE event, or the String value of the internal subset of the DTD.</td>
</tr>
<tr>
<td><code>getTextCharacters()</code></td>
<td>Returns an array which contains the characters from this event.</td>
</tr>
<tr>
<td><code>getTextCharacters(int sourceStart, char[] target, int targetStart, int length)</code></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** The above table contains the methods and their descriptions extracted from the text.
### getTextLength() Method
- **Description:**
  - Gets the length of the sequence of characters for this Text event within the text character array.
- **Returns:**
  - An `int` representing the length of the sequence of characters.

### getTextStart() Method
- **Description:**
  - Returns the offset into the text character array where the first character (of this text event) is stored.
- **Returns:**
  - An `int` representing the offset into the text character array.

### getVersion() Method
- **Description:**
  - Gets the XML version declared on the XML declaration.
  - Returns `null` if none was declared.
- **Returns:**
  - A `String` representing the XML version or `null`.

### hasName() Method
- **Description:**
  - Returns true if the current event has a name (is a START_ELEMENT or END_ELEMENT); otherwise, false.
- **Returns:**
  - A `boolean` indicating whether the current event has a name.

### hasNext() Method
- **Description:**
  - Returns true if there are more parsing events and false if there are no more events.
- **Returns:**
  - A `boolean` indicating whether there are more parsing events.

### hasText() Method
- **Description:**
  - Return true if the current event has text, false otherwise. The following events have text: CHARACTERS, DTD, ENTITY_REFERENCE, COMMENT, SPACE.
- **Returns:**
  - A `boolean` indicating whether the current event has text.

### isAttributeSpecified(int index) Method
- **Description:**
  - Returns a boolean which indicates if this attribute was created by default.
- **Parameters:**
  - `index` (int)
- **Returns:**
  - A `boolean` indicating if the attribute was created by default.

### isCharacters() Method
- **Description:**
  - Returns true if the cursor points to a character data event.
- **Returns:**
  - A `boolean` indicating whether the cursor points to a character data event.

### isEndElement() Method
- **Description:**
  - Returns true if the cursor points to an end tag (otherwise false).
- **Returns:**
  - A `boolean` indicating whether the cursor points to an end tag.

### isStandalone() Method
- **Description:**
  - Gets the standalone declaration from the XML declaration.
- **Returns:**
  - A `boolean` indicating whether the XML declaration is standalone.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean <code>isStartElement()</code></td>
<td>Returns true if the cursor points to a start tag (otherwise false)</td>
</tr>
<tr>
<td>boolean <code>isWhiteSpace()</code></td>
<td>Returns true if the cursor points to a character data event that consists of all whitespace</td>
</tr>
<tr>
<td>int <code>next()</code></td>
<td>Get next parsing event - a processor may return all contiguous character data in a single chunk, or it may split it into several chunks.</td>
</tr>
<tr>
<td>int <code>nextTag()</code></td>
<td>Skips any white space (isWhiteSpace() returns true), COMMENT, or PROCESSING_INSTRUCTION, until a START_ELEMENT or END_ELEMENT is reached.</td>
</tr>
<tr>
<td>void <code>require(int type, String namespaceURI, String localName)</code></td>
<td>Test if the current event is of the given type and if the namespace and name match the current namespace and name of the current event.</td>
</tr>
<tr>
<td>void <code>setParent(XMLStreamReader reader)</code></td>
<td>Set the parent of this instance.</td>
</tr>
<tr>
<td>boolean <code>standaloneSet()</code></td>
<td>Checks if standalone was set in the document</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

**Constructor Detail**

public `StreamReaderDelegate()`
StreamReaderDelegate

public StreamReaderDelegate()

Construct an empty filter with no parent.

public StreamReaderDelegate(XMLStreamReader reader)

reader

StreamReaderDelegate

public StreamReaderDelegate(XMLStreamReader reader)

Construct an filter with the specified parent.

Parameters:
reader - the parent

Method Detail

public void setParent(XMLStreamReader reader)

reader

setParent

public void setParent(XMLStreamReader reader)

Set the parent of this instance.

Parameters:
reader - the new parent
public **XMLElement** getParent()

    return null

getParent

public **XMLElement** getParent()

    Get the parent of this instance.

    **Returns:**
    the parent or null if none is set

public int next() throws **XMLStreamException**

next

public int next()
    throws **XMLStreamException**

    **Description copied from interface:** **XMLElement**
    Get next parsing event - a processor may return all contiguous
    character data in a single chunk, or it may split it into several chunks.
    If the property javax.xml.stream.isCoalescing is set to true element
    content must be coalesced and only one CHARACTERS event must
    be returned for contiguous element content or CDATA Sections. By
    default entity references must be expanded and reported
    transparently to the application. An exception will be thrown if an
    entity reference cannot be expanded. If element content is empty
    (i.e. content is "") then no CHARACTERS event will be reported.

    Given the following XML:
    <foo><!--description-->content text<!
    [CDATA[<greeting>Hello</greeting>]]>other content</foo>
    The behavior of calling next() when being on foo will be:
1- the comment (COMMENT)  
2- then the characters section (CHARACTERS)  
3- then the CDATA section (another CHARACTERS)  
4- then the next characters section (another CHARACTERS)  
5- then the END_ELEMENT  

**NOTE:** empty element (such as <tag/>) will be reported with two separate events: START_ELEMENT, END_ELEMENT - This preserves parsing equivalency of empty element to <tag></tag>. This method will throw an IllegalStateException if it is called after hasNext() returns false.  

**Specified by:**  
next in interface `XMLStreamReader`  

**Returns:**  
the integer code corresponding to the current parse event  

**Throws:**  
`XMLStreamException` - if there is an error processing the underlying XML source  

**See Also:**  
`XMLEvent`  

```java  
public int nextTag() throws XMLStreamException  
```

**nextTag**

```java  
public int nextTag()  
throws XMLStreamException  
```

**Description copied from interface:** `XMLStreamReader`  
Skips any white space (isWhiteSpace() returns true), COMMENT, or PROCESSING_INSTRUCTION, until a START_ELEMENT or END_ELEMENT is reached. If other than white space characters, COMMENT, PROCESSING_INSTRUCTION, START_ELEMENT, END_ELEMENT are encountered, an exception is thrown. This method should be used when processing element-only content seperated by white space.
Precondition: none
Postcondition: the current event is START_ELEMENT or END_ELEMENT and cursor may have moved over any whitespace event.
Essentially it does the following (implementations are free to optimized but must do equivalent processing):

```java
int eventType = next();
while((eventType == XMLStreamConstants.CHARACTERS && isWhiteSpace()) // skip whitespace
    || (eventType == XMLStreamConstants.CDATA && isWhiteSpace()) // skip whitespace
    || eventType == XMLStreamConstants.SPACE
    || eventType == XMLStreamConstants.PROCESSING_INSTRUCTION
    || eventType == XMLStreamConstants.COMMENT)
{
    eventType = next();
}
if (eventType != XMLStreamConstants.START_ELEMENT && eventType != XMLStreamConstants.END_ELEMENT)
    throw new String XMLStreamException("expected start or end tag"
} return eventType;
```

**Specified by:**
nextTag in interface XMLStreamReader

**Returns:**
the event type of the element read (START_ELEMENT or END_ELEMENT)

**Throws:**
XMLStreamException - if the current event is not white space, PROCESSING_INSTRUCTION, START_ELEMENT or END_ELEMENT

```
public String getElementText() throws XMLStreamException

getElementText
```

```
public String getElementText() throws XMLStreamException
```
Description copied from interface: `XMLStreamReader`

Reads the content of a text-only element, an exception is thrown if this is not a text-only element. Regardless of value of javax.xml.stream.isCoalescing this method always returns coalesced content.

Precondition: the current event is START_ELEMENT.

Postcondition: the current event is the corresponding END_ELEMENT.

The method does the following (implementations are free to optimized but must do equivalent processing):

```java
if(getEventType() != XMLStreamConstants.START_ELEMENT) {
    throw new XMLStreamException("parser must be on START_ELEMENT to read next text", getLocation());
}
int eventType = next();
StringBuffer content = new StringBuffer();
while(eventType != XMLStreamConstants.END_ELEMENT ) {
    if(eventType == XMLStreamConstants.CHARACTERS
        || eventType == XMLStreamConstants.CDATA
        || eventType == XMLStreamConstants.SPACE
        || eventType == XMLStreamConstants.ENTITY_REFERENCE) {
        buf.append(getText());
    } else if(eventType == XMLStreamConstants.PROCESSING_INSTRUCTION
        || eventType == XMLStreamConstants.COMMENT) {
        // skipping
    } else if(eventType == XMLStreamConstants.END_DOCUMENT) {
        throw new XMLStreamException("unexpected end of document when reading element text content",
            getLocation());
    } else if(eventType == XMLStreamConstants.START_ELEMENT) {
        throw new XMLStreamException("element text content may not contain START_ELEMENT",
            getLocation());
    } else {
        throw new XMLStreamException("Unexpected event type "+eventType, getLocation());
    }
    eventType = next();
} return buf.toString();
```

Specified by:
`getElementText` in interface `XMLStreamReader`

Throws:
`XMLStreamException` - if the current event is not a
START_ELEMENT or if a non text element is encountered

```java
public void require(int type, String namespaceURI, String localName) throws XMLStreamException

Description copied from interface: XMLStreamReader
Test if the current event is of the given type and if the namespace and name match the current namespace and name of the current event. If the namespaceURI is null it is not checked for equality, if the localName is null it is not checked for equality.

Specified by:
require in interface XMLStreamReader
Parameters:
type - the event type
namespaceURI - the uri of the event, may be null
localName - the localName of the event, may be null
Throws:
XMLStreamException - if the required values are not matched.
```

```java
public boolean hasNext() throws XMLStreamException

Description copied from interface: XMLStreamReader
```
Returns true if there are more parsing events and false if there are no more events. This method will return false if the current state of the XMLStreamReader is END_DOCUMENT

**Specified by:**
`hasNext` in interface `XMLStreamReader`

**Returns:**
true if there are more events, false otherwise

**Throws:**
`XMLStreamException` - if there is a fatal error detecting the next state

---

**public void close() throws `XMLStreamException`**

**close**

`public void close()`  
`throws XMLStreamException`

**Description copied from interface: `XMLStreamReader`**
Frees any resources associated with this Reader. This method does not close the underlying input source.

**Specified by:**
`close` in interface `XMLStreamReader`

**Throws:**
`XMLStreamException` - if there are errors freeing associated resources

---

**public String getNamespaceURI(String prefix)**

**getNamespaceURI**

`public String getNamespaceURI(String prefix)`
Description copied from interface: **XMLStreamReader**
Return the uri for the given prefix. The uri returned depends on the current state of the processor.

**NOTE:** The 'xml' prefix is bound as defined in [Namespaces in XML](http://www.w3.org/XML/1998/namespace) specification to "http://www.w3.org/XML/1998/namespace".

**NOTE:** The 'xmlns' prefix must be resolved to following namespace http://www.w3.org/2000/xmlns/

Specified by: 
  [getNamespaceURI](http://www.w3.org/XML/1998/namespace) in interface **XMLStreamReader**

**Parameters:**
 prefix - The prefix to lookup, may not be null

**Returns:**
 the uri bound to the given prefix or null if it is not bound

### public javax.xml.namespace.NamespaceContext getNamespaceContext()

**getNamespaceContext**

**Description copied from interface: **XMLStreamReader**

Returns a read only namespace context for the current position. The context is transient and only valid until a call to next() changes the state of the reader.

**Specified by:**
  [getNamespaceContext](http://www.w3.org/XML/1998/namespace) in interface **XMLStreamReader**

**Returns:**
 return a namespace context

### public boolean isStartElement()
isStartElement

public boolean isStartElement()

Description copied from interface: XMLStreamReader
Returns true if the cursor points to a start tag (otherwise false)

Specified by:
    isStartElement in interface XMLStreamReader
Returns:
    true if the cursor points to a start tag, false otherwise

isEndElement

public boolean isEndElement()

isCharacters

public boolean isCharacters()

Description copied from interface: XMLStreamReader
Returns true if the cursor points to an end tag (otherwise false)

Specified by:
    isEndElement in interface XMLStreamReader
Returns:
    true if the cursor points to an end tag, false otherwise
Returns true if the cursor points to a character data event

Specified by:
    isCharacters in interface XMLStreamReader

Returns:
    true if the cursor points to character data, false otherwise

public boolean isWhiteSpace()

isWhiteSpace

public boolean isWhiteSpace()

    Description copied from interface: XMLStreamReader
    Returns true if the cursor points to a character data event that consists of all whitespace

    Specified by:
        isWhiteSpace in interface XMLStreamReader

    Returns:
        true if the cursor points to all whitespace, false otherwise

public String getAttributeValue(String namespaceUri,
                                  String localName)

getAttributeValue

public String getAttributeValue(String namespaceUri,
                                  String localName)

    Description copied from interface: XMLStreamReader
    Returns the normalized attribute value of the attribute with the namespace and localName If the namespaceURI is null the namespace is not checked for equality
Specified by: 
`getAttributeValue` in interface `XMLStreamReader` 

Parameters:
- namespaceUri - the namespace of the attribute
- localName - the local name of the attribute, cannot be null

Returns:
returns the value of the attribute, returns null if not found

---

```java
public int getAttributeCount()
```

**getAttributeCount**

```java
public int getAttributeCount()
```

**Description copied from interface: XMLStreamReader**
Returns the count of attributes on this START_ELEMENT, this method is only valid on a START_ELEMENT or ATTRIBUTE. This count excludes namespace definitions. Attribute indices are zero-based.

**Specified by:**
`getAttributeCount` in interface `XMLStreamReader`

**Returns:**
returns the number of attributes

---

```java
public javax.xml.namespace.QName getAttributeName(int index)
```

**getAttributeName**

```java
public QName getAttributeName(int index)
```

**Description copied from interface: XMLStreamReader**
Returns the qname of the attribute at the provided index
**public String getAttributePrefix(int index)**

**getAttributePrefix**

```java
public String getAttributePrefix(int index)
```

**Description copied from interface: XMLStreamReader**

Returns the prefix of this attribute at the provided index.

**Specified by:**

```java
getAttributePrefix in interface XMLStreamReader
```

**Parameters:**

- `index` - the position of the attribute

**Returns:**

- the prefix of the attribute

---

**public String getAttributeNamespace(int index)**

**getAttributeNamespace**

```java
public String getAttributeNamespace(int index)
```

**Description copied from interface: XMLStreamReader**

Returns the namespace of the attribute at the provided index.

**Specified by:**

```java
getAttributeNamespace in interface XMLStreamReader
```

**Parameters:**

- `index` - the position of the attribute

**Returns:**

- the namespace of the attribute
index - the position of the attribute

**Returns:**
the namespace URI (can be null)

---

**public String getAttributeLocalName(int index)**

**getAttributeLocalName**

**public String getAttributeLocalName(int index)**

**Description copied from interface:** [XMLStreamReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/jaxen/flushable/impl/DefaultXMLStreamReader.html)
Returns the localName of the attribute at the provided index

**Specified by:**  
getAttributeLocalName in interface [XMLStreamReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/jaxen/flushable/impl/DefaultXMLStreamReader.html)

**Parameters:**
- **index** - the position of the attribute

**Returns:**  
the localName of the attribute

---

**public String getAttributeType(int index)**

**getAttributeType**

**public String getAttributeType(int index)**

**Description copied from interface:** [XMLStreamReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/jaxen/flushable/impl/DefaultXMLStreamReader.html)
Returns the XML type of the attribute at the provided index

**Specified by:**  
getAttributeType in interface [XMLStreamReader](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/jaxen/flushable/impl/DefaultXMLStreamReader.html)

**Parameters:**
- **index** - the position of the attribute

**Returns:**  
the XML type of the attribute
public String getAttributeValue(int index)

getAttributeValue

public String getAttributeValue(int index)

Description copied from interface: XMLStreamReader
Returns the value of the attribute at the index

Specified by: getAttributeValue in interface XMLStreamReader

Parameters:
  index - the position of the attribute

Returns:
  the attribute value

public boolean isAttributeSpecified(int index)

isAttributeSpecified

public boolean isAttributeSpecified(int index)

Description copied from interface: XMLStreamReader
Returns a boolean which indicates if this attribute was created by default

Specified by: isAttributeSpecified in interface XMLStreamReader

Parameters:
  index - the position of the attribute

Returns:
  true if this is a default attribute
public int getNamespaceCount()

getNamespaceCount

public int getNamespaceCount()

Description copied from interface: XMLStreamReader
Returns the count of namespaces declared on this
START_ELEMENT or END_ELEMENT, this method is only valid on
a START_ELEMENT, END_ELEMENT or NAMESPACE. On an
END_ELEMENT the count is of the namespaces that are about to go
out of scope. This is the equivalent of the information reported by
SAX callback for an end element event.

Specified by:
    getNamespaceCount in interface XMLStreamReader

Returns:
    returns the number of namespace declarations on this specific
element

public String getNamespacePrefix(int index)

getNamespacePrefix

public String getNamespacePrefix(int index)

Description copied from interface: XMLStreamReader
Returns the prefix for the namespace declared at the index. Returns
null if this is the default namespace declaration

Specified by:
    getNamespacePrefix in interface XMLStreamReader

Parameters:
    index - the position of the namespace declaration

Returns:
    returns the namespace prefix
public String getNamespaceURI(int index)

getNamespaceURI

public String getNamespaceURI(int index)

Description copied from interface: XMLStreamReader
Returns the uri for the namespace declared at the index.

Specified by:
getNamespaceURI in interface XMLStreamReader

Parameters:
index - the position of the namespace declaration

Returns:
returns the namespace uri

public int getEventType()

getEventType

public int getEventType()

Description copied from interface: XMLStreamReader
Returns an integer code that indicates the type of the event the cursor is pointing to.

Specified by:
getEventType in interface XMLStreamReader

public String getText()

getText
public String getText()

Description copied from interface: XMLStreamReader
Returns the current value of the parse event as a string, this returns the string value of a CHARACTERS event, returns the value of a COMMENT, the replacement value for an ENTITYREFERENCE, the string value of a CDATA section, the string value for a SPACE event, or the String value of the internal subset of the DTD. If an ENTITYREFERENCE has been resolved, any character data will be reported as CHARACTERS events.

Specified by:
   getText in interface XMLStreamReader
Returns:
   the current text or null

public int getTextCharacters(int sourceStart, char[] target, int targetStart, int length) throws XMLStreamException

ggetTextCharacters

Description copied from interface: XMLStreamReader
Gets the the text associated with a CHARACTERS, SPACE or CDATA event. Text starting a "sourceStart" is copied into "target" starting at "targetStart". Up to "length" characters are copied. The number of characters actually copied is returned. The "sourceStart" argument must be greater or equal to 0 and less than or equal to the number of characters associated with the event. Usually, one requests text starting at a "sourceStart" of 0. If the number of characters actually copied is less than the "length", then there is no more text. Otherwise, subsequent calls need to be made until all text has been retrieved. For example: int length = 1024; char[]
myBuffer = new char[ length ]; for ( int sourceStart = 0 ; ;
sourceStart += length ) { int nCopied =
stream.getTextCharacters( sourceStart, myBuffer, 0, length );
if ( nCopied < length ) break; } XMLStreamException may be
thrown if there are any XML errors in the underlying source. The
"targetStart" argument must be greater than or equal to 0 and less
than the length of "target", Length must be greater than 0 and
"targetStart + length" must be less than or equal to length of "target".

Specified by:
   getTextCharacters in interface XMLStreamReader
Parameters:
   sourceStart - the index of the first character in the source array
to copy
target - the destination array
targetStart - the start offset in the target array
length - the number of characters to copy
Returns:
   the number of characters actually copied
Throws:
   XMLStreamException - if the underlying XML source is not well-
formed

public char[] getTextCharacters()
**getTextCharacters** in interface [XMLStreamReader](https://www.w3.org/TR/xml-stream/#get-text-characters)

**Returns:**
the current text or an empty array

---

**public int getTextStart()**

ggetTextStart

**public int getTextStart()**

Description copied from interface: [XMLStreamReader](https://www.w3.org/TR/xml-stream/#get-text-start)

Returns the offset into the text character array where the first character (of this text event) is stored.

**Specified by:**
ggetTextStart in interface [XMLStreamReader](https://www.w3.org/TR/xml-stream/#get-text-start)

---

**public int getTextLength()**

ggetTextLength

**public int getTextLength()**

Description copied from interface: [XMLStreamReader](https://www.w3.org/TR/xml-stream/#get-text-length)

Returns the length of the sequence of characters for this Text event within the text character array.

**Specified by:**
ggetTextLength in interface [XMLStreamReader](https://www.w3.org/TR/xml-stream/#get-text-length)

---

**public String getEncoding()**
getEncoding

public String getEncoding()

Description copied from interface: XMLStreamReader
Return input encoding if known or null if unknown.

Specified by: getEncoding in interface XMLStreamReader

Returns:
the encoding of this instance or null

---------------------

public boolean hasText()

hasText

public boolean hasText()

Description copied from interface: XMLStreamReader
Return true if the current event has text, false otherwise The following events have text: CHARACTERS,DTD ,ENTITY_REFERENCE, COMMENT, SPACE

Specified by: hasText in interface XMLStreamReader

---------------------

public Location getLocation()

getLocation

public Location getLocation()

Description copied from interface: XMLStreamReader
Return the current location of the processor. If the Location is unknown the processor should return an implementation of Location
that returns -1 for the location and null for the publicId and systemId. The location information is only valid until next() is called.

**Specified by:**
*getLocation* in interface *XMLStreamReader*

---

**public javax.xml.namespace.QName getName()**

*getName*

**public QName getName()**

*Description copied from interface: XMLStreamReader*
Returns a QName for the current START_ELEMENT or END_ELEMENT event

**Specified by:**
*getName* in interface *XMLStreamReader*

**Returns:**
the QName for the current START_ELEMENT or END_ELEMENT event

---

**public String getLocalName()**

*getLocalName*

**public String getLocalName()**

*Description copied from interface: XMLStreamReader*
Returns the (local) name of the current event. For START_ELEMENT or END_ELEMENT returns the (local) name of the current element. For ENTITY_REFERENCE it returns entity name. The current event must be START_ELEMENT or END_ELEMENT, or ENTITY_REFERENCE
Specified by:
   `getLocalName` in interface `XMLStreamReader`  

Returns:
   the localName

---

**public boolean hasName()**

hasName

public boolean hasName()

Description copied from interface: `XMLStreamReader`
returns true if the current event has a name (is a START_ELEMENT or END_ELEMENT) returns false otherwise

Specified by:
   `hasName` in interface `XMLStreamReader`  

---

**public String getNamespaceURI()**

getNamespaceURI

public `String` getNamespaceURI()

Description copied from interface: `XMLStreamReader`
If the current event is a START_ELEMENT or END_ELEMENT this method returns the URI of the prefix or the default namespace. Returns null if the event does not have a prefix.

Specified by:
   `getNamespaceURI` in interface `XMLStreamReader`  

Returns:
   the URI bound to this elements prefix, the default namespace, or null
public String getPrefix()

getPrefix

public String getPrefix()

    Description copied from interface: XMLStreamReader
    Returns the prefix of the current event or null if the event does not have a prefix

    Specified by: getPrefix in interface XMLStreamReader
    Returns: the prefix or null

public String getVersion()

getVersion

public String getVersion()

    Description copied from interface: XMLStreamReader
    Get the xml version declared on the xml declaration Returns null if none was declared

    Specified by: getVersion in interface XMLStreamReader
    Returns: the XML version or null

public boolean isStandalone()

isStandalone
public boolean isStandalone()

Description copied from interface: XMLStreamReader
Get the standalone declaration from the xml declaration

Specified by:
   isStandalone in interface XMLStreamReader

Returns:
   true if this is standalone, or false otherwise

public boolean standaloneSet()

standaloneSet

public boolean standaloneSet()

Description copied from interface: XMLStreamReader
Checks if standalone was set in the document

Specified by:
   standaloneSet in interface XMLStreamReader

Returns:
   true if standalone was set in the document, or false otherwise

public String getCharacterEncodingScheme()

getCharacterEncodingScheme

public String getCharacterEncodingScheme()

Description copied from interface: XMLStreamReader
Returns the character encoding declared on the xml declaration
Returns null if none was declared

Specified by:
getCharacterEncodingScheme in interface XMLStreamReader
Returns:
the encoding declared in the document or null

public String getPITarget()

g getPITarget

public String getPITarget()

Description copied from interface: XMLStreamReader
Get the target of a processing instruction

Specified by:
getPITarget in interface XMLStreamReader
Returns:
the target or null

public String getPIData()

g getPIData

public String getPIData()

Description copied from interface: XMLStreamReader
Get the data section of a processing instruction

Specified by:
getPIData in interface XMLStreamReader
Returns:
the data or null

public Object getProperty(String name)
getProperty

public Object getProperty(String name)

Description copied from interface: XMLStreamReader
Get the value of a feature/property from the underlying implementation

Specified by: getProperties in interface XMLStreamReader

Parameters:
  name - The name of the property, may not be null

Returns:
  The value of the property
javax.xml.rpc.holders  Class StringHolder

java.lang.Object  
   javax.xml.rpc.holders.StringHolder

All Implemented Interfaces:
   Holder

public final class StringHolder
   extends Object
   implements Holder

Implements: Holder

Field Summary

| String value |

Constructor Summary

| StringHolder() |

| StringHolder(String myString) |

Method Summary

Methods inherited from class java.lang.Object
   clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Field Detail

value

public String value

Constructor Detail

public StringHolder()

StringHolder

public StringHolder()

public StringHolder(String myString)

StringHolder

public StringHolder(String myString)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.mail.search  Class StringTerm

java.lang.Object  
  javax.mail.search.SearchTerm  
    javax.mail.search.StringTerm

All Implemented Interfaces:
  Serializable

Direct Known Subclasses:
  AddressStringTerm, BodyTerm, HeaderTerm, MessageIDTerm, SubjectTerm

public abstract class StringTerm
extends SearchTerm

Extends: SearchTerm
Extended by: AddressStringTerm, BodyTerm, HeaderTerm, MessageIDTerm, SubjectTerm

String  strcmp ......

This class implements the match method for Strings. The current implementation provides only for substring matching. We could add comparisons (like strcmp ...).

Author:
  Bill Shannon, John Mani
See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>protected  boolean  ignoreCase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ignore case when comparing?</td>
</tr>
</tbody>
</table>
**protected String**

The pattern.

## Constructor Summary

| protected | StringTerm(String pattern) |
| protected | StringTerm(String pattern, boolean ignoreCase) |

## Method Summary

| boolean | equals(Object obj) |
| boolean | getIgnoreCase() |
| String | getPattern() |
| int | hashCode() |
| protected boolean | match(String s) |

Equality comparison.

Return true if we should ignore case when matching.

Return the string to match with.

Compute a hashCode for this object.

## Field Detail
pattern

protected String pattern

    The pattern.

ignoreCase

protected boolean ignoreCase

    Ignore case when comparing?

### Constructor Detail

protected StringTerm(String pattern)

**StringTerm**

protected StringTerm(String pattern)

protected StringTerm(String pattern, boolean ignoreCase)

**StringTerm**

protected StringTerm(String pattern, boolean ignoreCase)

### Method Detail

public String getPattern()
**getPattern**

```java
public String getPattern()
```

Return the string to match with.

**getIgnoreCase**

```java
public boolean getIgnoreCase()
```

```java
true
```

**getIgnoreCase**

```java
public boolean getIgnoreCase()
```

Return true if we should ignore case when matching.

**match**

```java
protected boolean match(String s)
```

**equals**

```java
public boolean equals(Object obj)
```

Equality comparison.
Overrides:

equals in class Object

public int hashCode()
hashCode

hashCode

public int hashCode()

Compute a hashCode for this object.

Overrides:

hashCode in class Object
javax.xml.rpc Interface Stub

public interface Stub

javax.xml.rpc.Stub

Stub

javax.xml.rpc.Stub

version 0.1

See also javax.xml.rpc.Service

The interface javax.xml.rpc.Stub is the common base interface for the stub classes. All generated stub classes are required to implement the javax.xml.rpc.Stub interface. An instance of a stub class represents a client side proxy or stub instance for the target service endpoint.

The javax.xml.rpc.Stub interface provides an extensible property mechanism for the dynamic configuration of a stub instance.

Version:

0.1

Author:

Rahul Sharma

See Also:

Service

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>ENDPOINT_ADDRESS_PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard property: Target service endpoint address.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>static String</th>
<th>PASSWORD_PROPERTY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Standard property: Password for authentication.</td>
</tr>
</tbody>
</table>
static String SESSION_MAINTAIN_PROPERTY
    Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint.

static String USERNAME_PROPERTY
    Standard property: User name for authentication.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Object _getProperty(String name)</td>
<td></td>
<td>Gets the value of a specific configuration property.</td>
</tr>
<tr>
<td>Iterator _getPropertyNames()</td>
<td></td>
<td>Returns an Iterator view of the names of the properties that can be configured on this stub instance.</td>
</tr>
<tr>
<td>void _setProperty(String name, Object value)</td>
<td></td>
<td>Sets the name and value of a configuration property for this Stub instance.</td>
</tr>
</tbody>
</table>

### Field Detail

USERNAME_PROPERTY

static final String USERNAME_PROPERTY

    Standard property: User name for authentication.

    Type: java.lang.String

**See Also:**

Constant Field Values
PASSWORD_PROPERTY

static final String PASSWORD_PROPERTY

Standard property: Password for authentication.

Type: java.lang.String

See Also: Constant Field Values

ENDPOINT_ADDRESS_PROPERTY

static final String ENDPOINT_ADDRESS_PROPERTY

Standard property: Target service endpoint address. The URI scheme for the endpoint address specification must correspond to the protocol/transport binding for this stub class.

Type: java.lang.String

See Also: Constant Field Values

SESSION_MAINTAIN_PROPERTY

static final String SESSION_MAINTAIN_PROPERTY

Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint. If this property is set to true, the service client indicates that it wants the session to be maintained. If set to false, the session is not maintained. The default value for this property is false.
Type: java.lang.Boolean

See Also:
Constant Field Values

Method Detail

public void _setProperty(String name, Object value)
Stub  Stub

_setProperty

name
value

_setProperty Stub

JAXRPCException:

Throws

- Stub

- 

_setProperty

void _setProperty(String name, Object value)

Sets the name and value of a configuration property for this Stub instance. If the Stub instances contains a value of the same property, the old value is replaced.

Note that the _setProperty method may not perform validity check on a configured property value. An example is the standard property for the target service endpoint address that is not checked for validity in the _setProperty method. In this case, stub configuration errors are detected at the remote method invocation.

Parameters:
name - Name of the configuration property
value - Value of the property

Throws:

JAXRPCException -
  • If an optional standard property name is specified, however this Stub implementation class does not support the configuration of this property.
  • If an invalid or unsupported property name is specified or if a value of mismatched property type is passed.
  • If there is any error in the configuration of a valid property.

public Object _getProperty(String name)

name
return
Throws JAXRPCException:

_getProperty

Object _getProperty(String name)

Gets the value of a specific configuration property.

Parameters:
  name - Name of the property whose value is to be retrieved

Returns:
  Value of the configuration property

Throws:
  JAXRPCException - if an invalid or unsupported property name is passed.

public java.util.Iterator<E> _getPropertyNames()

Stub

return java.lang.String Iterator
_getPropertyNames

*Iterator _getPropertyNames()*

Returns an *Iterator* view of the names of the properties that can be configured on this stub instance.

**Returns:**
Iterator for the property names of the type `java.lang.String`

---

**Overview**

**Package**

**Tree**

** Deprecated**

** Index**

** Help**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

---

PS:
javax.mail.search  Class SubjectTerm

java.lang.Object  ↓  javax.mail.search.SearchTerm  ↓  javax.mail.search.StringTerm  ↓  javax.mail.search.SubjectTerm

All Implemented Interfaces:
  Serializable

public final class SubjectTerm
extends StringTerm

Extends: searchTerm  >  StringTerm

Message Subject

This class implements comparisons for the Message Subject header. The comparison is case-insensitive.

Author:
  Bill Shannon, John Mani

See Also:
  Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.mail.search.StringTerm</th>
</tr>
</thead>
<tbody>
<tr>
<td>ignoreCase, pattern</td>
</tr>
</tbody>
</table>

Constructor Summary

| SubjectTerm(String pattern) |
### Constructor Detail

**public SubjectTerm(String pattern)**

*pattern*

### SubjectTerm

**public SubjectTerm(String pattern)**

Constructor.

**Parameters:**

*pattern* - the pattern to search for

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object obj)</td>
<td>Equality comparison.</td>
</tr>
<tr>
<td>boolean match(Message msg)</td>
<td>The match method.</td>
</tr>
</tbody>
</table>

### Methods inherited from class javax.mail.search.StringTerm

getIgnoreCase, getPattern, hashCode, match

### Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, toString, wait, wait
public boolean match(Message msg)

    msg       Message
    return    true false

match

public boolean match(Message msg)

    The match method.

    Specified by:
        match in class SearchTerm

    Parameters:
        msg - the pattern match is applied to this Message's subject header

    Returns:
        true if the pattern match succeeds, otherwise false

public boolean equals(Object obj)

equals

public boolean equals(Object obj)

    Equality comparison.

    Overrides:
        equals in class StringTerm
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.transaction Interface Synchronization

public interface Synchronization

registerSynchronization Transaction Synchronization

The transaction manager supports a synchronization mechanism that allows the interested party to be notified before and after the transaction completes. Using the registerSynchronization method, the application server registers a Synchronization object for the transaction currently associated with the target Transaction object.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void afterCompletion(int status)</td>
<td>This method is called by the transaction manager after the transaction is committed or rolled back.</td>
</tr>
<tr>
<td>void beforeCompletion()</td>
<td>The beforeCompletion method is called by the transaction manager prior to the start of the two-phase transaction commit process.</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
class Synchronization {
    public void beforeCompletion() {
        // Method implementation
    }
}```
void beforeCompletion()

The beforeCompletion method is called by the transaction manager prior to the start of the two-phase transaction commit process. This call is executed with the transaction context of the transaction that is being committed.

public void afterCompletion(int status)

status

afterCompletion

void afterCompletion(int status)

This method is called by the transaction manager after the transaction is committed or rolled back.

Parameters:
status - The status of the transaction completion.
javax.transaction  **Class SystemException**

**java.lang.Object**
- **java.lang.Throwable**
  - **java.lang.Exception**
    - **javax.transaction.SystemException**

**All Implemented Interfaces:**
- **Serializable**

```java
public class SystemException
    extends Exception

Extends: Throwable > Exception

SystemException
```

The SystemException is thrown by the transaction manager to indicate that it has encountered an unexpected error condition that prevents future transaction services from proceeding.

**See Also:**
- **Serialized Form**

---

**Field Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>errorCode</td>
<td>The error code with which to create the SystemException.</td>
</tr>
</tbody>
</table>

---

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>SystemException()</td>
</tr>
<tr>
<td>SystemException(int errorCode)</td>
</tr>
</tbody>
</table>
Create a **SystemException** with a given error code.

<table>
<thead>
<tr>
<th>SystemException(String s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create a SystemException with a given string.</td>
</tr>
</tbody>
</table>

## Method Summary

Methods inherited from class java.lang.**Throwable**

- `fillInStackTrace`
- `getCause`
- `getLocaleMessage`
- `getMessage`
- `getStackTrace`
- `initCause`
- `printStackTrace`
- `printStackTrace`
- `printStackTrace`
- `setStackTrace`
- `toString`

Methods inherited from class java.lang.**Object**

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `wait`
- `wait`
- `wait`

## Field Detail

**errorCode**

```java
class SystemException {
    public int errorCode;
    // The error code with which to create the SystemException.
}
```

## Constructor Detail

**public SystemException()**

**SystemException**

```java
class SystemException {
    public SystemException() {
        // Constructor to create a SystemException.
    }
}
```
public SystemException(String s)

XAException

s

SystemException

public SystemException(String s)

Create a SystemException with a given string.

Parameters:

s - The string message for the exception

public SystemException(int errcode)

SystemException

ercode

SystemException

public SystemException(int errcode)

Create a SystemException with a given error code.

Parameters:

errcode - The error code for the exception
PS:
javax.persistence  **Annotation Type Table**

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
public @interface Table

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)

<table>
<thead>
<tr>
<th>Implementations</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>SecondaryTable</td>
<td>SecondaryTables</td>
</tr>
</tbody>
</table>

Table

@Target
@Retention
public class Customer { ...

<table>
<thead>
<tr>
<th>since</th>
<th>Java Persistence 1.0</th>
<th>en</th>
</tr>
</thead>
</table>

This annotation specifies the primary table for the annotated entity. Additional tables may be specified using **SecondaryTable** or **SecondaryTables** annotation.

If no **Table** annotation is specified for an entity class, the default values apply.

Example:

```java
@Entity
@Table(name="CUST", schema="RECORDS")
public class Customer { ...
```

**Since:**  
Java Persistence 1.0
Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>catalog</td>
<td>(Optional) The catalog of the table.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>(Optional) The name of the table.</td>
</tr>
<tr>
<td>String</td>
<td>schema</td>
<td>(Optional) The schema of the table.</td>
</tr>
<tr>
<td>UniqueConstraint[]</td>
<td>uniqueConstraints</td>
<td>(Optional) Unique constraints that are to be placed on the table.</td>
</tr>
</tbody>
</table>

abstract public String name()

name

public abstract String name

(Optional) The name of the table.

Defaults to the entity name.

Default: ""

abstract public String catalog()
**catalog**

public abstract String catalog

(Optional) The catalog of the table.

Defaults to the default catalog.

**Default:**

```
""
```

---

**abstract public String schema()**

---

**schema**

public abstract String schema

(Optional) The schema of the table.

Defaults to the default schema for user.

**Default:**

```
""
```

---

**abstract public UniqueConstraint[] uniqueConstraints()**
public abstract UniqueConstraint[] uniqueConstraints

(Optional) Unique constraints that are to be placed on the table. These are only used if table generation is in effect. These constraints apply in addition to any constraints specified by the Column andJoinColumn annotations and constraints entailed by primary key mappings.

Defaults to no additional constraints.

Default:

{}
javax.persistence  **Annotation Type TableGenerator**

```java
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface TableGenerator

**Implements:** Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)

**GeneratedValue**

1

@Entity public class Employee {
  ...
  @TableGenerator(
      name="empGen",
      table="ID_GEN",
      pkColumnName="GEN_KEY",
      valueColumnName="GEN_VALUE",
      pkColumnValue="EMP_ID",
      allocationSize=1)
  @Id
  @GeneratedValue(strategy=TABLE, generator="empGen")
  public int id;
  ...
}

2

@Entity public class Address {
  ...
  @TableGenerator(
      name="addressGen",
      table="ID_GEN",
      pkColumnName="GEN_KEY",
      valueColumnName="GEN_VALUE",
      pkColumnValue="ADDR_ID")
  @Id
  @GeneratedValue(strategy=TABLE, generator="addressGen")
  public int id;
  ...
}
```
This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the `GeneratedValue` annotation. A table generator may be specified on the entity class or on the primary key field or property. The scope of the generator name is global to the persistence unit (across all generator types).

Example 1:

```java
@Entity public class Employee {
    ...
    @TableGenerator(
        name="empGen",
        table="ID_GEN",
        pkColumnName="GEN_KEY",
        valueColumnName="GEN_VALUE",
        pkColumnValue="EMP_ID",
        allocationSize=1)
    @Id
    @GeneratedValue(strategy=TABLE, generator="empGen")
    public int id;
    ...
}
```

Example 2:

```java
@Entity public class Address {
    ...
    @TableGenerator(
        name="addressGen",
        table="ID_GEN",
        pkColumnName="GEN_KEY",
        valueColumnName="GEN_VALUE",
        pkColumnValue="ADDR_ID")
    @Id
    @GeneratedValue(strategy=TABLE, generator="addressGen")
    public int id;
    ...
}
```

Since:
Java Persistence 1.0
## Required Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>String</td>
<td>(Required) A unique generator name that can be referenced by one or more classes to be the generator for id values.</td>
</tr>
</tbody>
</table>

## Optional Element Summary

<table>
<thead>
<tr>
<th>Element</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>allocationSize</td>
<td>int</td>
<td>(Optional) The amount to increment by when allocating id numbers from the generator.</td>
</tr>
<tr>
<td>catalog</td>
<td>String</td>
<td>(Optional) The catalog of the table.</td>
</tr>
<tr>
<td>initialValue</td>
<td>int</td>
<td>(Optional) The initial value to be used when allocating id numbers from the generator.</td>
</tr>
<tr>
<td>pkColumnName</td>
<td>String</td>
<td>(Optional) Name of the primary key column in the table.</td>
</tr>
<tr>
<td>pkColumnValue</td>
<td>String</td>
<td>(Optional) The primary key value in the generator table that distinguishes this set of generated values from others that may be stored in the table.</td>
</tr>
<tr>
<td>schema</td>
<td>String</td>
<td>(Optional) The schema of the table.</td>
</tr>
<tr>
<td>table</td>
<td>String</td>
<td>(Optional) Name of table that stores the generated id values.</td>
</tr>
<tr>
<td>uniqueConstraints</td>
<td>UniqueConstraint[]</td>
<td>(Optional) Unique constraints that are to be placed on the table.</td>
</tr>
<tr>
<td>valueColumnName</td>
<td>String</td>
<td>(Optional) Name of the column that stores the last value generated.</td>
</tr>
</tbody>
</table>
abstract public String name()

name

public abstract String name

(Required) A unique generator name that can be referenced by one or more classes to be the generator for id values.

abstract public String table()

ID

table

public abstract String table

(Optional) Name of table that stores the generated id values.

Defaults to a name chosen by persistence provider.

Default: 

abstract public String catalog()
**catalog**

public abstract String catalog

(Optional) The catalog of the table.

Defaults to the default catalog.

**Default:**

```
"
```

**abstract public String schema()**

**schema**

public abstract String schema

(Optional) The schema of the table.

Defaults to the default schema for user.

**Default:**

```
"
```

**abstract public String pkColumnName()**
pkColumn Name

public abstract String pkColumnName

(Optional) Name of the primary key column in the table.
Defaults to a provider-chosen name.

Default: 


valueColumn Name

public abstract String valueColumnName

(Optional) Name of the column that stores the last value generated.
Defaults to a provider-chosen name.

Default: 


pkColumn Value

abstract public String pkColumnValue()
**pkColumnValue**

```
public abstract String pkColumnValue
```

(Optional) The primary key value in the generator table that distinguishes this set of generated values from others that may be stored in the table.

Defaults to a provider-chosen value to store in the primary key column of the generator table

**Default:** ""

---

**abstract public int initialValue()**

**ID**

**initialValue**

```
public abstract int initialValue
```

(Optional) The initial value to be used when allocating id numbers from the generator.

**Default:** 0

---

**abstract public int allocationSize()**

**ID**

**allocationSize**
public abstract int allocationSize

(Optional) The amount to increment by when allocating id numbers from the generator.

Default:
50

abstract public UniqueConstraint[] uniqueConstraints()

uniqueConstraints

public abstract UniqueConstraint[] uniqueConstraints

(Optional) Unique constraints that are to be placed on the table. These are only used if table generation is in effect. These constraints apply in addition to primary key constraints.

Defaults to no additional constraints.

Default:
{}
PS:
javax.servlet.jsp.tagext Interface Tag

All Superinterfaces:
   JspTag

All Known Subinterfaces:
   BodyTag, IterationTag

All Known Implementing Classes:
   AttributeTag, BodyTagSupport, ConverterELTag, ConverterTag, FacetTag, TagAdapter, TagSupport, UIComponentBodyTag, UIComponentClassicTagBase, UIComponentELTag, UIComponentTag, ValidatorELTag, ValidatorTag

public interface Tag
extends JspTag

Implements: JspTag
Implemented by: IterationTag, TagAdapter, UIComponentELTag, UIComponentTag

Tag JSP

current JSP pageContext setter getter

JSP setPageContext setParent doStartTag() doEndTag()
The interface of a classic tag handler that does not want to manipulate its body. The Tag interface defines the basic protocol between a Tag handler and JSP page implementation class. It defines the life cycle and the methods to be invoked at start and end tag.

**Properties**

The Tag interface specifies the setter and getter methods for the core pageContext and parent properties.

The JSP page implementation object invokes setPageContext and setParent, in that order, before invoking doStartTag() or doEndTag().

**Methods**
There are two main actions: doStartTag and doEndTag. Once all appropriate properties have been initialized, the doStartTag and doEndTag methods can be invoked on the tag handler. Between these invocations, the tag handler is assumed to hold a state that must be preserved. After the doEndTag invocation, the tag handler is available for further invocations (and it is expected to have retained its properties).

**Lifecycle**

Lifecycle details are described by the transition diagram below, with the following comments:

- [1] This transition is intended to be for releasing long-term data. No guarantees are assumed on whether any properties have been retained or not.
- [2] This transition happens if and only if the tag ends normally without raising an exception.
- [3] Some setters may be called again before a tag handler is reused. For instance, setParent() is called if it's reused within the same page but at a different level, setPageContext() is called if it's used in another page, and attribute setters are called if the values differ or are expressed as request-time attribute values.
- Check the TryCatchFinally interface for additional details related to exception handling and resource management.

Once all invocations on the tag handler are completed, the release method is invoked on it. Once a release method is invoked all properties, including parent and pageContext, are assumed to have been reset to an unspecified value. The page compiler guarantees that release() will be invoked on the Tag handler before the handler is released to the GC.

**Empty and Non-Empty Action**

If the TagLibraryDescriptor file indicates that the action must always have an empty action, by an <body-content> entry of "empty", then the doStartTag() method must return SKIP_BODY.

Otherwise, the doStartTag() method may return SKIP_BODY or
If \texttt{SKIP\_BODY} is returned the body, if present, is not evaluated.

If \texttt{EVAL\_BODY\_INCLUDE} is returned, the body is evaluated and "passed through" to the current out.

---

### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>EVAL_BODY_INCLUDE</td>
<td>Evaluate body into existing out stream.</td>
</tr>
<tr>
<td>static int</td>
<td>EVAL_PAGE</td>
<td>Continue evaluating the page.</td>
</tr>
<tr>
<td>static int</td>
<td>SKIP_BODY</td>
<td>Skip body evaluation.</td>
</tr>
<tr>
<td>static int</td>
<td>SKIP_PAGE</td>
<td>Skip the rest of the page.</td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>doEndTag()</td>
<td>Process the end tag for this instance.</td>
</tr>
<tr>
<td>int</td>
<td>doStartTag()</td>
<td>Process the start tag for this instance.</td>
</tr>
<tr>
<td>Tag</td>
<td>getParent()</td>
<td>Get the parent (closest enclosing tag handler) for this tag handler.</td>
</tr>
<tr>
<td>void</td>
<td>release()</td>
<td>Called on a Tag handler to release state.</td>
</tr>
<tr>
<td>void</td>
<td>setPageContext(PageContext pc)</td>
<td>Set the current page context.</td>
</tr>
<tr>
<td>void</td>
<td>setParent(Tag t)</td>
<td>Set the parent (closest enclosing tag handler) of this tag handler.</td>
</tr>
</tbody>
</table>
Field Detail

SKIP_BODY

static final int SKIP_BODY

Skip body evaluation. Valid return value for doStartTag and doAfterBody.

See Also:
Constant Field Values

EVAL_BODY_INCLUDE

static final int EVAL_BODY_INCLUDE

Evaluate body into existing out stream. Valid return value for doStartTag.

See Also:
Constant Field Values

SKIP_PAGE

static final int SKIP_PAGE

Skip the rest of the page. Valid return value for doEndTag.

See Also:
Constant Field Values
static final int EVAL_PAGE

Continue evaluating the page. Valid return value for doEndTag().

See Also:
Constant Field Values

Method Detail

public void setPageContext(PageContext pc)

JSP  doStartTag()

doStartTag() ** doEndTag()

  pc

setPageContext

void setPageContext(PageContext pc)

  Set the current page context. This method is invoked by the JSP
  page implementation object prior to doStartTag().

  This value is *not* reset by doEndTag() and must be explicitly reset
  by a page implementation if it changes between calls to
  doStartTag().

  Parameters:
    pc - The page context for this tag handler.

public void setParent(Tag t)
JSP  doStartTag()

** doEndTag()

\[ t \quad \text{null} \]

setParent

```java
void setParent(Tag t)
```

Set the parent (closest enclosing tag handler) of this tag handler. Invoked by the JSP page implementation object prior to doStartTag().

This value is *not* reset by doEndTag() and must be explicitly reset by a page implementation.

**Parameters:**
- t - The parent tag, or null.

```java
public Tag getParent()
```

getParent() TagSupport findAncestorWithClass()

observable observable void JSP

\[ \text{return null} \]

See also findAncestorWithClass

getParent
getParent()

Get the parent (closest enclosing tag handler) for this tag handler.

The getParent() method can be used to navigate the nested tag handler structure at runtime for cooperation among custom actions; for example, the findAncestorWithClass() method in TagSupport provides a convenient way of doing this.

The current version of the specification only provides one formal way of indicating the observable type of a tag handler: its tag handler implementation class, described in the tag-class subelement of the tag element. This is extended in an informal manner by allowing the tag library author to indicate in the description subelement an observable type. The type should be a subtype of the tag handler implementation class or void. This additional constraint can be exploited by a specialized container that knows about that specific tag library, as in the case of the JSP standard tag library.

Returns:

the current parent, or null if none.

See Also:

TagSupport.findAncestorWithClass(javax.servlet.jsp.tagext.Tag, java.lang.Class)

public int doStartTag() throws JspException

JSP
doStartTag  pageContext

Tag.EVAL_BODY_INCLUDE
BodyTag.EVAL_BODY_BUFFERED SKIP_BODY

Tag EVAL_BODY_INCLUDE "out"
JspWriter doEndTag()
doStartTag()  JSP  AT-BEGIN  NESTED
TagExtraInfo  TLD  BodyTag  BodyTag
doStartTag()  BodyTag.EVAL_BODY_BUFFERED

return  EVAL_BODY_INCLUDE  SKIP_BODY
Throws  JspException:
See also  javax.servlet.jsp.tagext.BodyTag

doStartTag

int  doStartTag()
    throws  JspException

Process the start tag for this instance. This method is invoked by the
JSP page implementation object.

The doStartTag method assumes that the properties pageContext
and parent have been set. It also assumes that any properties
exposed as attributes have been set too. When this method is
invoked, the body has not yet been evaluated.

This method returns Tag.EVAL_BODY_INCLUDE or
BodyTag.EVAL_BODY_BUFFERED to indicate that the body of the
action should be evaluated or SKIP_BODY to indicate otherwise.

When a Tag returns EVAL_BODY_INCLUDE the result of evaluating
the body (if any) is included into the current "out" JspWriter as it
happens and then doEndTag() is invoked.

BodyTag.EVAL_BODY_BUFFERED is only valid if the tag handler
implements BodyTag.

The JSP container will resynchronize the values of any AT_BEGIN
and NESTED variables (defined by the associated TagExtraInfo or
TLD) after the invocation of doStartTag(), except for a tag handler
implementing BodyTag whose doStartTag() method returns
BodyTag.EVAL_BODY_BUFFERED.
Returns:
  EVAL_BODY_INCLUDE if the tag wants to process body,
  SKIP_BODY if it does not want to process it.

Throws:
  JspException - if an error occurred while processing this tag

See Also:
  BodyTag

---

public int doEndTag() throws JspException

JSP
doStartTag doStartTag

EVAL_PAGE SKIP_PAGE doEndTag()
Servlet
doEndTag() JSP AT_BEGIN AT_END
TagExtraInfo TLD

  return

  JspException: Threws

---

doEndTag

int doEndTag() throws JspException

Process the end tag for this instance. This method is invoked by the
JSP page implementation object on all Tag handlers.

This method will be called after returning from doStartTag. The body
of the action may or may not have been evaluated, depending on the
return value of doStartTag.

If this method returns EVAL_PAGE, the rest of the page continues to
be evaluated. If this method returns SKIP_PAGE, the rest of the
page is not evaluated, the request is completed, and the doEndTag() methods of enclosing tags are not invoked. If this request was forwarded or included from another page (or Servlet), only the current page evaluation is stopped.

The JSP container will resynchronize the values of any AT_BEGIN and AT_END variables (defined by the associated TagExtraInfo or TLD) after the invocation of doEndTag().

**Returns:**

Indication of whether to continue evaluating the JSP page.

**Throws:**

*JspException* - if an error occurred while processing this tag

```java
public void release()
JSP   doStartTag   doEndTag
```

**release**

```java
void release()
```

Called on a Tag handler to release state. The page compiler guarantees that JSP page implementation objects will invoke this method on all tag handlers, but there may be multiple invocations on doStartTag and doEndTag in between.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.servlet.jsp.tagext  **Class TagAdapter**

java.lang.Object

java.servlet.jsp.tagext.TagAdapter

**All Implemented Interfaces:**

  JspTag, Tag

---

public class TagAdapter

extends Object

implements Tag

**Implements:** Tag

  SimpleTag Tag  SimpleTag

  SimpleTag  Tag  Tag.setParent()  Tag  Tag
  SimpleTag  TagAdapter  SimpleTag  setParent()
  getAdaptee()  SimpleTag

since 2.0

Wraps any SimpleTag and exposes it using a Tag interface. This is used to allow collaboration between classic Tag handlers and SimpleTag handlers.

Because SimpleTag does not extend Tag, and because Tag.setParent() only accepts a Tag instance, a classic tag handler (one that implements Tag) cannot have a SimpleTag as its parent. To remedy this, a TagAdapter is created to wrap the SimpleTag parent, and the adapter is passed to setParent() instead. A classic Tag Handler can call getAdaptee() to retrieve the encapsulated SimpleTag instance.

**Since:**

  JSP 2.0
### Field Summary

Fields inherited from interface `javax.servlet.jsp.tagext.Tag`

- `EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

### Constructor Summary

**TagAdapter**(SimpleTag adaptee)

Creates a new TagAdapter that wraps the given SimpleTag and returns the parent tag when `getParent()` is called.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>doEndTag()</code></td>
<td>Must not be called.</td>
</tr>
<tr>
<td><code>doStartTag()</code></td>
<td>Must not be called.</td>
</tr>
<tr>
<td><code>getAdaptee()</code></td>
<td>Gets the tag that is being adapted to the Tag interface.</td>
</tr>
<tr>
<td><code>getParent()</code></td>
<td>Returns the parent of this tag, which is always <code>getAdaptee().getParent()</code>.</td>
</tr>
<tr>
<td><code>release()</code></td>
<td>Must not be called.</td>
</tr>
<tr>
<td><code>setPageContext</code>(PageContext pc)</td>
<td>Must not be called.</td>
</tr>
<tr>
<td><code>setParent</code>(Tag parentTag)</td>
<td>Must not be called.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public TagAdapter(SimpleTag adaptee)
SimpleTag TagAdapter getParent()

Creates a new TagAdapter that wraps the given SimpleTag and returns the parent tag when getParent() is called.

Parameters:
adaptee - The SimpleTag being adapted as a Tag.

public void setPageContext(PageContext pc)

pc

Throws UnsupportedOperationException:

setPageContext

public void setPageContext(PageContext pc)

Must not be called.

Specified by:
setPageContext in interface Tag

Parameters:
pc - ignored.

Throws:
public void setParent(Tag parentTag)
getAdaptee().getParent()

parentTag

Throws

UnsupportedOperationException:

setParent

public void setParent(Tag parentTag)

Must not be called. The parent of this tag is always
getAdaptee().getParent().

Specified by:

setParent in interface Tag

Parameters:

parentTag - ignored.

Throws:

UnsupportedOperationException - Must not be called.

public Tag getParent()

getAdaptee().getParent()

getAdaptee().getParent() Tag

getAdaptee().getParent() Tag

return

getParent

public Tag getParent()

Returns the parent of this tag, which is always
getAdaptee().getParent(). This will either be the enclosi
getAdaptee().getParent() implements Tag), or an adapter to the enclosing Tag (if getAdaptee().getParent() does not implement Tag).

**Specified by:**
- `getParent` in interface `Tag`

**Returns:**
The parent of the tag being adapted.

**See Also:**
- `TagSupport.findAncestorWithClass(javax.servlet.jsp.tagext.Tag, java.lang.Class)`

```
public JspTag getAdaptee()
Tag JSP 2.0 SimpleTag
return
```

**getAdaptee**

```
public JspTag getAdaptee()

Gets the tag that is being adapted to the Tag interface. This should be an instance of SimpleTag in JSP 2.0, but room is left for other kinds of tags in future spec versions.

**Returns:**
the tag that is being adapted
```

```
public int doStartTag() throws JspException

return UnsupportedOperationException
Throws UnsupportedOperationException:
JspException:
```

**doStartTag**
public int doStartTag() throws JspException

Must not be called.

Specified by:
        doStartTag in interface Tag

Returns:
        always throws UnsupportedOperationException

Throws:
        UnsupportedOperationException - Must not be called
        JspException - never thrown

See Also:
         BodyTag

public int doEndTag() throws JspException

    return UnsupportedOperationException

    Throws
    UnsupportedOperationException:

        JspException:

   doEndTag
public void release()

Throws

UnsupportedOperationException:

release

public void release()

Must not be called.

Specified by:

release in interface Tag

Throws:

UnsupportedOperationException - Must not be called
javax.servlet.jsp.tagext  

Class TagAttributeInfo

java.lang.Object  
└ javax.servlet.jsp.tagext.TagAttributeInfo

public class TagAttributeInfo
extends Object

Tag (TLD)

SCHEMA

Information on the attributes of a Tag, available at translation time. This class is instantiated from the Tag Library Descriptor file (TLD).

Only the information needed to generate code is included here. Other information like SCHEMA for validation belongs elsewhere.

Note from the Expert Group:
This should have been designed as an interface. Every time we change the TLD, we need to add a new constructor to this class (not good). This class should only be instantiated by container implementations (not by JSP developers).

---

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&quot;id&quot; is wired in to be ID.</td>
</tr>
</tbody>
</table>

Constructor Summary

(TagAttributeInfo(String name, boolean required, String type, 
boolean reqTime))
Constructor for TagAttributeInfo.

```java
TagAttributeInfo(String name, boolean required, String type, boolean reqTime, boolean fragment)
```

JSP 2.0 Constructor for TagAttributeInfo.

```java
TagAttributeInfo(String name, boolean required, String type, boolean reqTime, boolean fragment, String description, boolean deferredValue, boolean deferredMethod, String expectedTypeName, String methodSignature)
```

JSP 2.1 Constructor for TagAttributeInfo.

Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>canBeRequestTime()</code></td>
<td>Whether this attribute has been specified in the TLD as rtexprvalue.</td>
</tr>
<tr>
<td><code>getDescription()</code></td>
<td>Gets the description string of this tag attribute.</td>
</tr>
<tr>
<td><code>getExpectedTypeName()</code></td>
<td>Returns the name of the expected type (as a String) of this deferred value attribute.</td>
</tr>
<tr>
<td><code>getIdAttribute(TagAttributeInfo[] a)</code></td>
<td>Convenience static method that goes through an array of TagAttributeInfo objects and looks for &quot;id&quot;.</td>
</tr>
<tr>
<td><code>getMethodSignature()</code></td>
<td>Returns the expected method signature of this deferred method attribute.</td>
</tr>
<tr>
<td><code>getName()</code></td>
<td>The name of this attribute.</td>
</tr>
<tr>
<td><code>getTypeName()</code></td>
<td>The type (as a String) of this attribute.</td>
</tr>
<tr>
<td><code>isDeferredMethod()</code></td>
<td>Returns true if this attribute is to be passed a MethodExpression so that expression evaluation can be deferred.</td>
</tr>
</tbody>
</table>
| `isDeferredValue()` | }
### Method Details

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isFragment()</td>
<td>Whether this attribute is of type JspFragment.</td>
</tr>
<tr>
<td>boolean isRequired()</td>
<td>Whether this attribute is required.</td>
</tr>
<tr>
<td>String toString()</td>
<td>Returns a String representation of this TagAttributeInfo, suitable for debugging purposes.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Field Detail

**ID**

```java
public static final String ID
```

"id" is wired in to be ID. There is no real benefit in having it be something else IDREFs are not handled any differently.

**See Also:**

- Constant Field Values

### Constructor Detail

```java
public TagAttributeInfo(String name, boolean required, String type, boolean reqTime)
```

TagAttributeInfo TLD (Tag Library Descriptor) JSP
**TagLibrary**

name
required
type
reqTime

**TagAttributeInfo**

```java
public TagAttributeInfo(String name,
    boolean required,
    String type,
    boolean reqTime)
```

Constructor for TagAttributeInfo. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor).

**Parameters:**
- **name** - The name of the attribute.
- **required** - If this attribute is required in tag instances.
- **type** - The name of the type of the attribute.
- **reqTime** - Whether this attribute holds a request-time Attribute.

```java
public TagAttributeInfo(String name, boolean required, String type, boolean reqTime, boolean fragment)
```

TagAttributeInfo JSP 2.0 TLD (Tag Library Descriptor) JSP TagLibrary

```java
name
required
type
reqTime
fragment JspFragment
since 2.0
```
TagAttributeInfo

```java
public TagAttributeInfo(String name,
                        boolean required,
                        String type,
                        boolean reqTime,
                        boolean fragment)
```

JSP 2.0 Constructor for TagAttributeInfo. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor).

**Parameters:**
- `name` - The name of the attribute.
- `required` - If this attribute is required in tag instances.
- `type` - The name of the type of the attribute.
- `reqTime` - Whether this attribute holds a request-time Attribute.
- `fragment` - Whether this attribute is of type JspFragment

**Since:**
JSP 2.0

---

TagAttributeInfo

```java
public TagAttributeInfo(String name, 
                        boolean required, 
                        String type, 
                        boolean reqTime, 
                        boolean fragment, 
                        String description, 
                        boolean deferredValue, 
                        boolean deferredMethod, 
                        String expectedTypeName, 
                        String methodSignature)
```

JSP 2.1 Constructor for TagAttributeInfo. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor).

**Parameters:**
name - The name of the attribute.
required - If this attribute is required in tag instances.
type - The name of the type of the attribute.
reqTime - Whether this attribute holds a request-time Attribute.
fragment - Whether this attribute is of type JspFragment
description - The description of the attribute.
deferredValue - Whether this attribute is a deferred value.
deferredMethod - Whether this attribute is a deferred method.
rtexpr or deferred value.
expectedTypeName - The name of the expected type of this
deferred value (or null if this is not a deferred value).
methodSignature - The expected method signature of this
deferred method (or null if this is not a deferred method).

Since:
JSP 2.1

### Method Detail

**public String getName()**

```
public String getName()
```

return

g getName

**public String getTypeName()**

```
public String getTypeName()
```

String

```
String
```
**getTypeName**

```java
public String getTypeName()
```

The type (as a String) of this attribute.

This method must return "javax.el.ValueExpression" if `isDeferredValue()` returns true and `canBeRequestTime()` returns false. It must return "javax.el.MethodExpression" if `isDeferredMethod()` returns true. It must return "java.lang.Object" if `isDeferredValue()` returns true and `canBeRequestTime()` returns true.

**Returns:**
the type of the attribute

---

**public boolean canBeRequestTime()**

```java
return false
```

**canBeRequestTime**

```java
public boolean canBeRequestTime()
```

Whether this attribute has been specified in the TLD as rtexprvalue. If true, this means the attribute can hold a request-time value.

**Returns:**
true if the attribute has been specified in the TLD as rtexprvalue

---

**public boolean isRequired()**

```java
return false
```

**isRequired**
public boolean **isRequired**()  

Whether this attribute is required.  

**Returns:**  

if the attribute is required.  

---

public static **TagAttributeInfo** **getIdAttribute**(TagAttributeInfo[] a)  

TagAttributeInfo "id"  

    a                              TagAttributeInfo  
    return                         "id"  TagAttributeInfo  

**getIdAttribute**

---

public static **TagAttributeInfo** **getIdAttribute**(TagAttributeInfo[] a)  

Convenience static method that goes through an array of  
TagAttributeInfo objects and looks for "id".  

**Parameters:**  

a - An array of TagAttributeInfo  

**Returns:**  

The TagAttributeInfo reference with name "id"

---

public boolean **isFragment**()  

JspFragment  

    return                        JspFragment  
    since                         2.0  

**isFragment**

---

public boolean **isFragment**()
Whether this attribute is of type JspFragment.

**Returns:**
if the attribute is of type JspFragment

**Since:**
JSP 2.0

---

getDescription

```java
public String getDescription()
```

Gets the description string of this tag attribute.

**Returns:**
the description string of this tag attribute

---

isDeferredValue

```java
public boolean isDeferredValue()
```

Returns true if this attribute is to be passed a ValueExpression so that expression evaluation can be deferred.

If this method returns true, then `getExpectedType()` must return "javax.el.ValueExpression".

The `getExpectedType()` method can be used to retrieve the expected type this value expression will be constructed with.

**Returns:**
true if this attribute accepts a deferred value; false otherwise.

**Since:**
JSP 2.1
isDeferredMethod
public boolean isDeferredMethod()

Returns true if this attribute is to be passed a MethodExpression so that expression evaluation can be deferred.

If this method returns true, then getTypeName() must return "javax.el.MethodExpression".

The getMethodSignature() method can be used to retrieve the expected method signature this method expression will be constructed with.

Returns:
true if this attribute accepts a deferred method; false otherwise.

Since:
JSP 2.1

getExpectedTypeName
public String getExpectedTypeName()

Returns the name of the expected type (as a String) of this deferred value attribute.

This method returns null if isDeferredValue() returns false.

Returns:
the name of the expected type

Since:
JSP 2.1
getMethodSignature

public String getMethodSignature()

Returns the expected method signature of this deferred method attribute.

This method returns null if isDeferredMethod() returns false.

Returns: the method signature

Since: JSP 2.1

public String toString()

TagAttributeInfo String

return TagAttributeInfo String

toString

public String toString()

Returns a String representation of this TagAttributeInfo, suitable for debugging purposes.

Overrides: toString in class Object

Returns: a String representation of this TagAttributeInfo
to license terms.

PS:
javax.servlet.jsp.tagext  **Class TagData**

java.lang.Object  
  - javax.servlet.jsp.tagext.TagData

**All Implemented Interfaces:**  
  Cloneable

```java
public class TagData

extends Object
implements Cloneable

Implements: Cloneable
```

The (translation-time only) attribute/value information for a tag instance.

TagData is only used as an argument to the isValid, validate, and getVariableInfo methods of TagExtraInfo, which are invoked at translation time.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
</table>
| static Object REQUEST_TIME_VALUE  
  Distinguished value for an attribute to indicate its value is a request-time expression (which is not yet available because TagData instances are used at translation-time). |

| Constructor Summary |
TagData(Hashtable<String, Object> attrs)
   Constructor for a TagData.

TagData(Object[][] atts)
   Constructor for TagData.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAttribute</strong></td>
<td>String attName</td>
<td>The value of the attribute.</td>
</tr>
<tr>
<td><strong>getAttributes</strong></td>
<td></td>
<td>Enumerates the attributes.</td>
</tr>
<tr>
<td><strong>getAttributeString</strong></td>
<td>String attName</td>
<td>Get the value for a given attribute.</td>
</tr>
<tr>
<td><strong>getId</strong></td>
<td></td>
<td>The value of the tag’s id attribute.</td>
</tr>
<tr>
<td><strong>setAttribute</strong></td>
<td>String attName, Object value</td>
<td>Set the value of an attribute.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Field Detail

REQUEST_TIME_VALUE

public static final Object REQUEST_TIME_VALUE

Distinguished value for an attribute to indicate its value is a request-time expression (which is not yet available because TagData instances are used at translation-time).
**Constructor Detail**

```java
public TagData(Object[][] atts)
TagData
```

```java
static final Object[][] att = {{"connection", "conn0"}, {"id", "id0"}};
static final TagData td = new TagData(att);
```

### REQUEST_TIME_VALUE String
- **atts**
- **null**

**TagData**

```java
public TagData(Object[][] atts)
```

Constructor for TagData.

A typical constructor may be

```java
static final Object[][] att = {{"connection", "conn0"}, {"id", "id0"}};
static final TagData td = new TagData(att);
```

All values must be Strings except for those holding the distinguished object REQUEST_TIME_VALUE.

**Parameters:**
- **atts** - the static attribute and values. May be null.

```java
public TagData(java.util.Hashtable<K, V> attrs)
TagData
```

```java
atts
```
TagData

public TagData(Hashtable<String, Object> attrs)

Constructor for a TagData. If you already have the attributes in a hashtable, use this constructor.

Parameters:
attrs - A hashtable to get the values from.

Method Detail

public String getId()

id

return id null

getId

public String getId()

The value of the tag's id attribute.

Returns:
the value of the tag's id attribute, or null if no such attribute was specified.

public Object getAttribute(String attName)

REQUEST_TIME_VALUE null

attName

return

getAttribute
public Object getAttribute(String attName)

The value of the attribute. If a static value is specified for an attribute that accepts a request-time attribute expression then that static value is returned, even if the value is provided in the body of a <jsp:attribute> action. The distinguished object REQUEST_TIME_VALUE is only returned if the value is specified as a request-time attribute expression or via the <jsp:attribute> action with a body that contains dynamic content (scriptlets, scripting expressions, EL expressions, standard actions, or custom actions). Returns null if the attribute is not set.

Parameters:
attName - the name of the attribute

Returns:
the attribute's value

public void setAttribute(String attName, Object value)

setAttribute

public void setAttribute(String attName, Object value)

Set the value of an attribute.

Parameters:
attName - the name of the attribute
value - the value.

public String getAttributeString(String attName)
public String getAttributeString(String attName)

Get the value for a given attribute.

Parameters:
attName - the name of the attribute

Returns:
the attribute value string

Throws:
ClassCastException - if attribute value is not a String

public java.utilEnumeration getAttributes()

return TagData

Enumerates the attributes.

Returns:
An enumeration of the attributes in a TagData
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Optional class provided by the tag library author to describe additional translation-time information not described in the TLD. The TagExtraInfo class is mentioned in the Tag Library Descriptor file (TLD).

This class can be used:

- to indicate that the tag defines scripting variables
- to perform translation-time validation of the tag attributes.

It is the responsibility of the JSP translator that the initial value to be returned by calls to getTagInfo() corresponds to a TagInfo object for the
tag being translated. If an explicit call to setTagInfo() is done, then the object passed will be returned in subsequent calls to getTagInfo().

The only way to affect the value returned by getTagInfo() is through a setTagInfo() call, and thus, TagExtraInfo.setTagInfo() is to be called by the JSP translator, with a TagInfo object that corresponds to the tag being translated. The call should happen before any invocation on validate() and before any invocation on getVariableInfo().

**NOTE:** It is a (translation time) error for a tag definition in a TLD with one or more variable subelements to have an associated TagExtraInfo implementation that returns a VariableInfo array with one or more elements from a call to getVariableInfo().

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TagExtraInfo()</strong></td>
</tr>
<tr>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TagInfo</strong></td>
<td><strong>getTagInfo()</strong> Get the TagInfo for this class.</td>
</tr>
<tr>
<td><strong>VariableInfo[]</strong></td>
<td><strong>getVariableInfo</strong>(TagData data) information on scripting variables defined by the tag associated with this TagExtraInfo instance.</td>
</tr>
<tr>
<td><strong>isValid</strong>(TagData data)</td>
<td>Translation-time validation of the attributes.</td>
</tr>
<tr>
<td><strong>void</strong></td>
<td><strong>setTagInfo</strong>(TagInfo tagInfo) Set the TagInfo for this class.</td>
</tr>
<tr>
<td><strong>ValidationMessage[]</strong></td>
<td><strong>validate</strong>(TagData data) Translation-time validation of the attributes.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
**Constructor Detail**

```java
public TagExtraInfo()
```

**TagExtraInfo**

```java
public TagExtraInfo()
```

Sole constructor. (For invocation by subclass constructors, typically implicit.)

**Method Detail**

```java
public VariableInfo[] getVariableInfo(TagData data)
```

**getVariableInfo**

```java
public VariableInfo[] getVariableInfo(TagData data)
```

information on scripting variables defined by the tag associated with this TagExtraInfo instance. Request-time attributes are indicated as such in the TagData parameter.

**Parameters:**

data - The TagData instance.

**Returns:**
An array of VariableInfo data, or null or a zero length array if no scripting variables are to be defined.
public boolean isValid(TagData data)
TagData validate()
data	TagData
return
See also
isValid

isValid

public boolean isValid(TagData data)
Translation-time validation of the attributes. Request-time attributes are indicated as such in the TagData parameter. Note that the preferred way to do validation is with the validate() method, since it can return more detailed information.

Parameters:
    data - The TagData instance.

Returns:
    Whether this tag instance is valid.

See Also:
    validate(javax.servlet.jsp.tagext.TagData)

public ValidationMessage[] validate(TagData data)
TagData isValid()

JSP 2.0 validate() isValid() isValid() isValid()
false isValid() false ValidationMessage[]
public ValidationMessage[] validate(TagData data)

Translation-time validation of the attributes. Request-time attributes are indicated as such in the TagData parameter. Because of the higher quality validation messages possible, this is the preferred way to do validation (although isValid() still works).

JSP 2.0 and higher containers call validate() instead of isValid(). The default implementation of this method is to call isValid(). If isValid() returns false, a generic ValidationMessage[] is returned indicating isValid() returned false.

Parameters:
data - The TagData instance.

Returns:
A null object, or zero length array if no errors, an array of ValidationMessages otherwise.

Since:
JSP 2.0

---------------------------------------------------------------

final public void setTagInfo(TagInfo tagInfo)
TagInfo

tagName
TagInfo

setTagInfo

public final void setTagInfo(TagInfo tagInfo)

Set the TagInfo for this class.

Parameters:
tagInfo - The TagInfo this instance is extending

---------------------------------------------------------------

final public TagInfo getTagInfo()
TagInfo

default
TagInfo
getTagInfo

public final TagInfo getTagInfo()

Get the TagInfo for this class.

**Returns:**
the taginfo instance this instance is extending
public class TagFileInfo

extends Object

(TLD)

since 2.0

Tag information for a tag file in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

Since:
    JSP 2.0

Constructor Summary

TagFileInfo(String name, String path, TagInfo tagInfo)

Constructor for TagFileInfo from data in the JSP 2.0 format for TLD.

Method Summary

String getName()

The unique action name of this tag.

String getPath()

Where to find the .tag file implementing this action.

TagInfo getTagInfo()

Returns information about this tag, parsed from the
Methods inherited from class java.lang.\texttt{Object}
\begin{tabular}{l}
\texttt{clone, equals, finalize, getClass, hashCode, notify, notifyAll,}
\texttt{toString, wait, wait, wait}
\end{tabular}

\section*{Constructor Detail}

\texttt{public TagFileInfo(String name, String path, TagInfo tagInfo)}

\texttt{TagFileInfo JSP 2.0 TLD TagFileInfo TLD (Tag Library Descriptor) JSP TagLibrary}
\texttt{TagLibibrary Info TLD taglib TagFileInfo taglib}

\begin{itemize}
\item [\textit{name}]\hfill TagFileInfo
\item [\textit{path}]\hfill TagFileInfo
\item [\textit{tagInfo}]\hfill TagFileInfo
\end{itemize}

\texttt{TagFileInfo}

\texttt{public TagFileInfo(String name, String path, TagInfo tagInfo)}

Constructor for TagFileInfo from data in the JSP 2.0 format for TLD. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor). Note that, since TagLibraryInfo reflects both TLD information and taglib directive information, a TagFileInfo instance is dependent on a taglib directive. This is probably a design error, which may be fixed in the future.

\textbf{Parameters:}

\begin{itemize}
\item [\textit{name}] - The unique action name of this tag
\end{itemize}
path - Where to find the .tag file implementing this action, relative to the location of the TLD file.
tagInfo - The detailed information about this tag, as parsed from the directives in the tag file.

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public String getName()</td>
</tr>
<tr>
<td>return</td>
</tr>
</tbody>
</table>

getName

public String getName()

The unique action name of this tag.

Returns:
The (short) name of the tag.

public String getPath()
.tag
return TLD "." 

getPath

public String getPath()

Where to find the .tag file implementing this action.

Returns:
The path of the tag file, relative to the TLD, or "." if the tag file was defined in an implicit tag file.
public TagInfo getTagInfo()

    return TagInfo

getTagInfo

public TagInfo getTagInfo()

    Returns information about this tag, parsed from the directives in the tag file.

    Returns:
    a TagInfo object containing information about this tag
javax.servlet.jsp.tagext  

Class TagInfo

java.lang.Object  
   \ javax.servlet.jsp.tagext.TagInfo

public class TagInfo
extends Object

(TLD)

Tag information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

---

Field Summary

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>BODY_CONTENT_EMPTY</td>
<td>Static constant for getBodyContent() when it is empty.</td>
</tr>
<tr>
<td>static String</td>
<td>BODY_CONTENT_JSP</td>
<td>Static constant for getBodyContent() when it is JSP.</td>
</tr>
<tr>
<td>static String</td>
<td>BODY_CONTENT_SCRIPTLESS</td>
<td>Static constant for getBodyContent() when it is scriptless.</td>
</tr>
<tr>
<td>static String</td>
<td>BODY_CONTENT_TAG_DEPENDENT</td>
<td>Static constant for getBodyContent() when it is Tag dependent.</td>
</tr>
</tbody>
</table>

Constructor Summary

TagInfo(String tagName, String tagClassName, String bodycontent, String infoString, TagLibraryInfo taglib, TagExtraInfo tagExtraInfo, TagAttributeInfo[] attributeInfo)
   Constructor for TagInfo from data in the JSP 1.1 format for TLD.
Constructor for TagInfo from data in the JSP 1.2 format for TLD.

Constructor for TagInfo from data in the JSP 2.0 format for TLD.

**Method Summary**

<table>
<thead>
<tr>
<th>TagAttributeInfo[]</th>
<th>getAttributes()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Attribute information (in the TLD) on this tag.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getBodyContent()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The bodycontent information for this tag.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getDisplayName()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the displayName.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getInfoString()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The information string for the tag.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getLargeIcon()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the path to the large icon.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getSmallIcon()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Get the path to the small icon.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getTagName()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Name of the class that provides the handler for this tag.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TagExtraInfo</th>
<th>getTagExtraInfo()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The instance (if any) for extra tag information.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TagLibraryInfo</th>
<th>getTagLibrary()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The instance of TagLibraryInfo we belong to.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getTagName()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name of the Tag.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>getTagVariableInfos()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TagVariableInfo[]

Get TagVariableInfo objects associated with this TagInfo.

VariableInfo[]

getVariableInfo(TagData data)

Information on the scripting objects created by this tag at runtime.

boolean

hasDynamicAttributes()

Get dynamicAttributes associated with this TagInfo.

boolean

isValid(TagData data)

Translation-time validation of the attributes.

void

setTagExtraInfo(TagExtraInfo tei)

Set the instance for extra tag information.

void

setTagLibrary(TagLibraryInfo tl)

Set the TagLibraryInfo property.

ValidationMessage[]

validate(TagData data)

Translation-time validation of the attributes.

Methods inherited from class java.lang.Object

class, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

BODY_CONTENT_JSP

public static final String BODY_CONTENT_JSP

Static constant for getBodyContent() when it is JSP.

See Also:

Constant Field Values
BODY_CONTENT_TAG_DEPENDENT

public static final String BODY_CONTENT_TAG_DEPENDENT

Static constant for getBodyContent() when it is Tag dependent.

See Also:
Constant Field Values

BODY_CONTENT_EMPTY

public static final String BODY_CONTENT_EMPTY

Static constant for getBodyContent() when it is empty.

See Also:
Constant Field Values

BODY_CONTENT_SCRIPTLESS

public static final String BODY_CONTENT_SCRIPTLESS

Static constant for getBodyContent() when it is scriptless.

Since:
JSP 2.0
See Also:
Constant Field Values

Constructor Detail
public TagInfo(String tagName, String tagClassName, String bodycontent, String infoString, TagLibraryInfo taglib, TagExtraInfo tagExtraInfo, TagAttributeInfo[] attributeInfo)

TagInfo JSP 1.1 TLD TLD (Tag Library Descriptor) JSP TagLibrary TagLibibraryInfo TLD taglib TagInfo taglib
tagName
tagClassName
bodycontent
infoString
taglib
tagExtraInfo null
attributeInfo AttributeInfo null

TagInfo

public TagInfo(String tagName, String tagClassName, String bodycontent, String infoString, TagLibraryInfo taglib, TagExtraInfo tagExtraInfo, TagAttributeInfo[] attributeInfo)

Constructor for TagInfo from data in the JSP 1.1 format for TLD. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor). Note that, since TagLibibraryInfo reflects both TLD information and taglib directive information, a TagInfo instance is dependent on a taglib directive. This is probably a design error, which may be fixed in the future.

Parameters:
tagName - The name of this tag
tagClassName - The name of the tag handler class
bodycontent - Information on the body content of these tags
infoString - The (optional) string information for this tag
taglib - The instance of the tag library that contains us.
tagExtraInfo - The instance providing extra Tag info. May be null
attributeInfo - An array of AttributeInfo data from descriptor. May be null;

```
public TagInfo(String tagName, String tagClassName,
String bodycontent, String infoString, TagLibraryInfo taglib, TagExtraInfo tagExtraInfo, TagAttributeInfo[] attributeInfo, String displayName, String smallIcon, String largeIcon, TagVariableInfo[] tvi)
```

TagInfo TagInfo JSP 1.2 TLD TLD (Tag Library Descriptor) JSP TagLibrary TagLibLibraryInfo TLD taglib TagInfo taglib

tagName
tagClassName
bodycontent
infoString
taglib
tagExtraInfo null
attributeInfo AttributeInfo null
displayName
smallIcon
largeIcon
tvi TagVariableInfo null

**TagInfo**

```
public TagInfo(String tagName,
String tagClassName,
String bodycontent,
String infoString,
```
Constructor for TagInfo from data in the JSP 1.2 format for TLD. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor). Note that, since TagLibraryInfo reflects both TLD information and taglib directive information, a TagInfo instance is dependent on a taglib directive. This is probably a design error, which may be fixed in the future.

**Parameters:**

- **tagName** - The name of this tag
- **tagClassName** - The name of the tag handler class
- **bodycontent** - Information on the body content of these tags
- **infoString** - The (optional) string information for this tag
- **taglib** - The instance of the tag library that contains us.
- **tagExtraInfo** - The instance providing extra Tag info. May be null
- **attributeInfo** - An array of AttributeInfo data from descriptor. May be null;
- **displayName** - A short name to be displayed by tools
- **smallIcon** - Path to a small icon to be displayed by tools
- **largeIcon** - Path to a large icon to be displayed by tools
- **tvi** - An array of a TagVariableInfo (or null)
TagLiblibraryInfo TLD taglib TagInfo
taglib

    tagName
    tagClassName
    bodycontent
    infoString
taglib
    tagExtraInfo null
    attributeName AttributeInfo null
    displayName
    smallIcon
    largeIcon
tvi TagVariableInfo null
dynamicAttributes true
    since 2.0

TagInfo

public TagInfo(String tagName,
       String tagClassName,
       String bodycontent,
       String infoString,
       TagLibraryInfo taglib,
       TagExtraInfo tagExtraInfo,
       TagAttributeInfo[] attributeInfo,
       String displayName,
       String smallIcon,
       String largeIcon,
       TagVariableInfo[] tvi,
       boolean dynamicAttributes)

Constructor for TagInfo from data in the JSP 2.0 format for TLD. This class is to be instantiated only from the TagLibrary code under request from some JSP code that is parsing a TLD (Tag Library Descriptor). Note that, since TagLiblibraryInfo reflects both TLD information and taglib directive information, a TagInfo instance is dependent on a taglib directive. This is probably a design error, which may be fixed in the future.
Parameters:

tagName - The name of this tag
tagClassName - The name of the tag handler class
bodycontent - Information on the body content of these tags
infoString - The (optional) string information for this tag
taglib - The instance of the tag library that contains us.
tagExtraInfo - The instance providing extra Tag info. May be null
attributeInfo - An array of AttributeInfo data from descriptor. May be null;
displayName - A short name to be displayed by tools
smallIcon - Path to a small icon to be displayed by tools
largeIcon - Path to a large icon to be displayed by tools
tvi - An array of a TagVariableInfo (or null)
dynamicAttributes - True if supports dynamic attributes

Since:
JSP 2.0

Method Detail

public String getTagName()
Tag

return

getTagName

public String getTagName()

The name of the Tag.

Returns:
The (short) name of the tag.

public TagAttributeInfo[] getAttributes()
TLD TLD
getAttributes

public TagAttributeInfo[] getAttributes()

Attribute information (in the TLD) on this tag. The return is an array describing the attributes of this tag, as indicated in the TLD.

Returns:
- The array of TagAttributeInfo for this tag, or a zero-length array if the tag has no attributes.

getVariableInfo

public VariableInfo[] getVariableInfo(TagData data)

Information on the scripting objects created by this tag at runtime. This is a convenience method on the associated TagExtraInfo class.

Parameters:
- data - TagData describing this action.

Returns:
- if a TagExtraInfo object is associated with this TagInfo, the result of getTagExtraInfo().getVariableInfo( data ), otherwise null.

isValid

public boolean isValid(TagData data)

TagExtraInfo
isValid

public boolean isValid(TagData data)

Translation-time validation of the attributes. This is a convenience method on the associated TagExtraInfo class.

Parameters:
   data - The translation-time TagData instance.

Returns:
   Whether the data is valid.

validate

public ValidationMessage[] validate(TagData data)

Translation-time validation of the attributes. This is a convenience method on the associated TagExtraInfo class.

Parameters:
   data - The translation-time TagData instance.

Returns:
   A null object, or zero length array if no errors, an array of ValidationMessages otherwise.

Since:
   JSP 2.0
public void setTagExtraInfo(TagExtraInfo tei)

    tei TagExtraInfo

setTagExtraInfo

public void setTagExtraInfo(TagExtraInfo tei)

    Set the instance for extra tag information.

    Parameters:
        tei - the TagExtraInfo instance

public TagExtraInfo getTagExtraInfo()

    return TagExtraInfo

getTagExtraInfo

public TagExtraInfo getTagExtraInfo()

    The instance (if any) for extra tag information.

    Returns:
        The TagExtraInfo instance, if any.

gpublic String getTagClassName()

    return

ggetTagClassName
public String getTagClassName()

    Name of the class that provides the handler for this tag.

    **Returns:**
    The name of the tag handler class.

public String getBodyContent()

    bodycontent  bodycontent  JSP
    return

getBodyContent

public String getBodyContent()

    The bodycontent information for this tag. If the bodycontent is not defined for this tag, the default of JSP will be returned.

    **Returns:**
    the body content string.

public String getInfoString()

    return  null

getInfoString

public String getInfoString()

    The information string for the tag.

    **Returns:**
    the info string, or null if not defined
public void setTagLibrary(TagLibraryInfo tl)

**TagLibraryInfo**

TagLibraryInfo  
TagLibraryInfo  
TLD

**taglib**  
TagLib  

setter tl

tl  
TagLibraryInfo

**setTagLibrary**

public void setTagLibrary(TagLibraryInfo tl)

Set the TagLibraryInfo property. Note that a TagLibraryInfo element is dependent not just on the TLD information but also on the specific taglib instance used. This means that a fair amount of work needs to be done to construct and initialize TagLib objects. If used carefully, this setter can be used to avoid having to create new TagInfo elements for each taglib directive.

**Parameters:**
tl - the TagLibraryInfo to assign

---

public **TagLibraryInfo** getTagLibrary()

**TagLibraryInfo**

return

**getTagLibrary**

public **TagLibraryInfo** getTagLibrary()

The instance of TagLibraryInfo we belong to.

**Returns:**
the tag library instance we belong to
public String getDisplayName()

    displayName
    return null

getDisplayName

public String getDisplayName()

    Get the displayName.

    Returns:
    A short name to be displayed by tools, or null if not defined

public String getSmallIcon()

    return null

getSmallIcon

public String getSmallIcon()

    Get the path to the small icon.

    Returns:
    Path to a small icon to be displayed by tools, or null if not defined

public String getLargeIcon()

    return null

getLargeIcon

public String getLargeIcon()
public String getLargeIcon()

Get the path to the large icon.

**Returns:**
Path to a large icon to be displayed by tools, or null if not defined

---

public TagVariableInfo[] getTagVariableInfos()
TagInfo TagVariableInfo

return TagVariableInfo 0

getagVariableInfos

public TagVariableInfo[] getTagVariableInfos()

Get TagVariableInfo objects associated with this TagInfo.

**Returns:**
Array of TagVariableInfo objects corresponding to variables declared by this tag, or a zero length array if no variables have been declared

---

public boolean hasDynamicAttributes()
TagInfo dynamicAttributes

return true

since 2.0

hasDynamicAttributes

public boolean hasDynamicAttributes()

Get dynamicAttributes associated with this TagInfo.
Returns:
   True if tag handler supports dynamic attributes

Since:
   JSP 2.0
public abstract class TagLibraryInfo
extends Object

taglib TLD TLD taglib uri

Translation-time information associated with a taglib directive, and its underlying TLD file. Most of the information is directly from the TLD, except for the prefix and the uri values used in the taglib directive

Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>functions</td>
<td>FunctionInfo[]</td>
<td>An array describing the functions that are defined in this tag library.</td>
</tr>
<tr>
<td>info</td>
<td>String</td>
<td>Information (documentation) for this TLD.</td>
</tr>
<tr>
<td>jspversion</td>
<td>String</td>
<td>The version of the JSP specification this tag library is written to.</td>
</tr>
<tr>
<td>prefix</td>
<td>String</td>
<td>The prefix assigned to this taglib from the taglib directive.</td>
</tr>
<tr>
<td>shortname</td>
<td>String</td>
<td>The preferred short name (prefix) as indicated in the TLD.</td>
</tr>
<tr>
<td>tagFiles</td>
<td>TagFileInfo[]</td>
<td>An array describing the tag files that are defined in</td>
</tr>
</tbody>
</table>
### Constructor Summary

```java
protected TagLibraryInfo(String prefix, String uri)
    Constructor.
```

### Method Summary

```java
FunctionInfo getFunction(String name)
    Get the FunctionInfo for a given function name, looking through all the functions in this tag library.
```

```java
FunctionInfo[] getFunctions()
    An array describing the functions that are defined in this tag library.
```

```java
String getInfoString()
    Information (documentation) for this TLD.
```

```java
String getPrefixString()
    The prefix assigned to this taglib from the taglib directive
```

```java
String getReliableURN()
    The "reliable" URN indicated in the TLD (the uri element).
```

```java
String getRequiredVersion()
    A string describing the required version of the JSP
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getShortName()</td>
<td>The preferred short name (prefix) as indicated in the TLD.</td>
</tr>
<tr>
<td>TagInfo getTag(String shortname)</td>
<td>Get the TagInfo for a given tag name, looking through all the tags in this tag library.</td>
</tr>
<tr>
<td>TagFileInfo getTagFile(String shortname)</td>
<td>Get the TagFileInfo for a given tag name, looking through all the tag files in this tag library.</td>
</tr>
<tr>
<td>TagFileInfo[] getTagFiles()</td>
<td>An array describing the tag files that are defined in this tag library.</td>
</tr>
<tr>
<td>abstract TagLibraryInfo[] getTagLibraryInfos()</td>
<td>Returns an array of TagLibraryInfo objects representing the entire set of tag libraries (including this TagLibraryInfo) imported by taglib directives in the translation unit that references this TagLibraryInfo.</td>
</tr>
<tr>
<td>TagInfo[] getTags()</td>
<td>An array describing the tags that are defined in this tag library.</td>
</tr>
<tr>
<td>String getURI()</td>
<td>The value of the uri attribute from the taglib directive for this library.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

prefix
protected `String` `prefix`

The prefix assigned to this taglib from the taglib directive.

---

`uri`

protected `String` `uri`

The value of the uri attribute from the taglib directive for this library.

---

`tags`

protected `TagInfo[]` `tags`

An array describing the tags that are defined in this tag library.

---

`tagFiles`

protected `TagFileInfo[]` `tagFiles`

An array describing the tag files that are defined in this tag library.

**Since:**

JSP 2.0

---

`functions`

protected `FunctionInfo[]` `functions`
An array describing the functions that are defined in this tag library.

**Since:**
JSP 2.0

---

tlibversion

protected *String* `tlibversion`

The version of the tag library.

---

jspversion

protected *String* `jspversion`

The version of the JSP specification this tag library is written to.

---

shortname

protected *String* `shortname`

The preferred short name (prefix) as indicated in the TLD.

---

urn

protected *String* `urn`

The "reliable" URN indicated in the TLD.
protected String info

Information (documentation) for this TLD.

**Constructor Detail**

protected TagLibraryInfo(String prefix, String uri)

TagInfo TLD TagAttributeInfo

  prefix taglib
  uri taglib URI

TagLibraryInfo

protected TagLibraryInfo(String prefix, String uri)

Constructor.

**Parameters:**

  prefix - the prefix actually used by the taglib directive
  uri - the URI actually used by the taglib directive

**Method Detail**

public String getURI()

taglib uri

return uri

getURI
public String getURI()

    The value of the uri attribute from the taglib directive for this library.

    Returns:
    the value of the uri attribute

public String getPrefixString()

    The prefix assigned to this taglib from the taglib directive

    Returns:
    the prefix assigned to this taglib from the taglib directive

public String getShortName()

    The preferred short name (prefix) as indicated in the TLD. This may be used by authoring tools as the preferred prefix to use when creating an taglib directive for this library.

    Returns:
    the preferred short name for the library
public String getReliableURN()
TLD ""URNuri URN taglib
   return TLD URN

getReliableURN

public String getReliableURN()

   The "reliable" URN indicated in the TLD (the uri element). This may be used by authoring tools as a global identifier to use when creating a taglib directive for this library.

   Returns:
   a reliable URN to a TLD like this

public String getInfoString()
TLD
   return

getInfoString

public String getInfoString()

   Information (documentation) for this TLD.

   Returns:
   the info string for this tag lib

public String getRequiredVersion()
JSP
   return JSP
See also javax.servlet.jsp.JspEngineInfo
getRequiredVersion

public String getRequiredVersion()

A string describing the required version of the JSP container.

Returns:
the (minimal) required version of the JSP container.

See Also:
JspEngineInfo

public TagInfo[] getTags()

return TagInfo 0

getTags

public TagInfo[] getTags()

An array describing the tags that are defined in this tag library.

Returns:
the TagInfo objects corresponding to the tags defined by this tag library, or a zero length array if this tag library defines no tags

public TagFileInfo[] getTagFiles()

return TagFileInfo 0

since 2.0

getTagFiles
public TagFileInfo[] getTagFiles()

An array describing the tag files that are defined in this tag library.

**Returns:**
the TagFileInfo objects corresponding to the tag files defined by this tag library, or a zero length array if this tag library defines no tags files

**Since:**
JSP 2.0

---

public TagInfo getTag(String shortname)

TagInfo

```
shortname
return TagInfo null
```

getTag

public TagInfo getTag(String shortname)

Get the TagInfo for a given tag name, looking through all the tags in this tag library.

**Parameters:**
shortname - The short name (no prefix) of the tag

**Returns:**
the TagInfo for the tag with the specified short name, or null if no such tag is found

---

public TagFileInfo getTagFile(String shortname)

TagFileInfo

```
shortname
return TagFileInfo null
since 2.0
```
**getTagFile**

public [TagFileInfo](#) `getTagFile(String shortname)`

Get the TagFileInfo for a given tag name, looking through all the tag files in this tag library.

**Parameters:**
- `shortname` - The short name (no prefix) of the tag

**Returns:**
the TagFileInfo for the specified Tag file, or null if no Tag file is found

**Since:**
JSP 2.0

---

**getFunctions**

public `FunctionInfo[] getFunctions()`

```
    return 0
    since 2.0
```

**getFunctions**

public `FunctionInfo[] getFunctions()`

An array describing the functions that are defined in this tag library.

**Returns:**
the functions defined in this tag library, or a zero length array if the tag library defines no functions.

**Since:**
JSP 2.0

---

**getFunction**

public `FunctionInfo getFunction(String name)`

`FunctionInfo`

`name`
getFunction

```java
public FunctionInfo getFunction(String name)
```

Get the FunctionInfo for a given function name, looking through all the functions in this tag library.

**Parameters:**
- `name` - The name (no prefix) of the function

**Returns:**
- the FunctionInfo for the function with the given name, or null if no such function exists

**Since:**
- JSP 2.0

getTagLibraryInfos

```java
public abstract TagLibraryInfo[] getTagLibraryInfos()
```

Returns an array of TagLibraryInfo objects representing the entire set of tag libraries (including this TagLibraryInfo) imported by taglib directives in the translation unit that references this TagLibraryInfo. If a tag library is imported more than once and bound to different prefixes, only the TagLibraryInfo bound to the first prefix must be included in the returned array.

**Returns:**
- Array of TagLibraryInfo objects representing the entire set of tag libraries (including this TagLibraryInfo) imported by taglib directives in the translation unit that references this TagLibraryInfo.

**Since:**
javax.servlet.jsp.tagext  **Class TagLibraryValidator**

java.lang.Object  
  ↓ javax.servlet.jsp.tagext.TagLibraryValidator

```java
public abstract class TagLibraryValidator
    extends Object
```

JSP JSP XML

TLD TagLibraryValidator init

JSP

  •
  • setInitParams(Map)

validate(String, String, PageData) XML uri
URI XML URI TagLibraryValidator uri

TagLibraryValidator XSchema

JSP validate() TagLibraryValidator

  JSP 2.0 JSP jsp:id JSP "id" id jsp:id
XML XML TagLibraryValidator ValidationMessage

id jsp http://java.sun.com/JSP/PageTagLibraryValidator id
uri

Translation-time validator class for a JSP page. A validator operates on the XML view associated with the JSP page.
The TLD file associates a TagLibraryValidator class and some init arguments with a tag library.

The JSP container is responsible for locating an appropriate instance of the appropriate subclass by

- new a fresh instance, or reuse an available one
- invoke the setInitParams(Map) method on the instance

Once initialized, the validate(String, String, PageData) method will be invoked, where the first two arguments are the prefix and uri for this tag library in the XML View. The prefix is intended to make it easier to produce an error message. However, it is not always accurate. In the case where a single URI is mapped to more than one prefix in the XML view, the prefix of the first URI is provided. Therefore, to provide high quality error messages in cases where the tag elements themselves are checked, the prefix parameter should be ignored and the actual prefix of the element should be used instead. TagLibraryValidators should always use the uri to identify elements as belonging to the tag library, not the prefix.

A TagLibraryValidator instance may create auxiliary objects internally to perform the validation (e.g. an XSchema validator) and may reuse it for all the pages in a given translation run.

The JSP container is not guaranteed to serialize invocations of validate() method, and TagLibraryValidators should perform any synchronization they may require.

As of JSP 2.0, a JSP container must provide a jsp:id attribute to provide higher quality validation errors. The container will track the JSP pages as passed to the container, and will assign to each element a unique "id", which is passed as the value of the jsp:id attribute. Each XML element in the XML view available will be extended with this attribute. The TagLibraryValidator can then use the attribute in one or more ValidationMessage objects. The container then, in turn, can use these values to provide more precise information on the location of an error.

The actual prefix of the id attribute may or may not be jsp but it will always map to the namespace http://java.sun.com/JSP/Page. A
TagLibraryValidator implementation must rely on the uri, not the prefix, of the id attribute.

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>TagLibraryValidator()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Sole constructor.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getInitParameters()</td>
<td>Map&lt;String, Object&gt; -&gt; Map</td>
<td>Get the init parameters data as an immutable Map.</td>
</tr>
<tr>
<td>release()</td>
<td>void</td>
<td>Release any data kept by this instance for validation purposes.</td>
</tr>
<tr>
<td>setInitParameters(Map&lt;String, Object&gt; map)</td>
<td>void</td>
<td>Set the init data in the TLD for this validator.</td>
</tr>
<tr>
<td>validate(String prefix, String uri, PageData page)</td>
<td>ValidationMessage[] -&gt; void</td>
<td>Validate a JSP page.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

### Constructor Detail

```java
public TagLibraryValidator()
```

TagLibraryValidator
public TagLibraryValidator()

Sole constructor. (For invocation by subclass constructors, typically implicit.)

Method Detail

public void setInitParameters(java.util.Map<K, V> map)

TLD init

map init Map

setInitParameters

public void setInitParameters(Map<String, Object> map)

Set the init data in the TLD for this validator. Parameter names are keys, and parameter values are the values.

Parameters:
map - A Map describing the init parameters

public java.util.Map<K, V> getInitParameters()

Map init

return init

getInitParameters

public Map<String, Object> getInitParameters()

Get the init parameters data as an immutable Map. Parameter names are keys, and parameter values are the values.

Returns:
The init parameters as an immutable map.
public ValidationMessage[] validate(String prefix, String uri, PageData page)

JSP XML URI null

ValidationMessage 0

prefix XML
uri
page JspData
return null 0 ValidationMessage

validate

public ValidationMessage[] validate(String prefix, String uri, PageData page)

Validate a JSP page. This will get invoked once per unique tag library URI in the XML view. This method will return null if the page is valid; otherwise the method should return an array of ValidationMessage objects. An array of length zero is also interpreted as no errors.

Parameters:
prefix - the first prefix with which the tag library is associated, in the XML view. Note that some tags may use a different prefix if the namespace is redefined.
uri - the tag library’s unique identifier
page - the JspData page object

Returns:
A null object, or zero length array if no errors, an array of ValidationMessages otherwise.

public void release()
public void release()

Release any data kept by this instance for validation purposes.
javax.servlet.jsp.tagext  **Class TagSupport**

**java.lang.Object**
  ↓ javax.servlet.jsp.tagext.TagSupport

**All Implemented Interfaces:**
  Serializable, IterationTag, JspTag, Tag

**Direct Known Subclasses:**
  AttributeTag, BodyTagSupport, ConverterELTag, ConverterTag, FacetTag, ValidatorELTag, ValidatorTag

---

**public class TagSupport**

extends **Object**

implements **IterationTag, Serializable**

**Implements:** IterationTag, java.io.Serializable

**Extended by:** AttributeTag, BodyTagSupport, ConverterELTag, ConverterTag, FacetTag, ValidatorELTag, ValidatorTag

Tag

TagSupport TagSupport  Tag  IterationTag  Tag
getter TagSupport

TagSupport

A base class for defining new tag handlers implementing Tag.

The TagSupport class is a utility class intended to be used as the base class for new tag handlers. The TagSupport class implements the Tag and IterationTag interfaces and adds additional convenience methods including getter methods for the properties in Tag. TagSupport has one static method that is included to facilitate coordination among cooperating tags.
Many tag handlers will extend TagSupport and only redefine a few methods.

See Also:
  Serialized Form

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String id</td>
<td>The value of the id attribute of this tag; or null.</td>
</tr>
<tr>
<td>protected PageContext pageContext</td>
<td>The PageContext.</td>
</tr>
</tbody>
</table>

### Fields inherited from interface javax.servlet.jsp.tagext.IterationTag

- EVAL_BODY_AGAIN

### Fields inherited from interface javax.servlet.jsp.tagext.Tag

- EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

### Constructor Summary

TagSupport()  
Default constructor, all subclasses are required to define only a public constructor with the same signature, and to call the superclass constructor.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int doAfterBody()</td>
<td>Default processing for a body.</td>
</tr>
<tr>
<td>int doEndTag()</td>
<td>Default processing of the end tag returning EVAL_PAGE.</td>
</tr>
<tr>
<td>int doStartTag()</td>
<td>Default processing of the start tag, returning</td>
</tr>
</tbody>
</table>
### findAncestorWithClass

The public static method `findAncestorWithClass` accepts two `Tag`-type parameters, `Tag from` and `Class klass`, and returns a `Tag` instance. The method finds the instance of a given class type that is closest to a given instance.

```java
static Tag findAncestorWithClass(Tag from, Class klass)
```

**Example**: This method is often used in tag libraries to find the root `Tag` instance of a given type, such as `html:div`, in the hierarchy of `Tag` instances.

### getId

The method `getId` returns the value of the id attribute of this tag; or null.

```java
String getId()
```

**Example**: To access the id attribute of a `Tag` instance:

```java
String id = myTag.getId();
```

### getParent

The method `getParent` returns the `Tag` instance most closely enclosing this tag instance.

```java
Tag getParent()
```

**Example**: To find the parent `Tag` instance of a given `Tag`:

```java
Tag parent = myTag.getParent();
```

### getValue

The method `getValue` retrieves the value associated with a key.

```java
Object getValue(String k)
```

**Example**: To access a value associated with a key:

```java
Object value = myTag.getValue(key);
```

### getValues

The method `getValues` retrieves all the values kept by this tag handler.

```java
Enumeration<String> getValues()
```

**Example**: To retrieve all the values:

```java
Enumeration<String> values = myTag.getValues();
```

### release

The method `release` releases the state of this `Tag` instance.

```java
void release()
```

**Example**: To release the state:

```java
myTag.release();
```

### removeValue

The method `removeValue` removes a value associated with a key.

```java
void removeValue(String k)
```

**Example**: To remove a value:

```java
myTag.removeValue(key);
```

### setId

The method `setId` sets the id attribute for this `Tag` instance.

```java
void setId(String id)
```

**Example**: To set the id attribute:

```java
myTag.setId(id);
```

### setPageContext

The method `setPageContext` sets the page context.

```java
void setPageContext(PageContext pageContext)
```

**Example**: To set the page context:

```java
myTag.setPageContext(context);
```

### setParent

The method `setParent` sets the nesting `Tag` of this `Tag` instance.

```java
void setParent(Tag t)
```

**Example**: To set the nesting `Tag`:

```java
myTag.setParent(nestingTag);
```

### setValue

The method `setValue` associates a value with a `String` key.

```java
void setValue(String k, Object o)
```

**Example**: To associate a value with a key:

```java
myTag.setValue(key, value);
```

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
id

protected String id

The value of the id attribute of this tag; or null.

pageContext

protected PageContext pageContext

The PageContext.

### Constructor Detail

public TagSupport()

TagSupport

public TagSupport()

Default constructor, all subclasses are required to define only a public constructor with the same signature, and to call the superclass constructor. This constructor is called by the code generated by the JSP translator.

### Method Detail

final public static Tag findAncestorWithClass(Tag from, Class<T> klass)
Tag getParent

observable observable void

JSP

observable findAncestorWithClass

observable

    from
    klass
    return

findAncestorWithClass

public static final Tag findAncestorWithClass(Tag from,
                                               Class klass)

    Find the instance of a given class type that is closest to a given instance. This method uses the getParent method from the Tag interface. This method is used for coordination among cooperating tags.

    The current version of the specification only provides one formal way of indicating the observable type of a tag handler: its tag handler implementation class, described in the tag-class subelement of the tag element. This is extended in an informal manner by allowing the tag library author to indicate in the description subelement an observable type. The type should be a subtype of the tag handler implementation class or void. This additional constraint can be exploited by a specialized container that knows about that specific tag library, as in the case of the JSP standard tag library.

    When a tag library author provides information on the observable type of a tag handler, client programmatic code should adhere to that constraint. Specifically, the Class passed to findAncestorWithClass should be a subtype of the observable type.
Parameters:
  from - The instance from where to start looking.
  klass - The subclass of Tag or interface to be matched

Returns:
  the nearest ancestor that implements the interface or is an
  instance of the class specified

public int doStartTag() throws JspException

SKIP_BODY

    return SKIP_BODY

Throws
    JspException:

See also
    doStartTag()

doStartTag

public int doStartTag() throws JspException

Default processing of the start tag, returning SKIP_BODY.

Specified by:
    doStartTag in interface Tag

Returns:
    SKIP_BODY

Throws:
    JspException - if an error occurs while processing this tag

See Also:
    Tag.doStartTag()

public int doEndTag() throws JspException

EVAL_PAGE

    return EVAL_PAGE

Throws
    JspException:

See also
    doEndTag()
doEndTag

public int doEndTag()
    throws JspException

Default processing of the end tag returning EVAL_PAGE.

Specified by:
    doEndTag in interface Tag

Returns:
    EVAL_PAGE

Throws:
    JspException - if an error occurs while processing this tag

See Also:
    Tag.doEndTag()  

public int doAfterBody() throws JspException

    return SKIP_BODY

    Throws
    JspException:

    See also
    doAfterBody()

doAfterBody

public int doAfterBody()
    throws JspException

Default processing for a body.

Specified by:
    doAfterBody in interface IterationTag

Returns:
    SKIP_BODY

Throws:
    JspException - if an error occurs while processing this tag

See Also:
    IterationTag.doAfterBody()
public void release()

See also release()

release

public void release()

Release state.

Specified by: release in interface Tag
See Also: Tag.release()

public void setParent(Tag t)

See also setParent(Tag)

setParent

public void setParent(Tag t)

Set the nesting tag of this tag.

Specified by: setParent in interface Tag
Parameters: t - The parent Tag.
See Also: Tag.setParent(Tag)
**public Tag getParent()**

    return null

See also  
getParent()

---

**getParent**

**public Tag getParent()**

    The Tag instance most closely enclosing this tag instance.

    **Specified by:**
    getParents in interface Tag

    **Returns:**
    the parent tag instance or null

    **See Also:**
    Tag.getParent()

---

**public void setId(String id)**

    **id**

    id      String

**setId**

**public void setId(String id)**

    Set the id attribute for this tag.

    **Parameters:**
    id - The String for the id.

---

**public String getId()**

    **id**  null

    return     id  null
**getId**

public String getId()

The value of the id attribute of this tag; or null.

Returns:
the value of the id attribute, or null

---

**public void setPageContext(PageContext pageContext)**

*pageContext* PageContext
See also *setPageContext*

---

**setPageContext**

public void setPageContext(PageContext pageContext)

Set the page context.

Specified by:
*setPageContext* in interface *Tag*

Parameters:
pageContext - The PageContext.

See Also:
*Tag.setPageContext(javax.servlet.jsp.PageContext)*

---

**public void setValue(String k, Object o)**

String

k String
o
setValue

public void setValue(String k, Object o)

Associate a value with a String key.

Parameters:
  k - The key String.
  o - The value to associate.

getValue

public Object getValue(String k)

    k
    return null

getValue

public Object getValue(String k)

    Get the value associated with a key.

Parameters:
    k - The string key.

Returns:
    The value associated with the key, or null.

removeValue

public void removeValue(String k)

    k

removeValue

public void removeValue(String k)
Remove a value associated with a key.

Parameters:
  k - The string key.

public java.util.Enumeration<E> getValues()

  return null Enumeration

getValues

public Enumeration<String> getValues()

  Enumerate the keys for the values kept by this tag handler.

Returns:
  An enumeration of all the keys for the values set, or null or an empty Enumeration if no values have been set.
public class TagVariableInfo
extends Object

(Object)

Variable information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time. This object should be immutable. This information is only available in JSP 1.2 format TLDs or above.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>TagVariableInfo(String nameGiven, String nameFromAttribute, String className, boolean declare, int scope) Constructor for TagVariableInfo.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getClassName() The body of the &lt;variable-class&gt; element.</td>
</tr>
<tr>
<td>boolean getDeclare() The body of the &lt;declare&gt; element.</td>
</tr>
<tr>
<td>String getNameFromAttribute() The body of the &lt;name-from-attribute&gt; element.</td>
</tr>
<tr>
<td>String getNameGiven() The body of the &lt;name-given&gt; element.</td>
</tr>
<tr>
<td>int getScope()</td>
</tr>
</tbody>
</table>
The body of the `<scope>` element.

Methods inherited from class java.lang.**Object**
close, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

public TagVariableInfo(String nameGiven, String nameFromAttribute, String className, boolean declare, int scope)

TagVariableInfo

<table>
<thead>
<tr>
<th>nameGiven</th>
<th>&lt;name-given&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>nameFromAttribute</td>
<td>&lt;name-from-attribute&gt;</td>
</tr>
<tr>
<td>className</td>
<td>&lt;variable-class&gt;</td>
</tr>
<tr>
<td>declare</td>
<td>&lt;declare&gt;</td>
</tr>
<tr>
<td>scope</td>
<td>&lt;scope&gt;</td>
</tr>
</tbody>
</table>

TagVariableInfo

public TagVariableInfo(String nameGiven, String nameFromAttribute, String className, boolean declare, int scope)

Constructor for TagVariableInfo.

**Parameters:**

nameGiven - value of `<name-given>`
nameFromAttribute - value of `<name-from-attribute>`
className - value of `<variable-class>`
declare - value of `<declare>`
scope - value of `<scope>`
public String getNameGiven()

    return

getNameGiven

public String getNameGiven()

    The body of the <name-given> element.

    Returns:
    The variable name as a constant

public String getNameFromAttribute()

    return

getNameFromAttribute

public String getNameFromAttribute()

    The body of the <name-from-attribute> element. This is the name of an attribute whose (translation-time) value will give the name of the variable. One of <name-given> or <name-from-attribute> is required.

    Returns:
    The attribute whose value defines the variable name

public String getClassName()
getClassName

public String getClassName()

The body of the <variable-class> element.

Returns:
The name of the class of the variable or 'java.lang.String' if not defined in the TLD.

getDeclare

public boolean getDeclare()

The body of the <declare> element.

Returns:
Whether the variable is to be declared or not. If not defined in the TLD, 'true' will be returned.

getScope

public int getScope()

The body of the <scope> element.

Returns:
The scope of the variable.
public int getScope()

    The body of the <scope> element.

    **Returns:**
    The scope to give the variable. NESTED scope will be returned if not defined in the TLD.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
public interface Target

Target J2EE

A Target interface represents a single logical core server of one instance of a J2EE platform product. It is a designator for a server and the implied location to copy a configured application for the server to access.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getDescription()</code></td>
<td>Retrieve other descriptive information about the target.</td>
</tr>
<tr>
<td><code>getName()</code></td>
<td>Retrieve the name of the target server.</td>
</tr>
</tbody>
</table>

### Method Detail

**public String getName()**

`getName`

**String getName()**

Retrieve the name of the target server.
getDescription

String getDescription()

Retrieve other descriptive information about the target.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.enterprise.deploy.spi.exceptions Class TargetException

java.lang.Object
   ∧ java.lang.Throwable
      ∧ java.lang.Exception
         ∧ javax.enterprise.deploy.spi.exceptions.TargetException

All Implemented Interfaces:
   Serializable

public class TargetException
extends Exception

Extends: Throwable > Exception

This exception is to report bad target designators.

See Also:
  Serialized Form

Constructor Summary

| TargetException(String s) |
| Creates an new TargetException object.

Method Summary

Methods inherited from class java.lang.Throwables
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public TargetException(String s)
TargetException

    s

TargetException

public TargetException(String s)

    Creates an new TargetException object.

    Parameters:
    s - a string indicating what was wrong with the target.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.enterprise.deploy.spi  Interface TargetModuleID

public interface TargetModuleID

TargetModuleID EARJARWAR RAR
TargetModuleID EAR TargetModuleID J2EE
    TargetModuleID

A TargetModuleID interface represents a unique identifier for a deployed application module. A deployable application module can be an EAR, JAR, WAR or RAR file. A TargetModuleID can represent a root module or a child module. A root module TargetModuleID has no parent. It represents a deployed EAR file or stand alone module. A child module TargetModuleID represents a deployed sub module of a J2EE application. A child TargetModuleID has only one parent, the super module it was bundled and deployed with. The identifier consists of the target name and the unique identifier for the deployed application module.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>TargetModuleID[]</td>
</tr>
<tr>
<td>String</td>
</tr>
<tr>
<td>TargetModuleID</td>
</tr>
<tr>
<td>Target</td>
</tr>
</tbody>
</table>
### Method Detail

**public Target getTarget()**

```
  return Target
```

**getTarget**

```
Target getTarget()
```

Retrieve the name of the target server this module was deployed to.

**Returns:**

- Target an object representing a server target.

**public String getModuleID()**

```
  ID
```

**getModuleID**

```
String getModuleID()
```

Retrieve the id assigned to represent the deployed module.
public String getWebURL()

String getWebURL()

If this TargetModulID represents a web module retrieve the URL for it.

Returns:
the URL of a web module or null if the module is not a web module.

public String toString()

String toString()

Retrieve the identifier representing the deployed module.

Overrides:
toString in class Object

public TargetModuleID getParentTargetModuleID()

TargetModuleID getParentTargetModuleID()

return null
TargetModuleID getParentTargetModuleID()

Retrieve the identifier of the parent object of this deployed module. If there is no parent then this is the root object deployed. The root could represent an EAR file or it could be a stand alone module that was deployed.

Returns:
the TargetModuleID of the parent of this object. A null value means this module is the root object deployed.

public TargetModuleID[] getChildTargetModuleID()

return TargetModuleID null

getChildTargetModuleID

TargetModuleID[] getChildTargetModuleID()

Retrieve a list of identifiers of the children of this deployed module.

Returns:
a list of TargetModuleIDs identifying the children of this object. A null value means this module has no children.
public interface TelephoneNumber

A simple re-usable entity class that defines attributes of a telephone number.

Author:
   Farrukh S. Najmi

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getAreaCode()</strong></td>
<td>Gets the area code.</td>
</tr>
<tr>
<td><strong>getCountryCode()</strong></td>
<td>Gets the country code.</td>
</tr>
<tr>
<td><strong>getExtension()</strong></td>
<td>Gets the internal extension.</td>
</tr>
<tr>
<td><strong>getNumber()</strong></td>
<td>Gets the telephone number suffix, not including the country or area code.</td>
</tr>
<tr>
<td><strong>getType()</strong></td>
<td>The type of telephone number (for example, &quot;fax&quot;).</td>
</tr>
<tr>
<td><strong>getUrl()</strong></td>
<td>Gets the URL that can dial this number electronically.</td>
</tr>
<tr>
<td><strong>setAreaCode(String)</strong></td>
<td>Sets the area code.</td>
</tr>
<tr>
<td><strong>setCountryCode(String)</strong></td>
<td>Sets country code.</td>
</tr>
<tr>
<td><strong>setExtension(String)</strong></td>
<td>Sets internal extension.</td>
</tr>
</tbody>
</table>
void Sets the internal extension.

void setNumber(String number) Sets the telephone number suffix, not including the country or area code.

void setType(String type) Sets the type of telephone number (for example, "fax").

void setUrl(String url) Sets the URL that can dial this number electronically.

Method Detail

public String getCountryCode() throws JAXRException

1

   return

Throws JAXRException: JAXR

getCountryCode

String countryCode() throws JAXRException

   Gets the country code. Default is an empty String.

   Capability Level: 1

   Returns:
   the country code

   Throws:
   JAXRException - If the JAXR provider encounters an internal error
public String getAreaCode() throws JAXRException

1

    return
    Throws JAXRException: JAXR

getAreaCode

String getAreaCode() throws JAXRException

Gets the area code. Default is an empty String.

Capability Level: 1

Returns: the area code

Throws: JAXRException - If the JAXR provider encounters an internal error

-------------------

public String getNumber() throws JAXRException

0

    return
    Throws JAXRException: JAXR

getNumber

String getNumber() throws JAXRException
Gets the telephone number suffix, not including the country or area code. Default is an empty String.

**Capability Level: 0**

**Returns:**
the telephone number

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

**public String getExtension() throws JAXRException**

1

```
return

Throws JAXRException: JAXR
```

---

**getExtension**

**String getExtension()**

throws JAXRException

Gets the internal extension. Default is an empty String.

**Capability Level: 1**

**Returns:**
the internal extension number

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

**public String getUrl() throws JAXRException**
URL NULL

1

```java
public String getType() throws JAXRException
"fax" String
```

0

```java
String getType()
```

getURL

```java
String getURL() throws JAXRException
```

Gets the URL that can dial this number electronically. Default is a NULL String.

**Capability Level: 1**

**Returns:**
the url

**Throws:**

JAXRException - If the JAXR provider encounters an internal error

```
```
The type of telephone number (for example, “fax”). Any String would do.

**Capability Level:** 0

**Returns:**
the type for this TelephoneNumber, which is an arbitrary String

**Throws:**

*JAXRException* - If the JAXR provider encounters an internal error

```java
public void setCountryCode(String countryCode) throws JAXRException
```

1

`countryCode`

**Throws**

*JAXRException*: JAXR

**setCountryCode**

```java
void setCountryCode(String countryCode)
```

Sets country code.

**Capability Level:** 1

**Parameters:**
countryCode - the country code

**Throws:**

*JAXRException* - If the JAXR provider encounters an internal error
public void setAreaCode(String areaCode) throws JAXRException

  areaCode
  Throws  JAXRException:  JAXR

setAreaCode

void setAreaCode(String areaCode)
   throws JAXRException

Sets the area code.

Capability Level: 1

Parameters:
  areaCode - the area code

Throws:
  JAXRException - If the JAXR provider encounters an internal error

public void setNumber(String number) throws JAXRException

  number
  Throws  JAXRException:  JAXR

setNumber
void setNumber(String number) throws JAXRException

Sets the telephone number suffix, not including the country or area code.

Capability Level: 0

Parameters:
number - the telephone number

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setExtension(String extension) throws JAXRException

1

extension

Throws JAXRException: JAXR

setExtension

void setExtension(String extension) throws JAXRException

Sets the internal extension.

Capability Level: 1

Parameters:
extension - the internal extension number

Throws:
JAXRException - If the JAXR provider encounters an internal error
public void setUrl(String url) throws JAXRException

Sets the URL that can dial this number electronically.

**Capability Level: 1**

**Parameters:**
- url - the URL string

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

public void setType(String type) throws JAXRException

Sets the type of this number.

**Parameters:**
- type - the type string

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error
void setType(String type)
throws JAXRException

Sets the type of telephone number (for example, "fax"). Any String will do.

**Capability Level: 0**

**Parameters:**
- type - the type for this TelephoneNumber, which is an arbitrary String

**Throws:**
- JAXRException - If the JAXR provider encounters an internal error

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
`javax.persistence Annotation Type Temporal`

```java
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Temporal

Implements: Annotation
@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

    java.util.Date    java.util.Calendar

Temporal    Basic

    @Temporal(DATE)
    protected java.util.Date endDate;

    since    Java Persistence 1.0

This annotation must be specified for persistent fields or properties of type `Date` and `Calendar`. It may only be specified for fields or properties of these types.

The `Temporal` annotation may be used in conjunction with the `Basic` annotation.

Example:

    @Temporal(DATE)
    protected java.util.Date endDate;

Since:
Java Persistence 1.0
<table>
<thead>
<tr>
<th>Required Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TemporalType</strong></td>
</tr>
</tbody>
</table>

**Element Detail**

**abstract public TemporalType value()**

java.util.Date  java.util.Calendar

**value**

**public abstract TemporalType value**

The type used in mapping java.util.Date or java.util.Calendar.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.persistence **Enum TemporalType**

java.lang.Object
  ↳ java.lang.Enum<TemporalType>
  ↳ javax.persistence.TemporalType

**All Implemented Interfaces:**
  [Serializable, Comparable<TemporalType>]

---

```java
public enum TemporalType
  extends Enum<TemporalType>

Extends: Enum<E>
```

*java.util.Date  java.util.Calendar
  since  Java Persistence 1.0

Type used to indicate a specific mapping of `Date` or `Calendar`.

**Since:**
  Java Persistence 1.0

---

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATE</td>
<td>Map as java.sql.Date</td>
</tr>
<tr>
<td>TIME</td>
<td>Map as java.sql.Time</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>Map as java.sql.Timestamp</td>
</tr>
</tbody>
</table>

---

### Method Summary
<table>
<thead>
<tr>
<th>static TemporalType</th>
<th><code>valueOf(String name)</code></th>
<th>Returns the enum constant of this type with the specified name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static TemporalType[]</td>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang. Enum
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

Methods inherited from class java.lang. Object
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Enum Constant Detail

**DATE**

public static final TemporalType DATE

Map as java.sql.Date

**TIME**

public static final TemporalType TIME

Map as java.sql.Time
public static final TemporalType TIMESTAMP

Map as java.sql.Timestamp

**Method Detail**

final public static TemporalType[] values()

values

public static final TemporalType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```
for(TemporalType c : TemporalType.values())
    System.out.println(c);
```

**Returns:**
an array containing the constants of this enum type, in the order they're declared

public static TemporalType valueOf(String name)

valueOf

public static TemporalType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)
Parameters:
   name - the name of the enum constant to be returned.

Returns:
   the enum constant with the specified name

Throws:
   IllegalArgumentException - if this enum type has no constant
   with the specified name
javax.jms Interface TemporaryQueue

All Superinterfaces:
   Destination, Queue

public interface TemporaryQueue
extends Queue

Implements: Queue

TemporaryQueue Connection Queue Connection
TemporaryQueue Session QueueSession Session
TemporaryQueue Pub/Sub QueueSession PTP

version 1.1 - February 2, 2002
See also createTemporaryQueue(), createTemporaryQueue()

A TemporaryQueue object is a unique queue object created for the duration of a Connection. It is a system-defined queue that can be consumed only by the Connection that created it.

A TemporaryQueue object can be created at either the Session or QueueSession level. Creating it at the Session level allows to the TemporaryQueue to participate in transactions with objects from the Pub/Sub domain. If it is created at the QueueSession, it will only be able to participate in transactions with objects from the PTP domain.

Version:
   1.1 - February 2, 2002
Author:
   Mark Hapner, Rich Burridge, Kate Stout
See Also:
   Session.createTemporaryQueue(),
   QueueSession.createTemporaryQueue()
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>delete</td>
<td>void delete()</td>
<td>Deletes this temporary queue.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface javax.jms.Queue**

- getQueueName
- toString

**Method Detail**

```java
public void delete() throws JMSException
```

**Throws**

- JMSException: JMS

**delete**

```java
void delete()
```

Throws JMSException

Deletes this temporary queue. If there are existing receivers still using it, a JMSException will be thrown.

**Throws:**

- JMSException - if the JMS provider fails to delete the temporary queue due to some internal error.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to **license terms**.
PS:
javax.jms Interface TemporaryTopic

All Superinterfaces:
   Destination, Topic

public interface TemporaryTopic

extends Topic

Implements: Topic

TemporaryTopic  Connection  Topic  Connection
TemporaryTopic  Session  TopicSession  Session
TemporaryTopic  PTP  TopicSession  TemporaryTopic
Pub/Sub

version  1.1 - February 2, 2002  en
See also  createTemporaryTopic(), createTemporaryTopic()

A TemporaryTopic object is a unique Topic object created for the duration of a Connection. It is a system-defined topic that can be consumed only by the Connection that created it.

A TemporaryTopic object can be created either at the Session or TopicSession level. Creating it at the Session level allows the TemporaryTopic to participate in the same transaction with objects from the PTP domain. If a TemporaryTopic is created at the TopicSession, it will only be able participate in transactions with objects from the Pub/Sub domain.

Version:
   1.1 - February 2, 2002

Author:
   Mark Hapner, Rich Burridge, Kate Stout

See Also:
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>delete()</code></td>
<td>Deletes this temporary topic.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface `javax.jms.Topic`
- `getTopicName`
- `toString`

### Method Detail

#### public void `delete()` throws `JMSException`

**Throws**

- `JMSException`: JMS

**delete**

`void delete() throws JMSException`

Deletes this temporary topic. If there are existing subscribers still using it, a `JMSException` will be thrown.

**Throws:**

- `JMSException` - if the JMS provider fails to delete the temporary topic due to some internal error.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.soap  Interface Text

All Superinterfaces:
   CharacterData, Node, Text

public interface Text
extends Node, Text

Implements: Node, org.w3c.dom.Text

Text

A representation of a node whose value is text. A Text object may represent text that is content or text that is a comment.

Method Summary

<table>
<thead>
<tr>
<th>boolean isComment()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Retrieves whether this Text object represents a comment.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.xml.soap.Node

detachNode, getParentElement, getValue, recycleNode,
setParentElement, setValue

Methods inherited from interface org.w3c.dom.Text

getWholeText, isElementContentWhitespace, replaceWholeText,
splitText

Methods inherited from interface org.w3c.dom.CharacterData

appendData, deleteData, getData, getLength, insertData,
replaceData, setData, substringData
Method Detail

public boolean isComment()

    Text
    return

    Text    true    false

isComment

boolean isComment()

    Retrieves whether this Text object represents a comment.

    Returns:
    true if this Text object is a comment; false otherwise

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms  Interface  TextMessage

All Superinterfaces: Message

public interface TextMessage extends Message

Implements: Message

TextMessage  java.lang.String  Message

XML

TextMessage  MessageNotWriteableException  clearBody
version  1.1 - February 2, 2002

See  javax.jms.BytesMessage, javax.jms.MapMessage,
also  javax.jms.Message, javax.jms.ObjectMessage,
javax.jms.StreamMessage, java.lang.String

A TextMessage object is used to send a message containing a java.lang.String. It inherits from the Message interface and adds a text message body.

This message type can be used to transport text-based messages, including those with XML content.

When a client receives a TextMessage, it is in read-only mode. If a client attempts to write to the message at this point, a MessageNotWriteableException is thrown. If clearBody is called, the message can now be both read from and written to.

Version:
Field Summary

Fields inherited from interface javax.jms.Message
 DEFAULT DELIVERY MODE, DEFAULT PRIORITY, DEFAULT TIME TO LIVE

Method Summary

String getText()
    Gets the string containing this message's data.

void setText(String string)
    Sets the string containing this message's data.

Methods inherited from interface javax.jms.Message
public void setText(String string) throws JMSException

Throws:  
JMSException: JMS
MessageNotWriteableException:

setText

Sets the string containing this message's data.

Parameters:
string - the String containing the message's data

Throws:
JMSException - if the JMS provider fails to set the text due to some internal error.
MessageNotWriteableException - if the message is in read-only mode.

public String getText() throws JMSException

null

Throws:  
JMSException: JMS

getText

String getText()

throws JMSException

Gets the string containing this message's data. The default value is
null.

**Returns:**
the String containing the message's data

**Throws:**

`JMSException` - if the JMS provider fails to get the text due to some internal error.
**javax.ejb Interface TimedObject**

```java
public interface TimedObject
```

TimedObject Bean Bean Bean Bean

The TimedObject interface contains the callback method that is used to deliver timer expiration notifications. It is implemented by an entity bean or stateless session bean or message-driven bean class.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td><code>ejbTimeout(Timer timer)</code></td>
<td>Invoked by the EJB container upon timer expiration.</td>
</tr>
</tbody>
</table>

### Method Detail

**public void ejbTimeout(Timer timer)**

EJB

```java
timer
```

**ejbTimeout**

```java
void ejbTimeout(Timer timer)
```

Invoked by the EJB container upon timer expiration.

**Parameters:**

- `timer` - timer whose expiration caused this notification.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Annotation Type Timeout

@Target(value=METHOD)
@Retention(value=RUNTIME)
public @interface Timeout

Implements: Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)

Bean Bean Bean EJB

Designates a method on a stateless session bean class or message driven bean class that should receive EJB timer expirations for that bean.

Overview Package Tree Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Interface Timer

public interface Timer

Timer EJB Timer Service

The Timer interface contains information about a timer that was created through the EJB Timer Service.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong> cancel()</td>
</tr>
<tr>
<td>Cause the timer and all its associated expiration notifications to be cancelled.</td>
</tr>
<tr>
<td><strong>TimerHandle</strong> getHandle()</td>
</tr>
<tr>
<td>Get a serializable handle to the timer.</td>
</tr>
<tr>
<td><strong>Serializable</strong> getInfo()</td>
</tr>
<tr>
<td>Get the information associated with the timer at the time of creation.</td>
</tr>
<tr>
<td><strong>Date</strong> getNextTimeout()</td>
</tr>
<tr>
<td>Get the point in time at which the next timer expiration is scheduled to occur.</td>
</tr>
<tr>
<td><strong>long</strong> getTimeRemaining()</td>
</tr>
<tr>
<td>Get the number of milliseconds that will elapse before the next scheduled timer expiration.</td>
</tr>
</tbody>
</table>

Method Detail

public void cancel() throws IllegalStateException, NoSuchObjectLocalException, EJBException
Throws IllegalStateException:
Throws NoSuchObjectLocalException:
Throws EJBException:

cancel

void cancel() throws IllegalStateException,
NoSuchObjectLocalException,
EJBException

Cause the timer and all its associated expiration notifications to be cancelled.

Throws:
IllegalStateException - If this method is invoked while the instance is in a state that does not allow access to this method.
NoSuchObjectLocalException - If invoked on a timer that has expired or has been cancelled.
EJBException - If this method could not complete due to a system-level failure.

public long getTimeRemaining() throws
IllegalStateException, NoSuchObjectLocalException,
EJBException

return
Throws IllegalStateException:
Throws NoSuchObjectLocalException:
Throws EJBException:

getTimeRemaining

long getTimeRemaining()
Get the number of milliseconds that will elapse before the next scheduled timer expiration.

Returns:
the number of milliseconds that will elapse before the next scheduled timer expiration.

Throws:

 IllegalArgumentException - If this method is invoked while the instance is in a state that does not allow access to this method.
 NoSuchObjectLocalException - If invoked on a timer that has expired or has been cancelled.
 EJBException - If this method could not complete due to a system-level failure.

public java.util.Date getNextTimeout() throws IllegalArgumentException, NoSuchObjectLocalException, EJBException

return

Throws IllegalArgumentException:
Throws NoSuchObjectLocalException:
Throws EJBException:

getNextTimeout

Date getNextTimeout() throws IllegalArgumentException, NoSuchObjectLocalException, EJBException

Get the point in time at which the next timer expiration is scheduled to occur.
Returns:
the point in time at which the next timer expiration is scheduled to occur.

Throws:

- **IllegalStateException** - If this method is invoked while the instance is in a state that does not allow access to this method.
- **NoSuchObjectLocalException** - If invoked on a timer that has expired or has been cancelled.
- **EJBException** - If this method could not complete due to a system-level failure.

```java
public java.io.Serializable getInfo() throws
    IllegalStateException, NoSuchObjectLocalException, EJBException
```

- **return** Serializable info null null
- **Throws**  
  - **IllegalStateException**: 
  - **NoSuchObjectLocalException**: 
  - **EJBException**: 

GetInfo

```java
Serializable getINFO()
```

- **throws**  
  - **IllegalStateException**, **NoSuchObjectLocalException**, **EJBException**

Get the information associated with the timer at the time of creation.

**Returns:**
The Serializable object that was passed in at timer creation, or null if the info argument passed in at timer creation was null.

**Throws:**

- **IllegalStateException** - If this method is invoked while the instance is in a state that does not allow access to this method.
- **NoSuchObjectLocalException** - If invoked on a timer that has expired or has been cancelled.
- **EJBException** - If this method could not complete due to a system-level failure.
expired or has been cancelled.

**EJBException** - If this method could not complete due to a system-level failure.

```java
public TimerHandle getHandle() throws
IllegalStateException, NoSuchObjectLocalException, EJBException

return

Throws  
IllegalStateException:

Throws  NoSuchObjectLocalException:

Throws  EJBException:
```

**getHandle**

**TimerHandle** getHandle()

**Throws**

**IllegalStateException** - If this method is invoked while the instance is in a state that does not allow access to this method.

**NoSuchObjectLocalException** - If invoked on a timer that has expired or has been cancelled.

**EJBException** - If this method could not complete due to a system-level failure.

Get a serializable handle to the timer. This handle can be used at a later time to re-obtain the timer reference.

**Returns:**

a serializable handle to the timer.

**Throws:**

**IllegalStateException** - If this method is invoked while the instance is in a state that does not allow access to this method.

**NoSuchObjectLocalException** - If invoked on a timer that has expired or has been cancelled.

**EJBException** - If this method could not complete due to a system-level failure.
javax.ejb Interface TimerHandle

All Superinterfaces:  
Serializable

public interface TimerHandle
extends Serializable

Implements: java.io.Serializable

The TimerHandle interface is implemented by all EJB timer handles.

Method Summary

<table>
<thead>
<tr>
<th>getTimer()</th>
<th>Timer</th>
<th></th>
</tr>
</thead>
</table>
|            |       | Obtain a reference to the timer represented by this handle.

Method Detail

public Timer getTimer() throws IllegalStateException, NoSuchObjectLocalException, EJBException

  return

  Throws IllegalStateException:  
  Throws NoSuchObjectLocalException:  
  Throws EJBException:
getTimer

Timer getTimer()
throws IllegalStateException, NoSuchObjectLocalException, EJBException

Obtain a reference to the timer represented by this handle.

Returns:
  a reference to the timer represented by this handle.

Throws:
  IllegalState Exception - If this method is invoked while the instance is in a state that does not allow access to this method.
  NoSuchObjectLocalException - If invoked on a handle whose associated timer has expired or has been cancelled.
  EJBException - If this method could not complete due to a system-level failure.
javax.ejb Interface TimerService

public interface TimerService

TimerService Bean Timer Service EJB Timer Service Bean Bean Bean

The TimerService interface provides enterprise bean components with access to the container-provided Timer Service. The EJB Timer Service allows entity beans, stateless session beans, and message-driven beans to be registered for timer callback events at a specified time, after a specified elapsed time, or after a specified interval.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createTimer(Date initialExpiration, long intervalDuration, Serializable info)</code></td>
<td>Create an interval timer whose first expiration occurs at a given point in time and whose subsequent expirations occur after a specified interval.</td>
</tr>
<tr>
<td><code>createTimer(Date expiration, Serializable info)</code></td>
<td>Create a single-action timer that expires at a given point in time.</td>
</tr>
<tr>
<td><code>createTimer(long initialDuration, long intervalDuration, Serializable info)</code></td>
<td>Create an interval timer whose first expiration occurs after a specified duration, and whose subsequent expirations occur after a specified interval.</td>
</tr>
<tr>
<td><code>createTimer(long duration, Serializable info)</code></td>
<td>Create a single-action timer that expires after a specified duration.</td>
</tr>
<tr>
<td><code>getTimers()</code></td>
<td>Get all the active timers associated with this bean.</td>
</tr>
</tbody>
</table>
public Timer createTimer(long duration, java.io.Serializable info) throws IllegalArgumentException, IllegalStateException, EJBException

createTimer

Create a single-action timer that expires after a specified duration.

Parameters:
- duration - The number of milliseconds that must elapse before the timer expires.
- info - Application information to be delivered along with the timer expiration notification. This can be null.

Returns:
- The newly created Timer.

Throws:
- IllegalArgumentException - If duration is negative
- IllegalStateException - If this method is invoked while the instance is in a state that does not allow access to this method.
EJBException - If this method fails due to a system-level failure.

```java
public Timer createTimer(long initialDuration, long intervalDuration, java.io.Serializable info) throws IllegalArgumentException, IllegalStateException, EJBException
```

- `initialDuration` - The number of milliseconds that must elapse before the first timer expiration notification.
- `intervalDuration` - The number of milliseconds that must elapse between timer expiration notifications. Expiration notifications are scheduled relative to the time of the first expiration. If expiration is delayed (e.g. due to the interleaving of other method

Create an interval timer whose first expiration occurs after a specified duration, and whose subsequent expirations occur after a specified interval.

**Parameters:**
- `initialDuration` - The number of milliseconds that must elapse before the first timer expiration notification.
- `intervalDuration` - The number of milliseconds that must elapse between timer expiration notifications. Expiration notifications are scheduled relative to the time of the first expiration. If expiration is delayed (e.g. due to the interleaving of other method
calls on the bean) two or more expiration notifications may occur in close succession to "catch up".

info - Application information to be delivered along with the timer expiration. This can be null.

**Returns:**
The newly created Timer.

**Throws:**
- `IllegalArgumentException` - If initialDuration is negative, or intervalDuration is negative.
- `IllegalStateException` - If this method is invoked while the instance is in a state that does not allow access to this method.
- `EJBException` - If this method could not complete due to a system-level failure.

```java
public Timer createTimer(java.util.Date expiration, java.io.Serializable info) throws IllegalArgumentException, IllegalStateException, EJBException
```

Create a single-action timer that expires at a given point in time.
Parameters:
The expiration - The point in time at which the timer must expire.
info - Application information to be delivered along with the timer expiration notification. This can be null.

Returns:
The newly created Timer.

Throws:
IllegalStateException - If this method is invoked while the instance is in a state that does not allow access to this method.
EJBException - If this method could not complete due to a system-level failure.

```java
public Timer createTimer(java.util.Date initialExpiration, long intervalDuration, java.io.Serializable info) throws IllegalArgumentException, IllegalStateException, EJBException
```

```
initialExpiration
intervalDuration
info null
```

```
Throws IllegalArgumentException: initialExpiration null
initialExpiration.getTime() intervalDuration
Throws IllegalStateException: EJBException:
```

createTimer

```java
Timer createTimer(Date initialExpiration, long intervalDuration, Serializable info) throws IllegalArgumentException, IllegalStateException, EJBException
```
Create an interval timer whose first expiration occurs at a given point in time and whose subsequent expirations occur after a specified interval.

**Parameters:**
- `initialExpiration` - The point in time at which the first timer expiration must occur.
- `intervalDuration` - The number of milliseconds that must elapse between timer expiration notifications. Expiration notifications are scheduled relative to the time of the first expiration. If expiration is delayed (e.g. due to the interleaving of other method calls on the bean) two or more expiration notifications may occur in close succession to "catch up".
- `info` - Application information to be delivered along with the timer expiration. This can be null.

**Returns:**
The newly created Timer.

**Throws:**
- `IllegalArgumentException` - If `initialExpiration` is null, or `initialExpiration.getTime()` is negative, or `intervalDuration` is negative.
- `IllegalStateException` - If this method is invoked while the instance is in a state that does not allow access to this method.
- `EJBEException` - If this method could not complete due to a system-level failure.

```java
public java.util.Collection<E> getTimers() throws IllegalStateException, EJBEException
Bean
return javax.ejb.Timer
Throws IllegalArgumentException: IllegalStateException: 
Throws EJBEException:
```

gTimers

`Collection getTimers()`
Get all the active timers associated with this bean.

**Returns:**
A collection of javax.ejb.Timer objects.

**Throws:**
- `IllegalStateException` - If this method is invoked while the instance is in a state that does not allow access to this method.
- `EJBException` - If this method could not complete due to a system-level failure.
javax.management.j2ee.statistics Interface TimeStatistic

All Superinterfaces: Statistic

public interface TimeStatistic

extends Statistic

Implements: Statistic

Specifies standard timing measurements.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getCount()</strong></td>
</tr>
<tr>
<td>Long Number of times the operation was invoked since the beginning of this measurement.</td>
</tr>
</tbody>
</table>

| **getMaxTime()** |
| Long The maximum amount of time taken to complete one invocation of this operation since the beginning of this measurement. |

| **getMinTime()** |
| Long The minimum amount of time taken to complete one invocation of this operation since the beginning of this measurement. |

| **getTotalTime()** |
| Long This is the sum total of time taken to complete every invocation of this operation since the beginning of this measurement. |

Methods inherited from interface
javax.management.j2ee.statistics.Statistic
getDescription, getLastSampleTime, getName, getStartTime, getUnit

Method Detail

public long getCount()

getCount

long getCount()

Number of times the operation was invoked since the beginning of this measurement.

public long getMaxTime()

getMaxTime

long getMaxTime()

The maximum amount of time taken to complete one invocation of this operation since the beginning of this measurement.

public long getMinTime()

getMinTime
long `getMinTime()`

The minimum amount of time taken to complete one invocation of this operation since the beginning of this measurement.

---

`public long getTotalTime()`

`totalTime`

---

`getTotalTime`  

long `getTotalTime()`

This is the sum total of time taken to complete every invocation of this operation since the beginning of this measurement. Dividing `totalTime` by `count` will give you the average execution time for this operation.
javax.jms Interface Topic

All Superinterfaces:  
Destination

All Known Subinterfaces:  
TemporaryTopic

public interface Topic
extends Destination

Implements: Destination
Implemented by: TemporaryTopic

<table>
<thead>
<tr>
<th>Topic</th>
<th>JMS API</th>
<th>Destination</th>
<th>Topic</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MessageConsumer</td>
<td>MessageProducer</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Session.CreateConsumer(Destination destination)
- Session.CreateProducer(Destination destination)

/ (pub/sub) JMS API

pub/sub JMS API

version 1.1 - February 2, 2002

See createConsumer(Destination), createProducer(Destination), createTopic(String)

A Topic object encapsulates a provider-specific topic name. It is the way a client specifies the identity of a topic to JMS API methods. For those methods that use a Destination as a parameter, a Topic object may be used as an argument. For example, a Topic can be used to create a MessageConsumer and a MessageProducer by calling:

- Session.CreateConsumer(Destination destination)
Many publish/subscribe (pub/sub) providers group topics into hierarchies and provide various options for subscribing to parts of the hierarchy. The JMS API places no restriction on what a `Topic` object represents. It may be a leaf in a topic hierarchy, or it may be a larger part of the hierarchy.

The organization of topics and the granularity of subscriptions to them is an important part of a pub/sub application’s architecture. The JMS API does not specify a policy for how this should be done. If an application takes advantage of a provider-specific topic-grouping mechanism, it should document this. If the application is installed using a different provider, it is the job of the administrator to construct an equivalent topic architecture and create equivalent `Topic` objects.

**Version:**
1.1 - February 2, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
`Session.createConsumer(Destination)`,
`Session.createProducer(Destination)`,
`TopicSession.createTopic(String)`

### Method Summary

<table>
<thead>
<tr>
<th>String</th>
<th><code>getTopicName()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Gets the name of this topic.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th><code>toString()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Returns a string representation of this object.</td>
</tr>
</tbody>
</table>

### Method Detail

`public String getTopicName() throws JMSException`
getTopicName

String getTopicName() throws JMSException

Gets the name of this topic.

Clients that depend upon the name are not portable.

Returns:
the topic name

Throws:
JMSException - if the JMS provider implementation of Topic fails to return the topic name due to some internal error.

public String toString()

return

toString

String toString()

Returns a string representation of this object.

Overrides:
toString in class Object

Returns:
the provider-specific identity values for this topic
A TopicConnection object is an active connection to a publish/subscribe JMS provider. A client uses a TopicConnection object to create one or more TopicSession objects for producing and consuming messages.

A TopicConnection can be used to create a TopicSession, from which specialized topic-related objects can be created. A more general, and recommended approach is to use the Connection object.

The TopicConnection object should be used to support existing code.
Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createConnectionConsumer</code></td>
<td>Creates a connection consumer for this connection (optional operation).</td>
</tr>
<tr>
<td><code>createDurableConnectionConsumer</code></td>
<td>Create a durable connection consumer for this connection (optional operation).</td>
</tr>
<tr>
<td><code>createTopicSession</code></td>
<td>Creates a TopicSession object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.jms.Connection`

`close, createConnectionConsumer, createSession, getClientID, getExceptionListener, getMetaData, setClientID, setExceptionListener, start, stop`

Method Detail

```java
public TopicSession createTopicSession(boolean transacted, int acknowledgeMode) throws JMSException
    TopicSession
        transacted
```
acknowledgeMode Session.AUTO_ACKNOWLEDGE
Session.CLIENT_ACKNOWLEDGE
Session.DUPS_OK_ACKNOWLEDGE

return
Throws JMSException: TopicConnection
AUTO_ACKNOWLEDGE,
CLIENT_ACKNOWLEDGE,
DUPS_OK_ACKNOWLEDGE

See also
AUTO_ACKNOWLEDGE,
CLIENT_ACKNOWLEDGE,
DUPS_OK_ACKNOWLEDGE

createTopicSession

TopicSession createTopicSession(boolean transacted,
int acknowledgeMode)
throws JMSException

Creates a TopicSession object.

Parameters:
transacted - indicates whether the session is transacted
acknowledgeMode - indicates whether the consumer or the client
will acknowledge any messages it receives; ignored if the
session is transacted. Legal values are
Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, and
Session.DUPS_OK_ACKNOWLEDGE.

Returns:
a newly created topic session

Throws:
JMSException - if the TopicConnection object fails to create a
session due to some internal error or lack of support for the
specific transaction and acknowledgement mode.

See Also:
Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE,
Session.DUPS_OK_ACKNOWLEDGE

public ConnectionConsumer createConnectionConsumer(Topic topic, String
messageSelector, ServerSessionPool sessionPool, int
maxMessages) throws JMSException
JMS

**createConnectionConsumer**

```java
ConnectionConsumer createConnectionConsumer(Topic topic,
                                          String messageSelector,
                                          ServerSessionPool sessionPool,
                                          int maxMessages)

throws JMSException
```

Creates a connection consumer for this connection (optional operation). This is an expert facility not used by regular JMS clients.

**Parameters:**
- `topic` - the topic to access
- `messageSelector` - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- `sessionPool` - the server session pool to associate with this connection consumer
- `maxMessages` - the maximum number of messages that can be assigned to a server session at one time

**Returns:**
the connection consumer

**Throws:**
- `JMSException` - if the TopicConnection object fails to create a connection consumer due to some internal error or invalid
arguments for sessionPool and messageSelector.

InvalidDestinationException - if an invalid topic is specified.
InvalidSelectorException - if the message selector is invalid.

See Also:
  ConnectionConsumer

public ConnectionConsumer createDurableConnectionConsumer(Topic topic, String subscriptionName, String messageSelector, ServerSessionPool sessionPool, int maxMessages) throws JMSException

JMS

topic
subscriptionName
messageSelector null
sessionPool
maxMessages

return

Throws JMSException: TopicConnection sessionPool messageSelector

Throws InvalidDestinationException:

Throws InvalidSelectorException:

See also javax.jms.ConnectionConsumer

createDurableConnectionConsumer

ConnectionConsumer createDurableConnectionConsumer(Topic topic, String subscriptionName, String messageSelector, ServerSessionPool sessionPool, int maxMessages) throws JMSException

Create a durable connection consumer for this connection (optional operation). This is an expert facility not used by regular JMS clients.
Specified by:
createDurableConnectionConsumer in interface Connection

Parameters:
- topic - the topic to access
- subscriptionName - durable subscription name
- messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- sessionPool - the server session pool to associate with this durable connection consumer
- maxMessages - the maximum number of messages that can be assigned to a server session at one time

Returns:
the durable connection consumer

Throws:
- JMSEexception - if the TopicConnection object fails to create a connection consumer due to some internal error or invalid arguments for sessionPool and messageSelector.
- InvalidDestinationException - if an invalid topic is specified.
- InvalidSelectorException - if the message selector is invalid.

See Also:
ConnectionConsumer
javax.jms Interface TopicConnectionFactory

All Superinterfaces:  
   ConnectionFactory

All Known Subinterfaces:  
   XATopicConnectionFactory

public interface TopicConnectionFactory  
extends ConnectionFactory

Implements: ConnectionFactory  
Implemented by: XATopicConnectionFactory

      TopicConnectionFactory / JMS      TopicConnection
TopicConnectionFactory       TopicConnection       TopicConnectionFactory
      TopicConnectionFactory

version 1.1 - February 2, 2002      en
See also javax.jms.ConnectionFactory

A client uses a TopicConnectionFactory object to create TopicConnection objects with a publish/subscribe JMS provider.

A TopicConnectionFactory can be used to create a TopicConnection, from which specialized topic-related objects can be created. A more general, and recommended approach is to use the ConnectionFactory object.

The TopicConnectionFactory object should be used to support existing code.

Version:
1.1 - February 2, 2002

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
ConnectionFactory

---

## Method Summary

<table>
<thead>
<tr>
<th>TopicConnection</th>
<th>createTopicConnection()</th>
<th>Creates a topic connection with the default user identity.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>createTopicConnection(String userName, String password)</td>
<td>Creates a topic connection with the specified user identity.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.ConnectionFactory
createConnection, createConnection

---

## Method Detail

public TopicConnection createTopicConnection() throws JMSException

```
return
```

Throws

- JMSException: JMS
- JMSSecurityException:

createTopicConnection

```
TopicConnection createTopicConnection() throws JMSException
```
Creates a topic connection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

**Returns:**
- a newly created topic connection

**Throws:**
- `JMSException` - if the JMS provider fails to create a topic connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

```java
public TopicConnection createTopicConnection(String userName, String password) throws JMSException
```

**createTopicConnection**

```java
TopicConnection createTopicConnection(String userName, String password) throws JMSException
```

Creates a topic connection with the specified user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

**Parameters:**
- `userName` - the caller's user name
- `password` - the caller's password

**Returns:**
- a newly created topic connection
Throws:

- `JMSException` - if the JMS provider fails to create a topic connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jms Interface TopicPublisher

All Superinterfaces:
   MessageProducer

public interface TopicPublisher
   extends MessageProducer

Implements: MessageProducer

   TopicPublisher   TopicPublisher
   Topic Publisher  TopicPublisher publish
   java.lang.UnsupportedOperationException

   Topic Publisher  Topic publish
   java.lang.UnsupportedOperationException

   publish   publish

   JMSDestination  JMSDeliveryMode JMSExpiration JMSPriority JMSMessageID
   JMSTimeStamp     publish
   MessageProducer.setDisableMessageID
   MessageProducer.setDisableMessageTimestamp   publish
   JMSMessageID JMSTimeStamp

   MessageProducer  TopicPublisher  MessageProducer
   TopicPublisher   TopicPublisher

   TopicPublisher  MessageProducer  MessageProducer send
   send   publish

version 1.1 February 2, 2002
See also createProducer(Destination), createPublisher(Topic)
A client uses a TopicPublisher object to publish messages on a topic. A TopicPublisher object is the publish-subscribe form of a message producer.

Normally, the Topic is specified when a TopicPublisher is created. In this case, an attempt to use the publish methods for an unidentified TopicPublisher will throw a java.lang.UnsupportedOperationException.

If the TopicPublisher is created with an unidentified Topic, an attempt to use the publish methods that assume that the Topic has been identified will throw a java.lang.UnsupportedOperationException.

During the execution of its publish method, a message must not be changed by other threads within the client. If the message is modified, the result of the publish is undefined.

After publishing a message, a client may retain and modify it without affecting the message that has been published. The same message object may be published multiple times.

The following message headers are set as part of publishing a message: JMSDestination, JMSDeliveryMode, JMSExpiration, JMSPriority, JMSMessageID and JMSTimeStamp. When the message is published, the values of these headers are ignored. After completion of the publish, the headers hold the values specified by the method publishing the message. It is possible for the publish method not to set JMSMessageID and JMSTimeStamp if the setting of these headers is explicitly disabled by the MessageProducer.setDisableMessageID or MessageProducer.setDisableMessageTimestamp method.

Creating a MessageProducer provides the same features as creating a TopicPublisher. A MessageProducer object is recommended when creating new code. The TopicPublisher is provided to support existing code.

Because TopicPublisher inherits from MessageProducer, it inherits the send methods that are a part of the MessageProducer interface. Using the send methods will have the same effect as using the publish methods: they are functionally the same.
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getTopic()</code></td>
<td>Gets the topic associated with this <code>TopicPublisher</code>.</td>
</tr>
<tr>
<td><code>publish(Message message)</code></td>
<td>Publishes a message to the topic.</td>
</tr>
<tr>
<td><code>publish(Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Publishes a message to the topic, specifying delivery mode, priority, and time to live.</td>
</tr>
<tr>
<td><code>publish(Topic topic, Message message)</code></td>
<td>Publishes a message to a topic for an unidentified message producer.</td>
</tr>
<tr>
<td><code>publish(Topic topic, Message message, int deliveryMode, int priority, long timeToLive)</code></td>
<td>Publishes a message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface `javax.jms.MessageProducer`**

- `close`
- `getDeliveryMode`
- `getDestination`
- `getDisableMessageID`
- `getDisableMessageTimestamp`
- `getPriority`
- `getTimeToLive`
- `send`
- `send`
- `send`
- `send`
- `setDeliveryMode`
- `setDisableMessageID`
- `setDisableMessageTimestamp`
- `setPriority`
- `setTimeToLive`

### Method Detail

```java
public Topic getTopic() throws JMSException
```
### getTopic

```java
public Topic getTopic() throws JMSException {
    // Implementation
}
```

**Returns:**
- this publisher's topic

**Throws:**
- `JMSException` - if the JMS provider fails to get the topic for this `TopicPublisher` due to some internal error.

### publish

```java
public void publish(Message message) throws JMSException {
    // Implementation
}
```

**Throws:**
- `JMSException`: JMS
- `MessageFormatException`
- `InvalidDestinationException`
- `UnsupportedOperationException`: TopicPublisher

**See also:**
- `getDeliveryMode()`, `getTimeToLive()`, `getPriority()`

**Publishes a message to the topic. Uses the `TopicPublisher's default delivery mode, priority, and time to live.`**
Parameters:
message - the message to publish

Throws:

- **JMSException** - if the JMS provider fails to publish the message due to some internal error.
- **MessageFormatException** - if an invalid message is specified.
- **InvalidDestinationException** - if a client uses this method with a TopicPublisher with an invalid topic.
- **UnsupportedOperationException** - if a client uses this method with a TopicPublisher that did not specify a topic at creation time.

See Also:

- `MessageProducer.getDeliveryMode()`
- `MessageProducer.getTimeToLive()`
- `MessageProducer.getPriority()`

public void publish(Message message, int deliveryMode, int priority, long timeToLive) throws JMSException

message
deliveryMode
priority
timeToLive

Throws  JMSException: JMS

Throws  MessageFormatException:

Throws  InvalidDestinationException: Topic

Publish

do void publish(Message message, int deliveryMode, int priority, long timeToLive)

throws JMSException
Publishes a message to the topic, specifying delivery mode, priority, and time to live.

**Parameters:**
- `message` - the message to publish
- `deliveryMode` - the delivery mode to use
- `priority` - the priority for this message
- `timeToLive` - the message's lifetime (in milliseconds)

**Throws:**
- `JMSException` - if the JMS provider fails to publish the message due to some internal error.
- `MessageFormatException` - if an invalid message is specified.
- `InvalidDestinationException` - if a client uses this method with a `TopicPublisher` with an invalid topic.
- `UnsupportedOperationException` - if a client uses this method with a `TopicPublisher` that did not specify a topic at creation time.

```java
public void publish(Topic topic, Message message) throws JMSException
```

**JMS API**
- `topic`
- `message`

**Throws**
- `JMSException`: JMS
- `MessageFormatException`: 
- `InvalidDestinationException`: 

**See also**
- `getDeliveryMode()`, `getTimeToLive()`, `getPriority()`

```java
void publish(Topic topic,
            Message message)
```

**Throws**
- `JMSException`: 

Publishes a message to a topic for an unidentified message producer. Uses the TopicPublisher's default delivery mode, priority, and time to live.

Typically, a message producer is assigned a topic at creation time; however, the JMS API also supports unidentified message producers, which require that the topic be supplied every time a message is published.

**Parameters:**
- `topic` - the topic to publish this message to
- `message` - the message to publish

**Throws:**
- `JMSException` - if the JMS provider fails to publish the message due to some internal error.
- `MessageFormatException` - if an invalid message is specified.
- `InvalidDestinationException` - if a client uses this method with an invalid topic.

**See Also:**
- `MessageProducer.getDeliveryMode()`
- `MessageProducer.getTimeToLive()`
- `MessageProducer.getPriority()`

```java
public void publish(Topic topic, Message message, int deliveryMode, int priority, long timeToLive) throws JMSException
```

**JMS API**

- `topic`
- `message`
- `deliveryMode`
- `priority`
- `timeToLive`

**Throws**
- `JMSException`: JMS
- `MessageFormatException`
- `InvalidDestinationException`
publish

void publish(Topic topic,
             Message message,
             int deliveryMode,
             int priority,
             long timeToLive)
        throws JMSException

Publishes a message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.

Typically, a message producer is assigned a topic at creation time; however, the JMS API also supports unidentified message producers, which require that the topic be supplied every time a message is published.

Parameters:
- topic - the topic to publish this message to
- message - the message to publish
- deliveryMode - the delivery mode to use
- priority - the priority for this message
- timeToLive - the message's lifetime (in milliseconds)

Throws:
- JMSException - if the JMS provider fails to publish the message due to some internal error.
- MessageFormatException - if an invalid message is specified.
- InvalidDestinationException - if a client uses this method with an invalid topic.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
javax.jms Class TopicRequestor

java.lang.Object
   └─javax.jms.TopicRequestor

    public class TopicRequestor
       extends Object

TopicRequestor
TopicRequestor TopicSession Topic TemporaryTopic request

/JMS

   version 1.0 - 8 July 1998
   See also javax.jms.QueueRequestor

The TopicRequestor helper class simplifies making service requests.

The TopicRequestor constructor is given a non-transacted TopicSession
and a destination Topic. It creates a TemporaryTopic for the responses
and provides a request method that sends the request message and
waits for its reply.

This is a basic request/reply abstraction that should be sufficient for most
uses. JMS providers and clients are free to create more sophisticated
versions.

Version:
   1.0 - 8 July 1998

Author:
   Mark Hapner, Rich Burridge

See Also:
   QueueRequestor
Constructor Summary

```java
public TopicRequestor(TopicSession session, Topic topic)

Constructor for the TopicRequestor class.
```

Method Summary

```java
void close()

Closes the TopicRequestor and its session.
```

```java
Message request(Message message)

Sends a request and waits for a reply.
```

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`
Constructor for the TopicRequestor class.

This implementation assumes the session parameter to be non-transacted, with a delivery mode of either AUTO_ACKNOWLEDGE or DUPS_OK_ACKNOWLEDGE.

Parameters:
- `session`: the TopicSession the topic belongs to
- `topic`: the topic to perform the request/reply call on

Throws:
- `JMSException`: if the JMS provider fails to create the TopicRequestor due to some internal error.
- `InvalidDestinationException`: if an invalid topic is specified.

### Method Detail

#### public `Message` request(`Message` message) throws `JMSException`

Sends a request and waits for a reply. The temporary topic is used for the JMSReplyTo destination; the first reply is returned, and any following replies are discarded.

Parameters:
- `message`: the message to send
Returns:
the reply message

Throws:
   JMSException - if the JMS provider fails to complete the request due to some internal error.

public void close() throws JMSException

TopicRequestor

Java

TopicRequestor   TopicSession
Throws   JMSException: JMS

Closes the TopicRequestor and its session.

Since a provider may allocate some resources on behalf of a TopicRequestor outside the Java virtual machine, clients should close them when they are not needed. Relying on garbage collection to eventually reclaim these resources may not be timely enough.

Note that this method closes the TopicSession object passed to the TopicRequestor constructor.

Throws:
   JMSException - if the JMS provider fails to close the TopicRequestor due to some internal error.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.jms Interface TopicSession**

**All Superinterfaces:**  [Runnable, Session]

```java
public interface TopicSession
    extends Session

Implements: Session
```

**TopicSession** TopicPublisher TopicSubscriber TemporaryTopic

**Pub/Sub** Session

```java
TopicSession createBrowser
createQueue
createTemporaryQueue
```

**IllegalStateException**

- createBrowser
- createQueue
- createTemporaryQueue

**version** 1.1 - April 9, 2002

**See**  [javax.jms.Session, createSession(boolean, int), createTopicSession(boolean, int), getTopicSession()]

**A TopicSession object provides methods for creating TopicPublisher, TopicSubscriber, and TemporaryTopic objects. It also provides a method for deleting its client's durable subscribers.**

**A TopicSession is used for creating Pub/Sub specific objects. In general, use the Session object, and use TopicSession only to support existing code. Using the Session object simplifies the programming model, and allows transactions to be used across the two messaging domains.**
A `TopicSession` cannot be used to create objects specific to the point-to-point domain. The following methods inherit from `Session`, but must throw an `IllegalStateException` if used from `TopicSession`:

- `createBrowser`
- `createQueue`
- `createTemporaryQueue`

**Version:**
1.1 - April 9, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
- `Session`
- `Connection.createSession(boolean, int)`
- `TopicConnection.createTopicSession(boolean, int)`
- `XATopicSession.getTopicSession()`

### Field Summary

**Fields inherited from interface `javax.jms.Session`**

- `AUTO_ACKNOWLEDGE`
- `CLIENT_ACKNOWLEDGE`
- `DUPS_OK_ACKNOWLEDGE`
- `SESSION_TRANSACTED`

### Method Summary

- **`createDurableSubscriber`**
  - Method: `createDurableSubscriber(Topic topic, String name)`
  - Creates a durable subscriber to the specified topic.

- **`createDurableSubscriber`**
  - Method: `createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal)`
  - Creates a durable subscriber to the specified topic, using a message selector or specifying whether messages published by its own connection should be delivered to it.

- **`createPublisher`**
  - Method: `createPublisher(Topic topic)`
  - Creates a publisher for the specified topic.

- **`createSubscriber`**
  - Method: `createSubscriber(Topic topic)`
  - Creates a nondurable subscriber to the specified topic.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createSubscriber(Topic topic, String messageSelector, boolean noLocal)</code></td>
<td>Creates a nondurable subscriber to the specified topic, using a message selector or specifying whether messages published by its own connection should be delivered to it.</td>
</tr>
<tr>
<td><code>createTemporaryTopic()</code></td>
<td>Creates a <code>TemporaryTopic</code> object.</td>
</tr>
<tr>
<td><code>createTopic(String topicName)</code></td>
<td>Creates a topic identity given a <code>Topic</code> name.</td>
</tr>
<tr>
<td><code>unsubscribe(String name)</code></td>
<td>Unsubscribes a durable subscription that has been created by a client.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface javax.jms.Session**

- `close`, `commit`, `createBrowser`, `createBrowser`, `createBytesMessage`, `createConsumer`, `createConsumer`, `createConsumer`, `createMapMessage`, `createMessage`, `createObjectMessage`, `createObjectMessage`, `createProducer`, `createQueue`, `createStreamMessage`, `createTemporaryQueue`, `createTextMessage`, `createTextMessage`, `getAcknowledgeMode`, `getMessageListener`, `getTransacted`, `recover`, `rollback`, `run`, `setMessageListener`
createTopic

**Topic** createTopic(String topicName)

**Throws:** JMSException

Creates a topic identity given a Topic name.

This facility is provided for the rare cases where clients need to dynamically manipulate topic identity. This allows the creation of a topic identity with a provider-specific name. Clients that depend on this ability are not portable.

Note that this method is not for creating the physical topic. The physical creation of topics is an administrative task and is not to be initiated by the JMS API. The one exception is the creation of temporary topics, which is accomplished with the createTemporaryTopic method.

**Specified by:**
createTopic in interface Session

**Parameters:**
topicName - the name of this Topic

**Returns:**
a Topic with the given name

**Throws:**
JMSException - if the session fails to create a topic due to some internal error.

---

public **TopicSubscriber** createSubscriber(Topic topic)

**Throws:** JMSException

**Returns:**
TopicSubscriber
createSubscriber

`TopicSubscriber createSubscriber(Topic topic) throws JMSException`

Creates a nondurable subscriber to the specified topic.

A client uses a `TopicSubscriber` object to receive messages that have been published to a topic.

Regular `TopicSubscriber` objects are not durable. They receive only messages that are published while they are active.

In some cases, a connection may both publish and subscribe to a topic. The subscriber `NoLocal` attribute allows a subscriber to inhibit the delivery of messages published by its own connection. The default value for this attribute is false.

**Parameters:**

- `topic` - the Topic to subscribe to

**Throws:**

- `JMSException` - if the session fails to create a subscriber due to some internal error.
- `InvalidDestinationException` - if an invalid topic is specified.

```java
public TopicSubscriber createSubscriber(Topic topic, String messageSelector, boolean noLocal) throws JMSException
```
createSubscriber

```
TopicSubscriber createSubscriber(Topic topic, String messageSelector, boolean noLocal) throws JMSException
```

Creates a nondurable subscriber to the specified topic, using a message selector or specifying whether messages published by its own connection should be delivered to it.

A client uses a TopicSubscriber object to receive messages that have been published to a topic.

Regular TopicSubscriber objects are not durable. They receive only messages that are published while they are active.

Messages filtered out by a subscriber's message selector will never be delivered to the subscriber. From the subscriber's perspective, they do not exist.
In some cases, a connection may both publish and subscribe to a topic. The subscriber NoLocal attribute allows a subscriber to inhibit the delivery of messages published by its own connection. The default value for this attribute is false.

**Parameters:**

- `topic` - the Topic to subscribe to
- `messageSelector` - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- `noLocal` - if set, inhibits the delivery of messages published by its own connection

**Throws:**

- `JMSException` - if the session fails to create a subscriber due to some internal error.
- `InvalidDestinationException` - if an invalid topic is specified.
- `InvalidSelectorException` - if the message selector is invalid.

```java
public TopicSubscriber createDurableSubscriber(Topic topic, String name) throws JMSException
```

```
TopicSubscriber JMS

TopicSubscriber

/ TopicSubscriber

  NoLocal false

  topic Topic

  name

  Throws JMSException:

  Throws InvalidDestinationException:
```
createDurableSubscriber

TopicSubscriber createDurableSubscriber(Topic topic, String name)
throws JMSException

Creates a durable subscriber to the specified topic.

If a client needs to receive all the messages published on a topic, including the ones published while the subscriber is inactive, it uses a durable TopicSubscriber. The JMS provider retains a record of this durable subscription and insures that all messages from the topic’s publishers are retained until they are acknowledged by this durable subscriber or they have expired.

Sessions with durable subscribers must always provide the same client identifier. In addition, each client must specify a name that uniquely identifies (within client identifier) each durable subscription it creates. Only one session at a time can have a TopicSubscriber for a particular durable subscription.

A client can change an existing durable subscription by creating a durable TopicSubscriber with the same name and a new topic and/or message selector. Changing a durable subscriber is equivalent to unsubscribing (deleting) the old one and creating a new one.

In some cases, a connection may both publish and subscribe to a topic. The subscriber NoLocal attribute allows a subscriber to inhibit the delivery of messages published by its own connection. The default value for this attribute is false.

Specified by:
createDurableSubscriber in interface Session

Parameters:
- topic - the non-temporary Topic to subscribe to
- name - the name used to identify this subscription

Throws:
- JMSException - if the session fails to create a subscriber due to some internal error.
- InvalidDestinationException - if an invalid topic is specified.
public TopicSubscriber createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException

createDurableSubscriber

createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException

createDurableSubscriber(Topic topic, String name, String messageSelector, boolean noLocal) throws JMSException

Creates a durable subscriber to the specified topic, using a message selector or specifying whether messages published by its own connection should be delivered to it.

If a client needs to receive all the messages published on a topic, including the ones published while the subscriber is inactive, it uses a durable TopicSubscriber. The JMS provider retains a record of this durable subscription and insures that all messages from the topic's publishers are retained until they are acknowledged by this durable
subscriber or they have expired.

Sessions with durable subscribers must always provide the same client identifier. In addition, each client must specify a name which uniquely identifies (within client identifier) each durable subscription it creates. Only one session at a time can have a TopicSubscriber for a particular durable subscription. An inactive durable subscriber is one that exists but does not currently have a message consumer associated with it.

A client can change an existing durable subscription by creating a durable TopicSubscriber with the same name and a new topic and/or message selector. Changing a durable subscriber is equivalent to unsubscribing (deleting) the old one and creating a new one.

Specified by:
createDurableSubscriber in interface Session

Parameters:
- topic - the non-temporary Topic to subscribe to
- name - the name used to identify this subscription
- messageSelector - only messages with properties matching the message selector expression are delivered. A value of null or an empty string indicates that there is no message selector for the message consumer.
- noLocal - if set, inhibits the delivery of messages published by its own connection

Throws:
- JMSEException - if the session fails to create a subscriber due to some internal error.
- InvalidDestinationException - if an invalid topic is specified.
- InvalidSelectorException - if the message selector is invalid.

```java
public TopicPublisher createPublisher(Topic topic) throws JMSException
```
createPublisher

**TopicPublisher** createPublisher(**Topic** topic) throws **JMSException**

Creates a publisher for the specified topic.

A client uses a *TopicPublisher* object to publish messages on a topic. Each time a client creates a *TopicPublisher* on a topic, it defines a new sequence of messages that have no ordering relationship with the messages it has previously sent.

**Parameters:**
- `topic` - the *Topic* to publish to, or null if this is an unidentified producer

**Throws:**
- **JMSException** - if the session fails to create a publisher due to some internal error.
- **InvalidDestinationException** - if an invalid topic is specified.

---

public **TemporaryTopic** createTemporaryTopic() throws **JMSException**

**TemporaryTopic** createTemporaryTopic() throws **JMSException**

createTemporaryTopic

**TemporaryTopic** createTemporaryTopic() throws **JMSException**
Creates a TemporaryTopic object. Its lifetime will be that of the TopicConnection unless it is deleted earlier.

**Specified by:**
`createTemporaryTopic` in interface `Session`

**Returns:**
a temporary topic identity

**Throws:**
`JMSException` - if the session fails to create a temporary topic due to some internal error.

```java
public void unsubscribe(String name) throws JMSException
```

**Throws**
- `JMSException`
- `InvalidDestinationException`:

```java
unsubscribe(String name)
```

Unsubscribes a durable subscription that has been created by a client.

This method deletes the state being maintained on behalf of the subscriber by its provider.

It is erroneous for a client to delete a durable subscription while there is an active TopicSubscriber for the subscription, or while a
consumed message is part of a pending transaction or has not been acknowledged in the session.

**Specified by:**
unsubscribe in interface Session

**Parameters:**
name - the name used to identify this subscription

**Throws:**
JMSException - if the session fails to unsubscribe to the durable subscription due to some internal error.
InvalidDestinationException - if an invalid subscription name is specified.
public interface TopicSubscriber
extends MessageConsumer

Implements: MessageConsumer

Session.createConsumer
TopicSession TopicSubscriber

Session.CreateDurableSubscriber

See also createConsumer, createDurableSubscriber, javax.jms.TopicSession, createSubscriber, javax.jms.MessageConsumer
A client uses a TopicSubscriber object to receive messages that have been published to a topic. A TopicSubscriber object is the publish/subscribe form of a message consumer. A MessageConsumer can be created by using Session.createConsumer.

A TopicSession allows the creation of multiple TopicSubscriber objects per topic. It will deliver each message for a topic to each subscriber eligible to receive it. Each copy of the message is treated as a completely separate message. Work done on one copy has no effect on the others; acknowledging one does not acknowledge the others; one message may be delivered immediately, while another waits for its subscriber to process messages ahead of it.

Regular TopicSubscriber objects are not durable. They receive only messages that are published while they are active.

Messages filtered out by a subscriber's message selector will never be delivered to the subscriber. From the subscriber's perspective, they do not exist.

In some cases, a connection may both publish and subscribe to a topic. The subscriber NoLocal attribute allows a subscriber to inhibit the delivery of messages published by its own connection.

If a client needs to receive all the messages published on a topic, including the ones published while the subscriber is inactive, it uses a durable TopicSubscriber. The JMS provider retains a record of this durable subscription and insures that all messages from the topic's publishers are retained until they are acknowledged by this durable subscriber or they have expired.

Sessions with durable subscribers must always provide the same client identifier. In addition, each client must specify a name that uniquely identifies (within client identifier) each durable subscription it creates. Only one session at a time can have a TopicSubscriber for a particular durable subscription.

A client can change an existing durable subscription by creating a
durable TopicSubscriber with the same name and a new topic and/or message selector. Changing a durable subscription is equivalent to unsubscribing (deleting) the old one and creating a new one.

The unsubscribe method is used to delete a durable subscription. The unsubscribe method can be used at the Session or TopicSession level. This method deletes the state being maintained on behalf of the subscriber by its provider.

Creating a MessageConsumer provides the same features as creating a TopicSubscriber. To create a durable subscriber, use of Session.CreateDurableSubscriber is recommended. The TopicSubscriber is provided to support existing code.

Version:
  1.1 - February 2, 2002

Author:
  Mark Hapner, Rich Burridge, Kate Stout

See Also:
  Session.createConsumer(javax.jms.Destination),
  Session.createDurableSubscriber(javax.jms.Topic,
  java.lang.String), TopicSession,
  TopicSession.createSubscriber(javax.jms.Topic), MessageConsumer

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean getNoLocal()</td>
<td>Gets the NoLocal attribute for this subscriber.</td>
</tr>
<tr>
<td>Topic getTopic()</td>
<td>Gets the Topic associated with this subscriber.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface javax.jms.MessageConsumer

close, getMessageListener, getMessageSelector, receive, receive, receiveNoWait, setMessageListener

---

### Method Detail
public Topic getTopic() throws JMSException

    Topic
    return
    Throws    JMSException: JMS

getTopic

    Topic getTopic()
    throws JMSException

    Gets the Topic associated with this subscriber.

    Returns:  
               this subscriber's Topic

    Throws:
               JMSException - if the JMS provider fails to get the topic for this topic subscriber due to some internal error.

public boolean getNoLocal() throws JMSException

    NoLocal false
    return  true
    Throws  JMSException: JMS

getNoLocal

    boolean getNoLocal()
    throws JMSException

    Gets the NoLocal attribute for this subscriber. The default value for this attribute is false.

    Returns:  
               true if locally published messages are being inhibited

    Throws:
**JMSException** - if the JMS provider fails to get the `NoLocal` attribute for this topic subscriber due to some internal error.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The Transaction interface allows operations to be performed against the transaction in the target Transaction object. A Transaction object is created corresponding to each global transaction creation. The Transaction object can be used for resource enlistment, synchronization registration, transaction completion, and status query operations.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit()</td>
<td>Complete the transaction represented by this Transaction object.</td>
</tr>
<tr>
<td>delistResource(XAResource xaRes, int flag)</td>
<td>Disassociate the resource specified from the transaction associated with the target Transaction object.</td>
</tr>
<tr>
<td>enlistResource(XAResource xaRes)</td>
<td>Enlist the resource specified with the transaction associated with the target Transaction object.</td>
</tr>
<tr>
<td>getStatus()</td>
<td>Obtain the status of the transaction associated with the target Transaction object.</td>
</tr>
<tr>
<td>registerSynchronization(Synchronization sync)</td>
<td>Register a synchronization object for the transaction currently associated with the target object.</td>
</tr>
<tr>
<td>rollback()</td>
<td>Rollback the transaction represented by this Transaction object.</td>
</tr>
</tbody>
</table>

javax.transaction Interface Transaction

public interface Transaction

Transaction Transaction Transaction Transaction
### Method Detail

**public void commit() throws** *RollbackException, HeuristicMixedException, HeuristicRollbackException, SecurityException, IllegalStateException, SystemException*

<table>
<thead>
<tr>
<th>Throws</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>RollbackException</strong></td>
<td>Thrown to indicate that the transaction has been rolled back rather than committed.</td>
</tr>
<tr>
<td><strong>HeuristicMixedException</strong></td>
<td>Thrown to indicate that a heuristic</td>
</tr>
<tr>
<td><strong>HeuristicRollbackException</strong></td>
<td>Thrown to indicate that a heuristic</td>
</tr>
</tbody>
</table>

**commit**

**void commit() throws** *RollbackException, HeuristicMixedException, HeuristicRollbackException, SecurityException, IllegalStateException, SystemException*

Complete the transaction represented by this Transaction object.

**Throws:**

- **RollbackException** - Thrown to indicate that the transaction has been rolled back rather than committed.
- **HeuristicMixedException** - Thrown to indicate that a heuristic
decision was made and that some relevant updates have been committed while others have been rolled back. 

**HeuristicRollbackException** - Thrown to indicate that a heuristic decision was made and that all relevant updates have been rolled back.

**SecurityException** - Thrown to indicate that the thread is not allowed to commit the transaction.

**IllegalStateException** - Thrown if the transaction in the target object is inactive.

**SystemException** - Thrown if the transaction manager encounters an unexpected error condition.

---

```java
public boolean delistResource(XAResource xaRes, int flag) throws IllegalStateException, SystemException
```

Transaction

- **xaRes**: XAResource
- **flag**: TMSUCCESS, TMSUSPEND, TMFAIL

**Throws**:
- **IllegalStateException**: 
- **SystemException**: 

**return**:  
```
true  false
```

---

**delistResource**

```java
boolean delistResource(XAResource xaRes, int flag) throws IllegalStateException, SystemException
```

Disassociate the resource specified from the transaction associated with the target Transaction object.

**Parameters:**
- **xaRes**: The XAResource object associated with the resource (connection).
- **flag**: One of the values of TMSUCCESS, TMSUSPEND, or TMFAIL.
Returns: 
true if the resource was delisted successfully; otherwise false.

Throws:
IllegalStateException - Thrown if the transaction in the target object is inactive.
SystemException - Thrown if the transaction manager encounters an unexpected error condition.

```java
public boolean enlistResource(XAResource xaRes) throws RollbackException, IllegalStateException, SystemException {
    // Enlist the resource specified with the transaction associated with the target Transaction object.
    xaRes XAResource
    return true false

    Throws RollbackException:
    Throws IllegalStateException:
    Throws SystemException:
}
```

enlistResource

boolean enlistResource(XAResource xaRes)
throws RollbackException, IllegalStateException, SystemException

Enlist the resource specified with the transaction associated with the target Transaction object.

Parameters:
xaRes - The XAResource object associated with the resource (connection).

Returns: 
true if the resource was enlisted successfully; otherwise false.

Throws: 
RollbackException - Thrown to indicate that the transaction has been marked for rollback only.
**IllegalStateException** - Thrown if the transaction in the target object is in the prepared state or the transaction is inactive.

**SystemException** - Thrown if the transaction manager encounters an unexpected error condition.

```java
public int getStatus() throws SystemException
Transaction

return Status.NoTransaction

Throws SystemException:
```

**getStatus**

```java
int getStatus()
throws SystemException

Obtain the status of the transaction associated with the target Transaction object.

Returns:
The transaction status. If no transaction is associated with the target object, this method returns the Status.NoTransaction value.

Throws:
SystemException - Thrown if the transaction manager encounters an unexpected error condition.
```

```java
public void registerSynchronization(Synchronization sync) throws RollbackException, IllegalStateException, SystemException

beforeCompletion afterCompletion

 sync Synchronization

Throws RollbackException:

Throws IllegalStateException:

Throws SystemException:
```
registerSynchronization

```java
void registerSynchronization(Synchronization sync)
    throws RollbackException,
            IllegalStateException,
            SystemException
```

Register a synchronization object for the transaction currently associated with the target object. The transaction manager invokes the beforeCompletion method prior to starting the two-phase transaction commit process. After the transaction is completed, the transaction manager invokes the afterCompletion method.

**Parameters:**
- `sync` - The Synchronization object for the transaction associated with the target object.

**Throws:**
- `RollbackException` - Thrown to indicate that the transaction has been marked for rollback only.
- `IllegalStateException` - Thrown if the transaction in the target object is in the prepared state or the transaction is inactive.
- `SystemException` - Thrown if the transaction manager encounters an unexpected error condition.

```
public void rollback() throws IllegalStateException,
                        SystemException
```

**Transaction**

- Throws `IllegalStateException`:
- Throws `SystemException`:

**rollback**

```java
void rollback()
    throws RollbackException,
            SystemException
```
Rollback the transaction represented by this Transaction object.

Throws:

- `IllegalStateException` - Thrown if the transaction in the target object is in the prepared state or the transaction is inactive.
- `SystemException` - Thrown if the transaction manager encounters an unexpected error condition.

---

public void setRollbackOnly() throws `IllegalStateException`, `SystemException`

Throws

- `IllegalStateException`:  
- `SystemException`:

---

`setRollbackOnly`

void `setRollbackOnly()` throws `IllegalStateException`, `SystemException`

Modify the transaction associated with the target object such that the only possible outcome of the transaction is to roll back the transaction.

Throws:

- `IllegalStateException` - Thrown if the target object is not associated with any transaction.
- `SystemException` - Thrown if the transaction manager encounters an unexpected error condition.
to license terms.

PS:
javax.ejb  Annotation Type TransactionAttribute

@Target(value={METHOD, TYPE})
@Retention(value=RUNTIME)
public @interface TransactionAttribute

Implements: Annotation
@Target(value={METHOD, TYPE})
@Retention(value=RUNTIME)

TYPE-level  Bean

When applied at the TYPE-level, designates the default transaction attribute for all business methods of the session or message driven bean. When applied at the method-level, designates the transaction attribute for only that method.

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionAttributeType</td>
</tr>
</tbody>
</table>

abstract public TransactionAttributeType value()

value

public abstract TransactionAttributeType value

Default: REQUIRED
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Enum TransactionAttributeType

java.lang.Object
  ▼ java.lang.Enum<TransactionAttributeType>
  ▼ javax.ejb.TransactionAttributeType

All Implemented Interfaces:
  Serializable, Comparable<TransactionAttributeType>

public enum TransactionAttributeType
  extends Enum<TransactionAttributeType>

  Extends: Enum<E>

---

Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>MANDATORY</td>
</tr>
<tr>
<td>NEVER</td>
</tr>
<tr>
<td>NOT_SUPPORTED</td>
</tr>
<tr>
<td>REQUIRED</td>
</tr>
<tr>
<td>REQUIRES_NEW</td>
</tr>
<tr>
<td>SUPPORTS</td>
</tr>
</tbody>
</table>

---

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>valueOf(String name)</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>static <code>TransactionAttributeType</code></td>
</tr>
<tr>
<td>static <code>TransactionAttributeType[]</code> <code>values()</code></td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Enum`
- `clone`, `compareTo`, `equals`, `getDeclaringClass`, `hashCode`, `name`, `ordinal`, `toString`, `valueOf`

Methods inherited from class `java.lang.Object`
- `finalize`, `getClass`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
SUPPORTS
public static final TransactionAttributeType SUPPORTS

NOT_SUPPORTED
public static final TransactionAttributeType NOT_SUPPORTED

NEVER
public static final TransactionAttributeType NEVER

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
</table>

**final public static** TransactionAttributeType[] **values()**

**values**

public static final TransactionAttributeType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(TransactionAttributeType c : TransactionAttributeType.values
    System.out.println(c);
```

**Returns:**

an array containing the constants of this enum type, in the order they're declared
public static TransactionAttributeType valueOf(String name)

valueOf

public static TransactionAttributeType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
name - the name of the enum constant to be returned.

Returns:  
the enum constant with the specified name

Throws:  
IllegalArgumentException - if this enum type has no constant with the specified name

---

Overview  Package  Summary  Nested  Enum Constants  Field  Method  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS  SUMMARY: NESTED | ENUM CONSTANTS | FIELD | METHOD  FRAME | NO FRAMES  DETAIL: ENUM CONSTANTS | FIELD | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.jms Class TransactionInProgressException**

`java.lang.Object`<br>  └ `java.lang.Throwable`<br>     └ `java.lang.Exception`<br>         └ `javax.jms.JMSException`<br>             └ `javax.jms.TransactionInProgressException`

**All Implemented Interfaces:**

`Serializable`

---

```java
public class TransactionInProgressException
extends JMSException
```

**Extends:** Throwable > Exception > JMSException

**Version:** 26 August 1998

This exception is thrown when an operation is invalid because a transaction is in progress. For instance, an attempt to call `Session.commit` when a session is part of a distributed transaction should throw a `TransactionInProgressException`.

**Version:**
26 August 1998

**Author:**
Rahul Sharma

**See Also:**
Serialized Form

---

**Constructor Summary**
**Constructor Detail**

public TransactionInProgressException(String reason, String errorCode)

```
TransactionInProgressException

reason
errorCode
```

**Method Summary**

Methods inherited from class javax.jms.JMSException
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**TransactionInProgressException**

public TransactionInProgressException(String reason, String errorCode)
Constructs a `TransactionInProgressException` with the specified reason and error code.

**Parameters:**
- `reason` - a description of the exception
- `errorCode` - a string specifying the vendor-specific error code

```java
class TransactionInProgressException {
    public TransactionInProgressException(String reason)
        Constructs a `TransactionInProgressException` with the specified reason.
        The error code defaults to null.

        Parameters:
        - `reason` - a description of the exception
```

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.ejb Annotation Type TransactionManagement

@Target(value=TYPE)
@Retention(value=RUNTIME)
public @interface TransactionManagement

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)

Bean Bean Bean

Declares whether a session bean or message driven bean has container managed transactions or bean managed transactions.

### Optional Element Summary

<table>
<thead>
<tr>
<th>TransactionManagementType</th>
<th>value</th>
</tr>
</thead>
</table>

abstract public TransactionManagementType value()

value

public abstract TransactionManagementType value

**Default:**
CONTAINER
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMEs</td>
<td>NO FRAMEs</td>
<td>SUMMARY: NESTED</td>
<td>ENUM CONSTANTS</td>
</tr>
</tbody>
</table>
**java.ejb**  
**Enum** TransactionManagementType

```
java.lang.Object
 ⊄ java.lang.Enum<java.ejb.TransactionManagementType>
 ⊄ javax.ejb.TransactionManagementType
```

All Implemented Interfaces:  
`Serializable`, `Comparable<javax.ejb.TransactionManagementType>`

```java
class TransactionManagementType extends Enum<javax.ejb.TransactionManagementType> {

Extends: Enum<E>

<table>
<thead>
<tr>
<th>Enum Constant Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEAN</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>CONTAINER</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
</table>
| `valueOf(String name)`  
|     Returns the enum constant of this type with the specified name. |
| `values()`  
|     Returns an array containing the constants of this enum type, in the order they're declared. |

Methods inherited from class java.lang.Enum
```
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

CONTAINER

public static final TransactionManagementType CONTAINER

BEAN

public static final TransactionManagementType BEAN

Method Detail

final public static TransactionManagementType[] values()

values

public static final TransactionManagementType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(TransactionManagementType c : TransactionManagementType.values())
    System.out.println(c);
```
Returns:
an array containing the constants of this enum type, in the order they're declared

```java
public static TransactionManagementType valueOf(String name)
```

valueOf

```java
public static TransactionManagementType valueOf(String name)
```

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**
- `name` - the name of the enum constant to be returned.

**Returns:**
the enum constant with the specified name

**Throws:**
- `IllegalArgumentException` - if this enum type has no constant with the specified name
javax.transaction Interface TransactionManager

public interface TransactionManager

TransactionManager

The TransactionManager interface defines the methods that allow an application server to manage transaction boundaries.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void begin()</td>
<td>Create a new transaction and associate it with the current thread.</td>
</tr>
<tr>
<td>void commit()</td>
<td>Complete the transaction associated with the current thread.</td>
</tr>
<tr>
<td>int getStatus()</td>
<td>Obtain the status of the transaction associated with the current thread.</td>
</tr>
<tr>
<td>Transaction getTransaction()</td>
<td>Get the transaction object that represents the transaction context of the calling thread.</td>
</tr>
<tr>
<td>void resume(Transaction tobj)</td>
<td>Resume the transaction context association of the calling thread with the transaction represented by the supplied Transaction object.</td>
</tr>
<tr>
<td>void rollback()</td>
<td>Roll back the transaction associated with the current thread.</td>
</tr>
<tr>
<td>void setRollbackOnly()</td>
<td>Modify the transaction associated with the current thread.</td>
</tr>
</tbody>
</table>
void thread such that the only possible outcome of the transaction is to roll back the transaction.

void setTransactionTimeout(int seconds)

Modify the timeout value that is associated with transactions started by the current thread with the begin method.

Transaction suspend()

Suspend the transaction currently associated with the calling thread and return a Transaction object that represents the transaction context being suspended.

## Method Detail

public void begin() throws NotSupportedException, SystemException

Throws NotSupportedException: TransactionManager

Throws SystemException:

begin

void begin()

throws NotSupportedException, SystemException

Create a new transaction and associate it with the current thread.

**Throws:**
- NotSupportedException - Thrown if the thread is already associated with a transaction and the Transaction Manager implementation does not support nested transactions.
- SystemException - Thrown if the transaction manager encounters an unexpected error condition.
public void commit() throws RollbackException, HeuristicMixedException, HeuristicRollbackException, SecurityException, IllegalStateException, SystemException

Throws

- RollbackException
- HeuristicMixedException
- HeuristicRollbackException
- SecurityException
- IllegalStateException
- SystemException

commit

Complete the transaction associated with the current thread. When this method completes, the thread is no longer associated with a transaction.

Throws:

- RollbackException - Thrown to indicate that the transaction has been rolled back rather than committed.
- HeuristicMixedException - Thrown to indicate that a heuristic decision was made and that some relevant updates have been committed while others have been rolled back.
- HeuristicRollbackException - Thrown to indicate that a heuristic decision was made and that all relevant updates have been rolled back.
- SecurityException - Thrown to indicate that the thread is not allowed to commit the transaction.
- IllegalStateException - Thrown if the current thread is not associated with a transaction.
SystemException - Thrown if the transaction manager encounters an unexpected error condition.

public int getStatus() throws SystemException

    return Status.NoTransaction
    Throws SystemException:

getStatus

int getStatus()
    throws SystemException

Obtain the status of the transaction associated with the current thread.

Returns:
The transaction status. If no transaction is associated with the current thread, this method returns the Status.NoTransaction value.

Throws:
SystemException - Thrown if the transaction manager encounters an unexpected error condition.

public Transaction getTransaction() throws SystemException

    return Transaction
    Throws SystemException:

getTransaction

Transaction getTransaction()
    throws SystemException
Get the transaction object that represents the transaction context of the calling thread.

**Returns:**
the Transaction object representing the transaction associated with the calling thread.

**Throws:**
[SystemException](#) - Thrown if the transaction manager encounters an unexpected error condition.

---

```java
public void resume(Transaction obj) throws
InvalidTransactionException, IllegalStateException, SystemException
Transaction
obj
Throws
InvalidTransactionException:
Throws
IllegalStateException:
Throws
SystemException:
```

Resume the transaction context association of the calling thread with the transaction represented by the supplied Transaction object. When this method returns, the calling thread is associated with the transaction context specified.

**Parameters:**
obj - The Transaction object that represents the transaction to be resumed.

**Throws:**
[InvalidTransactionException](#) - Thrown if the parameter
transaction object contains an invalid transaction.

**IllegalStateException** - Thrown if the thread is already associated with another transaction.
**SystemException** - Thrown if the transaction manager encounters an unexpected error condition.

```java
public void rollback() throws IllegalStateException, SecurityException, SystemException
```

Throws:
- **SecurityException**: 
- **IllegalStateException**: 
- **SystemException**: 

**rollback**

```java
void rollback()
    throws IllegalStateException, SecurityException, SystemException
```

Roll back the transaction associated with the current thread. When this method completes, the thread is no longer associated with a transaction.

**Throws:**
- **SecurityException** - Thrown to indicate that the thread is not allowed to roll back the transaction.
- **IllegalStateException** - Thrown if the current thread is not associated with a transaction.
- **SystemException** - Thrown if the transaction manager encounters an unexpected error condition.

```java
public void setRollbackOnly() throws IllegalStateException, SystemException
```
Throws:  
  
**setRollbackOnly**  

```java
void setRollbackOnly()
  throws IllegalStateException, SystemException
```

Modify the transaction associated with the current thread such that the only possible outcome of the transaction is to roll back the transaction.

**Throws:**

- **IllegalStateException** - Thrown if the current thread is not associated with a transaction.
- **SystemException** - Thrown if the transaction manager encounters an unexpected error condition.

---

**public void setTransactionTimeout(int seconds) throws SystemException**

```java
begin
  seconds 0 SystemException

Throws  SystemException:
```

---

**setTransactionTimeout**  

```java
void setTransactionTimeout(int seconds)
  throws SystemException
```

Modify the timeout value that is associated with transactions started by the current thread with the begin method.
If an application has not called this method, the transaction service uses some default value for the transaction timeout.

**Parameters:**
- seconds - The value of the timeout in seconds. If the value is zero, the transaction service restores the default value. If the value is negative a SystemException is thrown.

** Throws:**
- SystemException - Thrown if the transaction manager encounters an unexpected error condition.

```java
public Transaction suspend() throws SystemException {
    return null;
}
```

**suspend**

Suspend the transaction currently associated with the calling thread and return a Transaction object that represents the transaction context being suspended. If the calling thread is not associated with a transaction, the method returns a null object reference. When this method returns, the calling thread is not associated with a transaction.

**Returns:**
- Transaction object representing the suspended transaction.

** Throws:**
- SystemException - Thrown if the transaction manager encounters an unexpected error condition.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.persistence Class TransactionRequiredException

java.lang.Object  
  java.lang.Throwable  
    java.lang.Exception  
      java.lang.RuntimeException  
        javax.persistence.PersistenceException  
          javax.persistence.TransactionRequiredException

All Implemented Interfaces: Serializable

public class TransactionRequiredException
  extends PersistenceException

Extends: Throwable > Exception > RuntimeException > PersistenceException

since Java Persistence 1.0

Thrown by the persistence provider when a transaction is required but is not active.

Since: Java Persistence 1.0

See Also: Serialized Form

Constructor Summary

TransactionRequiredException()  
  Constructs a new TransactionRequiredException exception with null as its detail message.
**TransactionRequiredException** *(String message)*

Constructs a new TransactionRequiredException exception with the specified detail message.

**Method Summary**

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

**Constructor Detail**

public TransactionRequiredException()

null TransactionRequiredException

TransactionRequiredException

public TransactionRequiredException()

Constructs a new TransactionRequiredException exception with null as its detail message.

public TransactionRequiredException(String message)

TransactionRequiredException message
TransactionRequiredException

public TransactionRequiredException(String message)

    Constructs a new TransactionRequiredException exception with the specified detail message.

Parameters:
    message - the detail message.
javax.transaction  Class TransactionRequiredException

java.lang.Object
  ▼ java.lang.Throwable
    ▼ java.lang.Exception
      ▼ java.io.IOException
        ▼ java.rmi.RemoteException
          ▼ javax.transaction.TransactionRequiredException

All Implemented Interfaces:
  Serializable

public class TransactionRequiredException
  extends RemoteException

  Extends: Throwable > Exception > java.io.IOException > java.rmi.RemoteException

  null

This exception indicates that a request carried a null transaction context, but the target object requires an activate transaction.

See Also:
  Serialized Form

Field Summary

| Fields inherited from class java.rmi.RemoteException |
| detail |

Constructor Summary

TransactionRequiredException()
**TransactionRequiredException** (String msg)

### Method Summary

Methods inherited from class java.rmi.**RemoteException**
- getCause
- getMessage

Methods inherited from class java.lang.**Throwable**
- fillInStackTrace
- getLocalizedMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.**Object**
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail

public **TransactionRequiredException**()

**TransactionRequiredException**

public **TransactionRequiredException**()

public **TransactionRequiredException**(String msg)

**TransactionRequiredException**

public **TransactionRequiredException**(String msg)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.ejb  **Class TransactionRequiredLocalException**

```java
java.lang.Object
├ java.lang.Throwable
│  └ java.lang.Exception
│     └ java.lang.RuntimeException
│          └ javax.ejb.EJBException
└ javax.ejb.TransactionRequiredLocalException
```

All Implemented Interfaces:
  ```
  Serializable
  ```

```java
public class TransactionRequiredLocalException

extends EJBException
```

**Extends**: Throwable > Exception > RuntimeException > **EJBException**

null

This exception indicates that a request carried a null transaction context, but the target object requires an active transaction.

**See Also:**
  ```
 Serialized Form
  ```

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TransactionRequiredLocalException()</code></td>
<td>Constructs a TransactionRequiredLocalException with no detail message.</td>
</tr>
<tr>
<td><code>TransactionRequiredLocalException(String message)</code></td>
<td>Constructs an TransactionRequiredLocalException with the specified detailed message.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Methods inherited from class</th>
<th>javax.ejb.EJBException</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getCausedByException</code>, <code>getMessage</code>, <code>printStackTrace</code>, <code>printStackTrace</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class</th>
<th>java.lang.Throwable</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>fillInStackTrace</code>, <code>getCause</code>, <code>getLocalizedMessage</code>, <code>getStackTrace</code>, <code>initCause</code>, <code>setStackTrace</code>, <code>toString</code></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class</th>
<th>java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clone</code>, <code>equals</code>, <code>finalize</code>, <code>getClass</code>, <code>hashCode</code>, <code>notify</code>, <code>notifyAll</code>, <code>wait</code>, <code>wait</code>, <code>wait</code></td>
<td></td>
</tr>
</tbody>
</table>

### Constructor Detail

```java
public TransactionRequiredLocalException()
TransactionRequiredLocalException

public TransactionRequiredLocalException()
    Constructs a TransactionRequiredLocalException with no detail message.

public TransactionRequiredLocalException(String message)
TransactionRequiredLocalException
```

```
```
public TransactionRequiredLocalException(String message)

Constructs an TransactionRequiredLocalException with the specified detailed message.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Class TransactionRolledBackException

java.lang.Object
    ▼ java.lang.Throwable
      ▼ java.lang.Exception
        ▼ javax.jms.JMSException
          ▼ javax.jms.TransactionRolledBackException

All Implemented Interfaces:
    Serializable

public class TransactionRolledBackException
    extends JMSException

Extends: Throwable > Exception > JMSException

This exception must be thrown when a call to Session.commit results in a rollback of the current transaction.

Version:
    26 August 1998

Author:
    Rahul Sharma

See Also:
    Serialized Form

---

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransactionRolledBackException(String reason)</td>
</tr>
<tr>
<td>Constructs a TransactionRolledBackException with the specified</td>
</tr>
</tbody>
</table>
TransactionRolledBackException (String reason, String errorCode)

Constructs a TransactionRolledBackException with the specified reason and error code.

Method Summary

Methods inherited from class javax.jms.JMSEception
getErrorCode, getLinkedException, setLinkedException

Methods inherited from class java.lang.Throwable
fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Constructor Detail

public TransactionRolledBackException(String reason, String errorCode)

TransactionRolledBackException reason
errorCode

TransactionRolledBackException

public TransactionRolledBackException(String reason, String errorCode)

Constructs a TransactionRolledBackException with the specified
reason and error code.

**Parameters:**
- reason - a description of the exception
- errorCode - a string specifying the vendor-specific error code

```java
public TransactionRolledBackException(String reason)

TransactionRolledBackException null
reason
```

TransactionRolledBackException

```java
public TransactionRolledBackException(String reason)

Constructs a TransactionRolledBackException with the specified reason. The error code defaults to null.

**Parameters:**
- reason - a description of the exception
```
javax.transaction Class TransactionRolledbackException

All Implemented Interfaces:
Serializable

public class TransactionRolledbackException extends RemoteException

Extends: Throwable > Exception > java.io.IOException > java.rmi.RemoteException

This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back. Thus the requested operation either could not be performed or was not performed because further computation on behalf of the transaction would be fruitless

See Also:
Serialized Form

Field Summary

Fields inherited from class java.rmi.RemoteException

detail
## Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TransactionRolledbackException()</code></td>
</tr>
<tr>
<td><code>TransactionRolledbackException(String msg)</code></td>
</tr>
</tbody>
</table>

## Method Summary

### Methods inherited from class java.rmi.RemoteException
- `getCause`, `getMessage`

### Methods inherited from class java.lang.Throwable
- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class java.lang.Object
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
TransactionRolledbackException

public TransactionRolledbackException(String msg)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.ejb Class
TransactionRolledbackLocalException

java.lang.Object
  ↓ java.lang.Throwable
  ↓ java.lang.Exception
  ↓ java.lang.RuntimeException
  ↓ javax.ejb.EJBException
  ↓ javax.ejb.TransactionRolledbackLocalException

All Implemented Interfaces:
  Serializable

public class TransactionRolledbackLocalException
  extends EJBException

Extends: Throwable > Exception > RuntimeException > EJBException

This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back. Thus the requested operation either could not be performed or was not performed because further computation on behalf of the transaction would be fruitless

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>TransactionRolledbackLocalException()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a TransactionRolledbackLocalException with no detail message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TransactionRolledbackLocalException(String message)</th>
</tr>
</thead>
</table>
Constructs a TransactionRolledbackLocalException with the specified detailed message.

```java
TransactionRolledbackLocalException(String message, Exception ex)
```

Constructs a TransactionRolledbackLocalException with the specified detail message and a nested exception.

**Method Summary**

**Methods inherited from class javax.ejb.EJBException**

- getCausedByException
- getMessage
- printStackTrace
- printStackTrace
- printStackTrace

**Methods inherited from class java.lang.Throwable**

- fillInStackTrace
- getCause
- getLocalizedMessage
- getStackTrace
- initCause
- setStackTrace
- toString

**Methods inherited from class java.lang.Object**

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

**Constructor Detail**

```java
public TransactionRolledbackLocalException()
```

**TransactionRolledbackLocalException**

```java
public TransactionRolledbackLocalException()
```

Constructs a TransactionRolledbackLocalException with no detail message.
public TransactionRolledbackLocalException(String message)
TransactionRolledbackLocalException

TransactionRolledbackLocalException

public TransactionRolledbackLocalException(String message)

    Constructs a TransactionRolledbackLocalException with the specified detailed message.

TransactionRolledbackLocalException

public TransactionRolledbackLocalException(String message, Exception ex)
TransactionRolledbackLocalException

TransactionRolledbackLocalException

public TransactionRolledbackLocalException(String message, Exception ex)

    Constructs a TransactionRolledbackLocalException with the specified detail message and a nested exception.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
This interface is intended for use by system level application server components such as persistence managers, resource adapters, as well as EJB and Web application components. This provides the ability to register synchronization objects with special ordering semantics, associate resource objects with the current transaction, get the transaction context of the current transaction, get current transaction status, and mark the current transaction for rollback. This interface is implemented by the application server by a stateless service object. The same object can be used by any number of components with thread safety.

In standard application server environments, an instance implementing this interface can be looked up by a standard name via JNDI. The standard name is java:comp/TransactionSynchronizationRegistry.

Since: JTA 1.1

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getResource(key)</code></td>
<td>Get an object from the Map of resources being managed</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>boolean getRollbackOnly()</td>
<td>Get the rollbackOnly status of the transaction bound to the current thread at the time this method is called.</td>
</tr>
<tr>
<td>Object getTransactionKey()</td>
<td>Return an opaque object to represent the transaction bound to the current thread at the time this method is called.</td>
</tr>
<tr>
<td>int getTransactionStatus()</td>
<td>Return the status of the transaction bound to the current thread at the time this method is called.</td>
</tr>
<tr>
<td>void putResource(Object key, Object value)</td>
<td>Add or replace an object in the Map of resources being managed for the transaction bound to the current thread at the time this method is called.</td>
</tr>
<tr>
<td>void registerInterposedSynchronization(Synchronization sync)</td>
<td>Register a Synchronization instance with special ordering semantics.</td>
</tr>
<tr>
<td>void setRollbackOnly()</td>
<td>Set the rollbackOnly status of the transaction bound to the current thread at the time this method is called.</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public Object getTransactionKey()
hashCode  equals hashMap  null

hashCode

toString  StringtoString  toString

return since  JTA 1.1
```
getTransactionKey

Object getTransactionKey()

Return an opaque object to represent the transaction bound to the current thread at the time this method is called. This object overrides hashCode and equals to allow its use as the key in a hashMap for use by the caller. If there is no transaction currently active, return null.

This object will return the same hashCode and compare equal to all other objects returned by calling this method from any component executing in the same transaction context in the same application server.

The toString method returns a String that might be usable by a human reader to usefully understand the transaction context. The toString result is otherwise not defined. Specifically, there is no forward or backward compatibility guarantee of the results of toString.

The object is not necessarily serializable, and has no defined behavior outside the virtual machine whence it was obtained.

Returns:
an opaque object representing the transaction bound to the current thread at the time this method is called.

Since:
JTA 1.1

public void putResource(Object key, Object value)
Map hashCode equals
null null Map put(Object, Object) value

key Map
putResource

void putResource(Object key, Object value)

Add or replace an object in the Map of resources being managed for the transaction bound to the current thread at the time this method is called. The supplied key should be of an caller-defined class so as not to conflict with other users. The class of the key must guarantee that the hashCode and equals methods are suitable for use as keys in a map. The key and value are not examined or used by the implementation. The general contract of this method is that of Map.put(Object, Object) for a Map that supports non-null keys and null values. For example, if there is already an value associated with the key, it is replaced by the value parameter.

Parameters:
  key - the key for the Map entry.
  value - the value for the Map entry.

Throws:
  IllegalStateException - if no transaction is active.
  NullPointerException - if the parameter key is null.

Since:
  JTA 1.1

public Object getResource(Object key)
Map putResouce Map null
null null Map get(Object) null null
key Map
return
Throws IllegalStateException:
Throws

NullPointerException: key null

since

JTA 1.1

getResource

Object getResource(Object key)

Get an object from the Map of resources being managed for the transaction bound to the current thread at the time this method is called. The key should have been supplied earlier by a call to putResource in the same transaction. If the key cannot be found in the current resource Map, null is returned. The general contract of this method is that of Map.get(Object) for a Map that supports non-null keys and null values. For example, the returned value is null if there is no entry for the parameter key or if the value associated with the key is actually null.

Parameters:
key - the key for the Map entry.

Returns:
the value associated with the key.

Throws:

IllegalStateException - if no transaction is active.
NullPointerExcetion - if the parameter key is null.

Since:
JTA 1.1

public void
registerInterposedSynchronization(Synchronization sync)
Synchronization beforeCompletion
SessionSynchronization beforeCompletion Transaction
afterCompletion SessionSynchronization
Transaction afterCompletion

beforeCompletion Connector““
Bean registerInterposedSynchronization
registerInterposedSynchronization

afterCompletion ""

IllegalStateException

IllegalStateException

throws

IllegalStateException: 

since

JTA 1.1

registerInterposedSynchronization

void registerInterposedSynchronization(Synchronization sync)

Register a Synchronization instance with special ordering semantics. Its beforeCompletion will be called after all SessionSynchronization beforeCompletion callbacks and callbacks registered directly with the Transaction, but before the 2-phase commit process starts. Similarly, the afterCompletion callback will be called after 2-phase commit completes but before any SessionSynchronization and Transaction afterCompletion callbacks.

The beforeCompletion callback will be invoked in the transaction context of the transaction bound to the current thread at the time this method is called. Allowable methods include access to resources, e.g. Connectors. No access is allowed to "user components" (e.g. timer services or bean methods), as these might change the state of data being managed by the caller, and might change the state of data that has already been flushed by another caller of registerInterposedSynchronization. The general context is the component context of the caller of registerInterposedSynchronization.

The afterCompletion callback will be invoked in an undefined
context. No access is permitted to "user components" as defined above. Resources can be closed but no transactional work can be performed with them.

If this method is invoked without an active transaction context, an IllegalStateException is thrown.

If this method is invoked after the two-phase commit processing has started, an IllegalStateException is thrown.

**Parameters:**
- `sync` - the Synchronization instance.

**Throws:**
- `IllegalStateException` - if no transaction is active.

**Since:**
- JTA 1.1

```java
public int getTransactionStatus()
TransactionManager.getStatus()
    return
    since
JTA 1.1
```

**getTransactionStatus**

```java
int getTransactionStatus()
```

Return the status of the transaction bound to the current thread at the time this method is called. This is the result of executing `TransactionManager.getStatus()` in the context of the transaction bound to the current thread at the time this method is called.

**Returns:**
- the status of the transaction bound to the current thread at the time this method is called.

**Since:**
- JTA 1.1
public void setRollbackOnly()

void setRollbackOnly()

Set the rollbackOnly status of the transaction bound to the current thread at the time this method is called.

Throws:
IllegalStateException - if no transaction is active.

Since:
JTA 1.1

public boolean getRollbackOnly()

boolean getRollbackOnly()

Get the rollbackOnly status of the transaction bound to the current thread at the time this method is called.

Returns:
the rollbackOnly status.

Throws:
IllegalStateException - if no transaction is active.

Since:
javax.persistence Annotation Type Transient

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)
public @interface Transient

**Implements:** Annotation

@Target(value={METHOD, FIELD})
@Retention(value=RUNTIME)

@Entity
public class Employee {
    @Id int id;
    @Transient User currentUser;
    ...
}

since Java Persistence 1.0

This annotation specifies that the property or field is not persistent. It is used to annotate a property or field of an entity class, mapped superclass, or embeddable class.

Example:

@Entity
public class Employee {
    @Id int id;
    @Transient User currentUser;
    ...
}

Since: Java Persistence 1.0
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail Class Transport

java.lang.Object
   \ javax.mail.Service
      \ javax.mail.Transport

public abstract class Transport extends Service

Extends: Service

An abstract class that models a message transport. Subclasses provide actual implementations.

Note that Transport extends the Service class, which provides many common methods for naming transports, connecting to transports, and listening to connection events.

Version: 1.40, 07/05/17

Author: John Mani, Max Spivak, Bill Shannon

See Also: Service, ConnectionEvent, TransportEvent

Field Summary
### Fields inherited from class javax.mail.Service

depbug, session, url

---

### Constructor Summary

**Transport** *(Session session, URLName urlname)*

Constructor.

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
</table>
| **void** | **addTransportListener** *(TransportListener l)*  
Add a listener for Transport events. |
| **protected void** | **notifyTransportListeners** *(int type, Address[] validSent, Address[] validUnsent, Address[] invalid, Message msg)*  
Notify all TransportListeners. |
| **void** | **removeTransportListener** *(TransportListener l)*  
Remove a listener for Transport events. |
| **static void** | **send** *(Message msg)*  
Send a message. |
| **static void** | **send** *(Message msg, Address[] addresses)*  
Send the message to the specified addresses, ignoring any recipients specified in the message itself. |
| **abstract void** | **sendMessage** *(Message msg, Address[] addresses)*  
Send the Message to the specified list of addresses. |

---

### Methods inherited from class javax.mail.Service

addConnectionListener, close, connect, connect, connect, connect, finalize, getURLName, isConnected, notifyConnectionListeners, protocolConnect, queueEvent, removeConnectionListener, setConnected, setURLName, toString

---

### Methods inherited from class java.lang.Object

clone, equals, getClass, hashCode, notify, notifyAll, wait, wait, wait
### Constructor Detail

**public Transport(Session session, URLName urlName)**

- **session** - Transport Session
- **urlName** - Transport URLName

### Transport

**public Transport(Session session, URLName urlName)**

Constructor.

**Parameters:**
- **session** - Session object for this Transport.
- **urlName** - URLName object to be used for this Transport

### Method Detail

**public static void send(Message msg) throws MessagingException**

```
Transport sendFailedException = Transport.send(msg);
```

**Throws**
- **SendFailedException**
- **MessagingException**
send

public static void send(Message msg)
throws MessagingException

Send a message. The message will be sent to all recipient addresses specified in the message (as returned from the Message method getAllRecipients), using message transports appropriate to each address. The send method calls the saveChanges method on the message before sending it.

If any of the recipient addresses is detected to be invalid by the Transport during message submission, a SendFailedException is thrown. Clients can get more detail about the failure by examining the exception. Whether or not the message is still sent successfully to any valid addresses depends on the Transport implementation. See SendFailedException for more details. Note also that success does not imply that the message was delivered to the ultimate recipient, as failures may occur in later stages of delivery. Once a Transport accepts a message for delivery to a recipient, failures that occur later should be reported to the user via another mechanism, such as returning the undeliverable message.

Note that send is a static method that creates and manages its own connection. Any connection associated with any Transport instance used to invoke this method is ignored and not used. This method should only be invoked using the form Transport.send(msg);, and should never be invoked using an instance variable.

Parameters:
  - msg - the message to send

Throws:
  - SendFailedException - if the message could not be sent to some or any of the recipients.
public static void send(Message msg, Address[] addresses) throws MessagingException

Throws:
- SendFailedException - if the message could not be sent to some or any of the recipients.
- MessagingException

See Also:
- Message.saveChanges(), send(Message), SendFailedException

send

public static void send(Message msg, Address[] addresses) throws MessagingException

Parameters:
- msg - the message to send
- addresses - the addresses to which to send the message

Throws:
- SendFailedException - if the message could not be sent to some or any of the recipients.
- MessagingException

See Also:
- Message.saveChanges(), send(Message), SendFailedException
abstract public void sendMessage(Message msg, Address[] addresses) throws MessagingException

Message Transport TransportListener
TransportEvent SendFailedException
Transport

sendMessage

public abstract void sendMessage(Message msg, Address[] addresses) throws MessagingException

Throws SendFailedException:
Throws MessagingException:
See also javax.mail.event.TransportEvent

sendMessage

public abstract void sendMessage(Message msg, Address[] addresses) throws MessagingException

Send the Message to the specified list of addresses. An appropriate TransportEvent indicating the delivery status is delivered to any TransportListener registered on this Transport. Also, if any of the addresses is invalid, a SendFailedException is thrown. Whether or not the message is still sent successfully to any valid addresses depends on the Transport implementation.

Unlike the static send method, the sendMessage method does not call the saveChanges method on the message; the caller should do so.

Parameters:
msg - The Message to be sent
addresses - array of addresses to send this message to

Throws:
SendFailedException - if the send failed because of invalid addresses.
MessagingException - if the connection is dead or not in the
public void addTransportListener(TransportListener l)

Transport

TransportListener

/  

See also javax.mail.event.TransportEvent

addTransportListener

public void addTransportListener(TransportListener l)

Add a listener for Transport events.

The default implementation provided here adds this listener to an internal list of TransportListeners.

Parameters:

l - the Listener for Transport events

See Also:

TransportEvent

public void removeTransportListener(TransportListener l)

Transport

TransportListener

/  

See also addTransportListener
removeTransportListener

public void removeTransportListener(TransportListener l)

Remove a listener for Transport events.

The default implementation provided here removes this listener from the internal list of TransportListeners.

Parameters:
   l - the listener

See Also:
   addTransportListener(javax.mail.event.TransportListener)

protected void notifyTransportListeners(int type, Address[] validSent, Address[] validUnsent, Address[] invalid, Message msg)
TransportListenerTransport  TransportEvent
TransportListener

notifyTransportListeners

protected void notifyTransportListeners(int type, Address[] validSent, Address[] validUnsent, Address[] invalid, Message msg)

Notify all TransportListeners. Transport implementations are expected to use this method to broadcast TransportEvents.

The provided default implementation queues the event into an internal event queue. An event dispatcher thread dequeues events from the queue and dispatches them to the registered TransportListeners. Note that the event dispatching occurs in a separate thread, thus avoiding potential deadlock problems.
javax.mail.event Class TransportAdapter

java.lang.Object
   ^ javax.mail.event.TransportAdapter

All Implemented Interfaces:
   EventListener, TransportListener

public abstract class TransportAdapter
extends Object
implements TransportListener

Implements: TransportListener

Transport

The adapter which receives Transport events. The methods in this class are empty; this class is provided as a convenience for easily creating listeners by extending this class and overriding only the methods of interest.

Author:
   John Mani

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransportAdapter()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>void messageDelivered(TransportEvent e)</td>
</tr>
<tr>
<td>Invoked when a Message is succesfully delivered.</td>
</tr>
<tr>
<td>void messageNotDelivered(TransportEvent e)</td>
</tr>
<tr>
<td>void</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>void messagePartiallyDelivered(TransportEvent e)</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public TransportAdapter()

TransportAdapter

public TransportAdapter()

Method Detail

public void messageDelivered(TransportEvent e)

messageDelivered

public void messageDelivered(TransportEvent e)

Description copied from interface: TransportListener
Invoked when a Message is succesfully delivered.

Specified by:
messageDelivered in interface TransportListener

Parameters:
e - TransportEvent
public void messageNotDelivered(TransportEvent e)

Description copied from interface: TransportListener
Invoked when a Message is not delivered.

Specified by:
messageNotDelivered in interface TransportListener
Parameters:
e - TransportEvent
See Also:
TransportEvent

public void messagePartiallyDelivered(TransportEvent e)

Description copied from interface: TransportListener
Invoked when a Message is partially delivered.

Specified by:
messagePartiallyDelivered in interface TransportListener
Parameters:
e - TransportEvent
See Also:
TransportEvent
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.mail.event  **Class TransportEvent**

**java.lang.Object**
   ▼ **java.util.EventObject**
      ▼ **javax.mail.event.MailEvent**
         ▼ **javax.mail.event.TransportEvent**

**All Implemented Interfaces:**
   **Serializable**

```java
public class TransportEvent extends MailEvent

Extends: java.util.EventObject > MailEvent

Transport

See also javax.mail.Transport, javax.mail.event.TransportListener

This class models Transport events.

Author:
   John Mani, Max Spivak

See Also:
   Transport, TransportListener, Serialized Form
```

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected <code>Address[]</code></td>
<td><code>invalid</code></td>
</tr>
<tr>
<td>static <code>int</code></td>
<td><code>MESSAGE_DELIVERED</code></td>
</tr>
</tbody>
</table>

Message has been successfully delivered to all recipients by the transport firing this event. `validSent[]` contains all the addresses this transport sent to successfully.
validUnsent[] and invalid[] should be null,

static int MESSAGE_NOT_DELIVERED
Message was not sent for some reason. validSent[] should be null.

static int MESSAGE_PARTIALLY_DELIVERED
Message was successfully sent to some recipients but not to all.

protected Message msg

protected int type
The event type.

protected Address[] validSent

protected Address[] validUnsent

Fields inherited from class java.util.EventObject
source

Constructor Summary
TransportEvent(Transport transport, int type, Address[] validSent, Address[] validUnsent, Address[] invalid, Message msg)
Constructor.

Method Summary
void dispatch(Object listener)
Invokes the appropriate TransportListener method.

Address[] getInvalidAddresses()
Return the addresses to which this message could not be sent.

Message getMessage()
Get the Message object associated with this Transport Event.
**MessageType**

Return the type of this event.

**getValidSentAddresses()**

Return the addresses to which this message was sent successfully.

**getValidUnsentAddresses()**

Return the addresses that are valid but to which this message was not sent.

### Methods inherited from class java.util.EventObject

- getSource, toString

### Methods inherited from class java.lang.Object

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

### Field Detail

**MESSAGE_DELIVERED**

public static final int MESSAGE_DELIVERED

Message has been successfully delivered to all recipients by the transport firing this event. validSent[] contains all the addresses this transport sent to successfully. validUnsent[] and invalid[] should be null,

**See Also:**

Constant Field Values
MESSAGE_NOT_DELIVERED

public static final int MESSAGE_NOT_DELIVERED

    Message was not sent for some reason. validSent[] should be null. validUnsent[] may have addresses that are valid (but the message wasn't sent to them). invalid[] should likely contain invalid addresses.

See Also:
    Constant Field Values

MESSAGE_PARTIALLY_DELIVERED

public static final int MESSAGE_PARTIALLY_DELIVERED

    Message was successfully sent to some recipients but not to all. validSent[] holds addresses of recipients to whom the message was sent. validUnsent[] holds valid addresses to which the message was not sent. invalid[] holds invalid addresses, if any.

See Also:
    Constant Field Values

type

protected int type

    The event type.

validSent
protected transient Address[] validSent

validUnsent

protected transient Address[] validUnsent

invalid

protected transient Address[] invalid

msg

protected transient Message msg

Constructor Detail

public TransportEvent(Transport transport, int type, Address[] validSent, Address[] validUnsent, Address[] invalid, Message msg)

TransportEvent

public TransportEvent(Transport transport, int type,
Address[] validSent,
Address[] validUnsent,
Address[] invalid, Message msg)

Constructor.

Parameters:
transport - The Transport object

Method Detail

public int getType()

  return

getType

public int getType()

  Return the type of this event.

  Returns:
  type

public Address[] getValidSentAddresses()

  return null

getValidSentAddresses

public Address[] getValidSentAddresses()

  Return the addresses to which this message was sent successfully.

  Returns:
  Addresses to which the message was sent successfully or null
public Address[] getValidUnsentAddresses()
    return null

getValidUnsentAddresses

public Address[] getValidUnsentAddresses()
    Return the addresses that are valid but to which this message was not sent.

    Returns:
    Addresses that are valid but to which the message was not sent successfully or null

public Address[] getInvalidAddresses()
    return null

getInvalidAddresses

public Address[] getInvalidAddresses()
    Return the addresses to which this message could not be sent.

    Returns:
    Addresses to which the message sending failed or null

public Message getMessage()
    return Message

    since JavaMail 1.2

    en
getMessage

public Message getMessage()

Get the Message object associated with this Transport Event.

Returns:
the Message object

Since:
JavaMail 1.2

public void dispatch(Object listener)
TransportListener

dispatch

public void dispatch(Object listener)

Invokes the appropriate TransportListener method.

Specified by:
dispatch in class MailEvent

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature
javax.mail.event Interface TransportListener

All Superinterfaces:
 EventListener

All Known Implementing Classes:
TransportAdapter

public interface TransportListener
extends EventListener

Implements: java.util.EventListener
Implemented by: TransportAdapter

Transport Listener

See also  javax.mail.Transport, javax.mail.event.TransportEvent

This is the Listener interface for Transport events

Author:
  John Mani, Max Spivak

See Also:
  Transport, TransportEvent

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void messageDelivered(TransportEvent e)</td>
</tr>
<tr>
<td>Invoked when a Message is succesfully delivered.</td>
</tr>
<tr>
<td>void messageNotDelivered(TransportEvent e)</td>
</tr>
<tr>
<td>Invoked when a Message is not delivered.</td>
</tr>
<tr>
<td>void messagePartiallyDelivered(TransportEvent e)</td>
</tr>
<tr>
<td>Invoked when a Message is partially delivered.</td>
</tr>
</tbody>
</table>
Method Detail

public void messageDelivered(TransportEvent e)  
Message  
e  
TransportEvent  

messageDelivered  
void messageDelivered(TransportEvent e)  
  Invoked when a Message is successfully delivered.  
  Parameters: 
    e - TransportEvent  

public void messageNotDelivered(TransportEvent e)  
Message  
e  
TransportEvent  
  See also  
    javax.mail.event.TransportEvent  

messageNotDelivered  
void messageNotDelivered(TransportEvent e)  
  Invoked when a Message is not delivered.  
  Parameters: 
    e - TransportEvent  
  See Also: 
    TransportEvent  

public void messagePartiallyDelivered(TransportEvent e)
messagePartiallyDelivered

void messagePartiallyDelivered(TransportEvent e)

Invoked when a Message is partially delivered.

Parameters:
  e - TransportEvent

See Also:
  TransportEvent
javax.servlet.jsp.tagext Interface TryCatchFinally

public interface TryCatchFinally

(hook) TagIterationTag BodyTag
doCatch(Throwbaible) doFinally()

h = get a Tag(); // get a tag handler, perhaps from pool
h setPageContext(pc); // initialize as desired
h.setParent(null);
h.setFoo("foo");

// tag invocation protocol; see Tag.java
try {
  doStartTag()...
  ....
  doEndTag()...
} catch (Throwable t) {
  // react to exceptional condition
  h.doCatch(t);
} finally {
  // restore data invariants and release per-invocation resources
  h.doFinally();
}

... other invocations perhaps with some new setters
...

h.release(); // release long-term resources

The auxiliary interface of a Tag, IterationTag or BodyTag tag handler that wants additional hooks for managing resources.

This interface provides two new methods: doCatch(Throwbaible) and doFinally(). The prototypical invocation is as follows:

h = get a Tag(); // get a tag handler, perhaps from pool
h.setPageContext(pc); // initialize as desired
h.setParent(null);
h.setFoo("foo");

// tag invocation protocol; see Tag.java
try {
    doStartTag()...
    ....
    doEndTag()...
} catch (Throwable t) {
    // react to exceptional condition
    h.doCatch(t);
} finally {
    // restore data invariants and release per-invocation resources
    h.doFinally();
}

... other invocations perhaps with some new setters
...

h.release(); // release long-term resources

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>doCatch(Throwables t)</code></td>
<td>Invoked if a Throwable occurs while evaluating the BODY inside a tag or in any of the following methods: Tag.doStartTag(), Tag.doEndTag(), IterationTag.doAfterBody() and BodyTag.doInitBody.</td>
</tr>
<tr>
<td><code>doFinally()</code></td>
<td>Invoked in all cases after doEndTag() for any class implementing Tag, IterationTag or BodyTag.</td>
</tr>
</tbody>
</table>

---

### Method Detail

```java
public void doCatch(Throwables t) throws Throwable
Tag.doStartTag()Tag.doEndTag()
IterationTag.doAfterBody() BodyTag.doInitBody() Throwable
```
setter Throwable

doFinally()

    t        throwable
Throws       Throwable:

doCatch

void doCatch(Throwable t)
        throws Throwable

Invoked if a Throwable occurs while evaluating the BODY inside a
tag or in any of the following methods: Tag.doStartTag(),
Tag.doEndTag(), IterationTag.doAfterBody() and
BodyTag.doInitBody().

This method is not invoked if the Throwable occurs during one of the
setter methods.

This method may throw an exception (the same or a new one) that
will be propagated further up the nest chain. If an exception is
thrown, doFinally() will be invoked.

This method is intended to be used to respond to an exceptional
condition.

Parameters:
    t - The throwable exception navigating through this tag.

Throws:
    Throwable - if the exception is to be rethrown further up the nest
    chain.

public void doFinally()
doFinally

void doFinally()

Invoked in all cases after doEndTag() for any class implementing Tag, IterationTag or BodyTag. This method is invoked even if an exception has occurred in the BODY of the tag, or in any of the following methods: Tag.doStartTag(), Tag.doEndTag(), IterationTag.doAfterBody() and BodyTag.doInitBody().

This method is not invoked if the Throwable occurs during one of the setter methods.

This method should not throw an Exception.

This method is intended to maintain per-invocation data integrity and resource management actions.
to license terms.

PS:
javax.xml.bind **Class TypeConstraintException**

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.xml.bind.TypeConstraintException

**All Implemented Interfaces:**
  Serializable

---

public class TypeConstraintException

extends RuntimeException

**Extends:** Throwable > Exception > RuntimeException

Java JAXB

version $Revision: 1.1 $

since JAXB1.0

See also [javax.xml.bind.ValidationEvent](#)

This exception indicates that a violation of a dynamically checked type constraint was detected.

This exception can be thrown by the generated setter methods of the schema derived Java content classes. However, since fail-fast validation is an optional feature for JAXB Providers to support, not all setter methods will throw this exception when a type constraint is violated.

If this exception is throw while invoking a fail-fast setter, the value of the property is guaranteed to remain unchanged, as if the setter were never
called.

Since:
   JAXB1.0
Version:
   $Revision: 1.1 $
Author:
   • Ryan Shoemaker, Sun Microsystems, Inc.
   • Joe Fialli, Sun Microsystems, Inc.
See Also:
   ValidationEvent, Serialized Form

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>TypeConstraintException(String message)</td>
<td>Construct a TypeConstraintException with the specified detail message.</td>
</tr>
<tr>
<td>TypeConstraintException(String message, String errorCode)</td>
<td>Construct a TypeConstraintException with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td>TypeConstraintException(String message, String errorCode, Throwable exception)</td>
<td>Construct a TypeConstraintException with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td>TypeConstraintException(String message, Throwable exception)</td>
<td>Construct a TypeConstraintException with the specified detail message and linkedException.</td>
</tr>
<tr>
<td>TypeConstraintException(Throwable exception)</td>
<td>Construct a TypeConstraintException with a linkedException.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getErrorCode()</td>
<td>Get the vendor specific error code</td>
</tr>
<tr>
<td>Throwable getLinkedException()</td>
<td>Get the linked exception</td>
</tr>
</tbody>
</table>
public TypeConstraintException(String message)
TypeConstraintException errorCode
linkedException
null

message
Construct a TypeConstraintException with the specified detail message. The errorCode and linkedException will default to null.

**Parameters:**
message - a description of the exception

```java
public TypeConstraintException(String message, String errorCode)
errorCode TypeConstraintException linkedException null
message
errorCode
```

TypeConstraintException

Construct a TypeConstraintException with the specified detail message and vendor specific errorCode. The linkedException will default to null.

**Parameters:**
message - a description of the exception
errorCode - a string specifying the vendor specific error code

```java
public TypeConstraintException(Throwable exception)
linkedException TypeConstraintException errorCode null
exception
```

TypeConstraintException
public TypeConstraintException(Throwable exception)

Construct a TypeConstraintException with a linkedException. The detail message and vendor specific errorCode will default to null.

Parameters:
   exception - the linked exception

-----

equality

public TypeConstraintException(String message, Throwable exception)

Construct a TypeConstraintException with the specified detail message and linkedException. The errorCode will default to null.

Parameters:
   message - a description of the exception
   exception - the linked exception

-----

public TypeConstraintException(String message, String errorCode, Throwable exception)

Construct a TypeConstraintException with the specified detail message and linkedException. The errorCode will default to null.

Parameters:
   message - a description of the exception
   errorCode - the vendor specific error code
   exception - the linked exception
TypeConstraintException

public TypeConstraintException(String message, String errorCode, Throwable exception)

Construct a TypeConstraintException with the specified detail message, vendor specific errorCode, and linkedException.

Parameters:
message - a description of the exception
errorCode - a string specifying the vendor specific error code
exception - the linked exception

Method Detail

public String getErrorCode()

    return

getErrorCode

public String getErrorCode()

    Get the vendor specific error code

    Returns:
    a string specifying the vendor specific error code

    public Throwable getLinkedException()

        return null
**getLinkedException**

public Throwable getLinkedException()

Get the linked exception

**Returns:**
the linked Exception, null if none exists

**public void setLinkedException(Throwable exception)**

    exception    null

**setLinkedException**

public void setLinkedException(Throwable exception)

Add a linked Exception.

**Parameters:**
exception - the linked Exception (A null value is permitted and indicates that the linked exception does not exist or is unknown).

**public String toString()**

TypeConstraintException

**toString**

public String toString()

Returns a short description of this TypeConstraintException.

**Overrides:**
toString in class Throwable
public void printStackTrace(java.io.OutputStream s)
TypeConstraintException PrintStream null
linkedException

public void printStackTrace(java.io.PrintStream s)

printStackTrace

public void printStackTrace(java.io.PrintStream s)

Overrides:

printStackTrace in class Throwable

Parameters:

s - PrintStream to use for output

public void printStackTrace()
TypeConstraintException System.err null
linkedException

printStackTrace

public void printStackTrace()

Overrides:

printStackTrace in class Throwable

Parameters:
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>

PREV CLASS  NEXT CLASS
FRAMES  NO FRAMES
javax.xml.rpc.encoding Interface TypeMapping

public interface TypeMapping

javax.xml.rpc.encoding.TypeMapping TypeMapping

TypeMapping {Java, SerializerFactory, DeserializerFactory, XML}

version 1.0

The javax.xml.rpc.encoding.TypeMapping is the base interface for the representation of a type mapping. A TypeMapping implementation class may support one or more encoding styles.

For its supported encoding styles, a TypeMapping instance maintains a set of tuples of the type {Java type, SerializerFactory, DeserializerFactory, XML type}.

Version:
1.0
Author:
Rahul Sharma

Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeserializerFactory</td>
<td>getDeserializer(Class javaType, QName xmlType)</td>
<td>Gets the DeserializerFactory registered for the specified pair of Java type and XML data type.</td>
</tr>
<tr>
<td>SerializerFactory</td>
<td>getSerializer(Class javaType, QName xmlType)</td>
<td>Gets the SerializerFactory registered for the specified pair of Java type and XML data type.</td>
</tr>
<tr>
<td></td>
<td>getSupportedEncodings()</td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><code>String[] getSupportedEncodings()</code></td>
<td>Returns the encodingStyle URIs (as String[]) supported by this TypeMapping instance.</td>
<td></td>
</tr>
<tr>
<td><code>boolean isRegistered(Class javaType, QName xmlType)</code></td>
<td>Checks whether or not type mapping between specified XML type and Java type is registered.</td>
<td></td>
</tr>
<tr>
<td><code>void register(Class javaType, QName xmlType, SerializerFactory sf, DeserializerFactory dsf)</code></td>
<td>Registers SerializerFactory and DeserializerFactory for a specific type mapping between an XML type and Java type.</td>
<td></td>
</tr>
<tr>
<td><code>void removeDeserializer(Class javaType, QName xmlType)</code></td>
<td>Removes the DeserializerFactory registered for the specified pair of Java type and XML data type.</td>
<td></td>
</tr>
<tr>
<td><code>void removeSerializer(Class javaType, QName xmlType)</code></td>
<td>Removes the SerializerFactory registered for the specified pair of Java type and XML data type.</td>
<td></td>
</tr>
<tr>
<td><code>void setSupportedEncodings(String[] encodingStyleURIs)</code></td>
<td>Sets the encodingStyle URIs supported by this TypeMapping instance.</td>
<td></td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public String[] getSupportedEncodings()
TypeMapping encodingStyle URI String[]
Serializer Deserializer TypeMapping
return encodingStyle URI
```

**getSupportedEncodings**

`String[] getSupportedEncodings()`

Returns the encodingStyle URIs (as String[]) supported by this TypeMapping instance. A TypeMapping that contains only encoding style independent serializers and deserializers returns `null` from this.
method.

Returns:
Array of encodingStyle URIs for the supported encoding styles

```java
public void setSupportedEncodings(String[] encodingStyleURIs)
```

Sets the encodingStyle URIs supported by this TypeMapping instance. A TypeMapping that contains only encoding independent serializers and deserializers requires `null` as the parameter for this method.

**Parameters:**
- `encodingStyleURIs` - Array of encodingStyle URIs for the supported encoding styles

```java
public boolean isRegistered(Class<T> javaType, javax.xml.namespace.QName xmlType)
```

**XML Java**
```
javaType           Java
QName             XML
return            boolean XML Java
```

isRegistered
boolean **isRegistered**(Class javaType,
    QName xmlType)

Checks whether or not type mapping between specified XML type and Java type is registered.

**Parameters:**
- javaType - Class of the Java type
- QName - Qualified name of the XML data type

**Returns:**
- boolean; true if type mapping between the specified XML type and Java type is registered; otherwise false

```java
public void register(Class<T> javaType,
    javax.xml.namespace.QName xmlType, SerializerFactory sf,
    DeserializerFactory dsf)
XML  Java  SerializerFactory
DeserializerFactory SerializerFactory

javaType  Java
QName    XML
sf  SerializerFactory
dsf  DeserializerFactory

**Throws**
- JAXRPCException:
```

```java
void register(Class javaType,
    QName xmlType,
    SerializerFactory sf,
    DeserializerFactory dsf)

Registers SerializerFactory and DeserializerFactory for a specific type mapping between an XML type and Java type. This method replaces any existing registered SerializerFactory DeserializerFactory instances.
```
public SerializerFactory getSerializer(Class<T> javaType, javax.xml.namespace.QName xmlType)

Java XML SerializerFactory

javaType Java
QName XML

return SerializerFactory

null

getSerializer

SerializerFactory getSerializer(Class javaType,
QName xmlType)

Gets the SerializerFactory registered for the specified pair of Java type and XML data type.

Parameters:
javaType - Class of the Java type
QName - Qualified name of the XML data type

Returns:
Registered SerializerFactory or null if there is no registered factory

public DeserializerFactory getDeserializer(Class<T> javaType, javax.xml.namespace.QName xmlType)

Java XML DeserializerFactory

javaType Java
getDeseralizer

DeserializerFactory getDeseralizer(Class javaType, QName xmlType)

Gets the DeserializerFactory registered for the specified pair of Java type and XML data type.

Parameters:
  javaType - Class of the Java type
  QName - Qualified name of the XML data type

Returns:
  Registered DeserializerFactory or null if there is no registered factory

public void removeSerializer(Class<T> javaType, javax.xml.namespace.QName xmlType)

Java XML SerializerFactory

Throws JAXRPCException: SerializerFactory

removeSerialser

void removeSerializer(Class javaType, QName xmlType)

Removes the SerializerFactory registered for the specified pair of Java type and XML data type.

Throws:
  JAXRPCException - If there is error in removing the registered SerializerFactory
public void removeDeserializer(Class<T> javaType, javax.xml.namespace.QName xmlType)

Throws JAXRPCException: DeserializerFactory

removeDeserializer

void removeDeserializer(Class javaType, QName xmlType)

Removes the DeserializerFactory registered for the specified pair of Java type and XML data type.

Throws:

JAXRPCException - If there is error in removing the registered DeserializerFactory

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The interface `javax.xml.rpc.encoding.TypeMappingRegistry` defines a registry of TypeMapping instances for various encoding styles.

**Version:**
1.0

**Author:**
Rahul Sharma

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>clear()</code></td>
<td>Removes all registered TypeMappings and encodingStyleURIs from this TypeMappingRegistry.</td>
</tr>
<tr>
<td><code>createTypeMapping()</code></td>
<td>Creates a new empty TypeMapping object.</td>
</tr>
<tr>
<td><code>getDefaultTypeMapping()</code></td>
<td>Gets the registered default TypeMapping instance.</td>
</tr>
<tr>
<td><code>getRegisteredEncodingStyleURIs()</code></td>
<td>Returns a list of registered encodingStyle URIs in this TypeMappingRegistry instance.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>getTypeMapping(String encodingStyleURI)</strong></td>
<td>Returns the registered TypeMapping for the specified encodingStyle URI.</td>
</tr>
<tr>
<td><strong>register(String encodingStyleURI, TypeMapping mapping)</strong></td>
<td>Registers a TypeMapping instance with the TypeMappingRegistry.</td>
</tr>
<tr>
<td><strong>registerDefault(TypeMapping mapping)</strong></td>
<td>Registers the TypeMapping instance that is default for all encoding styles supported by the TypeMappingRegistry.</td>
</tr>
<tr>
<td><strong>removeTypeMapping(TypeMapping mapping)</strong></td>
<td>Removes a TypeMapping from the TypeMappingRegistry.</td>
</tr>
<tr>
<td><strong>unregisterTypeMapping(String encodingStyleURI)</strong></td>
<td>Unregisters a TypeMapping instance, if present, from the specified encodingStyleURI.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public TypeMapping register(String encodingStyleURI, TypeMapping mapping)
```

- **encodingStyleURI**: URI
- **mapping**: TypeMapping
- **return**: TypeMapping

Throws: JAXRPCException

```
TypeMapping register(String encodingStyleURI, TypeMapping mapping)
```
Registers a `TypeMapping` instance with the `TypeMappingRegistry`. This method replaces any existing registered `TypeMapping` instance for the specified `encodingStyleURI`.

**Parameters:**
- `encodingStyleURI` - An encoding style specified as an URI. An example is "http://schemas.xmlsoap.org/soap/encoding/"
- `mapping` - `TypeMapping` instance

**Returns:**
Previous `TypeMapping` associated with the specified `encodingStyleURI`, or `null` if there was no `TypeMapping` associated with the specified `encodingStyleURI`

**Throws:**
- `JAXRPCException` - If there is an error in the registration of the `TypeMapping` for the specified `encodingStyleURI`.

```java
public void registerDefault(TypeMapping mapping)
```

Registers the `TypeMapping` instance that is default for all encoding styles supported by the `TypeMappingRegistry`. A default `TypeMapping` should include serializers and deserializers that are independent of and usable with any encoding style. Successive invocations of the `registerDefault` method replace any existing default `TypeMapping`.
If the default `TypeMapping` is registered, any other `TypeMapping` instances registered through the `TypeMappingRegistry.register` method (for a set of encodingStyle URIs) override the default `TypeMapping`.

**Parameters:**
- `mapping` - `TypeMapping` instance

**Throws:**
- `JAXRPCException` - If there is an error in the registration of the default `TypeMapping`

```java
public `TypeMapping` getDefaultTypeMapping() {
  `TypeMapping` defaultMapping = TypeMappingRegistry.getDefaultTypeMapping();
  return defaultMapping;
}
```

**getRegisteredEncodingStyleURIs**

```java
public `String[]` getRegisteredEncodingStyleURIs()`
  `TypeMappingRegistry` encodingStyleURIs = TypeMappingRegistry.getRegisteredEncodingStyleURIs();
  return encodingStyleURIs;
```

String[] getRegisteredEncodingStyleURIs()

Returns a list of registered encodingStyle URIs in this TypeMappingRegistry instance.

Returns:
Array of the registered encodingStyle URIs

public TypeMapping getTypeMapping(String encodingStyleURI)

Returns the registered TypeMapping for the specified encodingStyle URI. If there is no registered TypeMapping for the specified encodingStyleURI, this method returns null.

Parameters:
encodingStyleURI - Encoding style specified as an URI

Returns:
TypeMapping for the specified encodingStyleURI or null

public TypeMapping createTypeMapping()

Returns the registered TypeMapping for the specified encodingStyle URI. If there is no registered TypeMapping for the specified encodingStyleURI, this method returns null.

Parameters:
encodingStyleURI - Encoding style specified as an URI

Returns:
TypeMapping for the specified encodingStyleURI or null
**TypeMapping** createTypeMapping()

Creates a new empty TypeMapping object.

**Returns:**
TypeMapping instance

---

```java
public TypeMapping unregisterTypeMapping(String encodingStyleURI)
```

Unregisters a TypeMapping instance, if present, from the specified encodingStyleURI.

**Parameters:**
- `encodingStyleURI` - Encoding style specified as an URI

**Returns:**
TypeMapping instance that has been unregistered or null if there was no TypeMapping registered for the specified encodingStyleURI

---

```java
public boolean removeTypeMapping(TypeMapping mapping)
```

```java
return TypeMappingRegistry encodingStyleURI TypeMapping encodingStyleURIs Ty TypeMapping
```

```java
return TypeMappingRegistry TypeMapping true TypeMapping
```
removeTypeMapping

boolean removeTypeMapping(TypeMapping mapping)

Removes a TypeMapping from the TypeMappingRegistry. A TypeMapping is associated with 1 or more encodingStyleURIs. This method unregisters the specified TypeMapping instance from all associated encodingStyleURIs and then removes this TypeMapping instance from the registry.

Parameters:
  mapping - TypeMapping to be removed

Returns:
  true if specified TypeMapping is removed from the TypeMappingRegistry; false if the specified TypeMapping was not in the TypeMappingRegistry

public void clear()

Removes all registered TypeMappings and encodingStyleURIs from this TypeMappingRegistry.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.component  Class UIColumn

java.lang.Object  
  | javax.faces.component.UIComponent  
  |   | javax.faces.component.UIComponentBase  
  |   | javax.faces.component.UIColumn

All Implemented Interfaces:
  StateHolder

Direct Known Subclasses:
  HtmlColumn

public class UIColumn
  extends UIComponentBase

Extends: UIComponent > UIComponentBase
Extended by: HtmlColumn

UIColumn  UIData  UICOMPONENT

UIColumn is a UICOMPONENT that represents a single column of data within a parent UIData component.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
</table>
| static String COMPONENT_FAMILY  
  The standard component family for this component. |
| static String COMPONENT_TYPE  
  The standard component type for this component. |

Fields inherited from class javax.faces.component.UIComponent
### Constructor Summary

**UIColumn()**
Create a new `UIColumn` instance with default property values.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getFamily()</td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td>UIColumn</td>
<td>getFooter()</td>
<td>Return the footer facet of the column (if any).</td>
</tr>
<tr>
<td>UIColumn</td>
<td>getHeader()</td>
<td>Return the header facet of the column (if any).</td>
</tr>
<tr>
<td>void</td>
<td>setFooter(UIColumn footer)</td>
<td>Set the footer facet of the column.</td>
</tr>
<tr>
<td>void</td>
<td>setHeader(UIColumn header)</td>
<td>Set the header facet of the column.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildren`, `getComponent`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueExpression`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `restoreState`, `saveAttachedState`, `saveState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueExpression`

### Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`
Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UIColumn()

UIColumn
**UIColumn**

```java
public UIColumn()
```

Create a new `UIColumn` instance with default property values.

## Method Detail

**public String getFamily()**

### getFamily

```java
public String getFamily()
```

**Description copied from class: `UIComponent`**

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

**Specified by:**

`getFamily` in class `UIComponent`

```java
public UIColumn getFooter()
```

### facet getFacet("footer")

**getFooter**

```java
public UIColumn getFooter()
```
Return the footer facet of the column (if any). A convenience method for getFacet("footer").

```java
public void setFooter(UIComponent footer)

facet getFacets().put("footer", footer)

footer facet

Throws NullPointerException: footer null
```

**setFooter**

```java
public void setFooter(UIComponent footer)

Set the footer facet of the column. A convenience method for getFacets().put("footer", footer).

Parameters:
footer - the new footer facet

Throws:
NullPointerException - if footer is null
```

```java
public UIComponent getHeader()

facet getFacet("header")
```

**getHeader**

```java
public UIComponent getHeader()

Return the header facet of the column (if any). A convenience method for getFacet("header").
```
public void setHeader(UIComponent header)

facet getFacets().put("header", header)

Throws NullPointerException: header null

setHeader

public void setHeader(UIComponent header)

Set the header facet of the column. A convenience method for getFacets().put("header", header).

Parameters:
  header - the new header facet

Throws:
  NullPointerException - if header is null

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.component Class UICommand

java.lang.Object
   ↘ javax.faces.component.UIComponent
      ↘ javax.faces.component.UIComponentBase
         ↘ javax.faces.component.UICommand

All Implemented Interfaces:
   ActionSource, ActionSource2, StateHolder

Direct Known Subclasses:
   HtmlCommandButton, HtmlCommandLink

public class UICommand
extends UIComponentBase
implements ActionSource2

Extends: UIComponent > UIComponentBase
Implements: ActionSource2
Extended by: HtmlCommandButton, HtmlCommandLink

UICommand  UIComponent "command" "action"

  decode()  Renderer  ActionEvent
  broadcast()

1. ActionListener
2. "actionListener" MethodExpression  MethodBinding
   "actionListener"
3. Application  ActionListener - "action" MethodExpression

rendererType "javax.faces.Button"  setRendererType()
**UICommand** is a **UIComponent** that represents a user interface component which, when activated by the user, triggers an application specific "command" or "action". Such a component is typically rendered as a push button, a menu item, or a hyperlink.

When the `decode()` method of this **UICommand**, or its corresponding **Renderer**, detects that this control has been activated, it will queue an **ActionEvent**. Later on, the `broadcast()` method will ensure that this event is broadcast to all interested listeners.

Listeners will be invoked in the following order:

1. **ActionListener**, in the order in which they were registered.
2. The "actionListener" **MethodExpression** (which will cover the "actionListener" that was set as a MethodBinding).
3. The default **ActionListener**, retrieved from the **Application** - and therefore, any attached "action" **MethodExpression**.

By default, the `rendererType` property must be set to "javax.faces.Button". This value can be changed by calling the `setRendererType()` method.

### Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>COMPONENT_FAMILY</th>
<th>The standard component family for this component.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.faces.component.UIComponent bindings

### Constructor Summary

**UICommand** ()

Create a new **UICommand** instance with default property values.
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addActionListener(ActionListener listener)</code></td>
<td>Add a new <code>ActionListener</code> to the set of listeners interested in being notified when <code>ActionEvent</code> s occur.</td>
</tr>
<tr>
<td><code>void broadcast(FacesEvent event)</code></td>
<td>In addition to the default <code>UIComponent.broadcast(javax.faces.event.FacesEvent)</code> processing, pass the <code>ActionEvent</code> being broadcast to the method referenced by <code>actionListener</code> (if any), and to the default <code>ActionListener</code> registered on the <code>Application</code>.</td>
</tr>
<tr>
<td><code>MethodBinding/getAction()</code></td>
<td><strong>Deprecated.</strong> This has been replaced by <code>getActionExpression()</code>.</td>
</tr>
<tr>
<td><code>MethodExpression/getActionExpression()</code></td>
<td>Return the <code>MethodExpression</code> pointing at the application action to be invoked, if this <code>UIComponent</code> is activated by the user, during the <code>Apply Request Values</code> or <code>Invoke Application</code> phase of the request processing lifecycle, depending on the value of the <code>immediate</code> property.</td>
</tr>
<tr>
<td><code>MethodBinding/getActionListener()</code></td>
<td><strong>Deprecated.</strong> Use <code>getActionListeners()</code> instead.</td>
</tr>
<tr>
<td><code>ActionListener[]/getActionListeners()</code></td>
<td>Return the set of registered <code>ActionListener</code> s for this <code>ActionSource</code> instance.</td>
</tr>
<tr>
<td><code>String/getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>Object/getValue()</code></td>
<td>Returns the <code>value</code> property of the <code>UICommand</code>.</td>
</tr>
</tbody>
</table>
| `boolean/isImmediate()` | Return a flag indicating that the default `ActionListener` provided by the JavaServer Faces implementation should be executed immediately (that is, during `Apply Request Values` phase of the request processing lifecycle), rather than waiting until the `Invoke`...
**Application phase.**

```java
void queueEvent(FacesEvent e)
    Intercept queueEvent and, for ActionEvents, mark the phaseld for the event to be PhaseId.APPLY_REQUEST_VALUES if the immediate flag is true, PhaseId.INVOKE_APPLICATION otherwise.
```

```java
void removeActionListener(ActionListener listener)
    Remove an existing ActionListener (if any) from the set of listeners interested in being notified when ActionEvents occur.
```

```java
void restoreState(FacesContext context, Object state)
    Perform any processing required to restore the state from the entries in the state Object.
```

```java
Object saveState(FacesContext context)
    Gets the state of the instance as a Serializable Object.
```

```java
void setAction(MethodBinding action)
    Deprecated. This has been replaced by setActionExpression(javax.el.MethodExpression).
```

```java
void setActionExpression(MethodExpression actionExpression)
    Set the MethodExpression pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.
```

```java
void setImmediate(boolean immediate)
    Set the "immediate execution" flag for this UIComponent.
```

```java
void setValue(Object value)
    Sets the value property of the UICommand.
```

**Methods inherited from class**
javax.faces.component.UIComponentBase
addFacesListener, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY
The standard component family for this component.

See Also:

Constant Field Values

Constructor Detail

public UICommand()

UICommand

UICommand

public UICommand()

Create a new UICommand instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:

getFamily in class UIComponent
public MethodBinding getAction()

NullPointerException

ActionSource2 ActionSource2#getActionExpression
#setAction MethodBinding MethodBinding
javax.el.MethodExpression

UIComponent MethodBinding MethodBinding
immediate

deprecated #getActionExpression

getAction

public MethodBinding getAction()

Deprecated. This has been replaced by getActionExpression().

If the implementing class also implements ActionSource2, the implementation of this method must call through to ActionSource2.getActionExpression() and examine the result. If the result came from a previous call to ActionSource2.setAction(javax.faces.el.MethodBinding), extract the MethodBinding from it and return it. Otherwise, wrap the returned MethodExpression in a MethodBinding implementation, and return it.

If the implementing class does not implement ActionSource2, return the MethodBinding pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

Specified by:
    getAction in interface ActionSource
public void setAction(MethodBinding action)

NullPointerException

ActionSource2 javax.el.MethodExpression action

ActionSource2#setActionExpression action

ActionSource2 MethodBinding MethodBinding

UIComponent immediate

String

deprecated #setActionExpression(javax.el.MethodExpression)

setAction

public void setAction(MethodBinding action)

Deprecated. This has been replaced by


If the implementing class also implements ActionSource2, the implementation of this method must wrap the argument action in a class that implements MethodExpression and call through to

ActionSource2.setActionExpression(javax.el.MethodExpression),

passing the wrapped action.

If the implementing class does not implement ActionSource2, set the MethodBinding pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

Any method referenced by such an expression must be public, with a return type of String, and accept no parameters.

Specified by:

setAction in interface ActionSource
Parameters:
  action - The new MethodBinding expression

public MethodBinding getActionListener()

NullPointerException

#setActionListener null #setActionListener
#setActionListener MethodBinding

UIComponent immediate

deprecated #getActionListeners

getActionListener

public MethodBinding getActionListener()

Deprecated. Use getActionListeners() instead.

If ActionSource.setActionListener(javax.faces.el.MethodBinding) was not previously called for this instance, this method must return null. If it was called, this method must return the exact MethodBinding instance that was passed to ActionSource.setActionListener(javax.faces.el.MethodBinding).

The method to be invoked, if this UIComponent is activated by the user, will be called during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending upon the value of the immediate property.

Specified by:
  getActionListener in interface ActionSource

public void setActionListener(MethodBinding actionListener)
NullPointerException

**ActionListener**  ActionListener  #getActionListeners

**setActionListener**

```java
public void setActionListener(MethodBinding actionListener)
```

**Deprecated. This has been replaced by**

```java
addActionListener(javax.faces.event.ActionListener)
```

Wrap the argument `actionListener` in an implementation of `ActionListener` and store it in the internal data structure that backs the `ActionSource.getActionListeners()` method, taking care to overwrite any instance that was stored by a previous call to `setActionListener`.

Any method referenced by such an expression must be public, with a return type of `void`, and accept a single parameter of type `ActionEvent`.

**Specified by:**

`setActionListener` in interface `ActionSource`

**Parameters:**

- `actionListener` - The new method binding expression

---

**public boolean isImmediate()**

**isImmediate**
public boolean isImmediate()

Description copied from interface: ActionSource

Return a flag indicating that the default ActionListener provided by the JavaServer Faces implementation should be executed immediately (that is, during Apply Request Values phase of the request processing lifecycle), rather than waiting until the Invoke Application phase. The default value for this property must be false.

Specified by:
   isImmediate in interface ActionSource

public void setImmediate(boolean immediate)

setImmediate

public void setImmediate(boolean immediate)

Description copied from interface: ActionSource

Set the "immediate execution" flag for this UIComponent.

Specified by:
   setImmediate in interface ActionSource

Parameters:
   immediate - The new immediate execution flag

public Object getValue()

UICommand   value

getValue
public Object getValue()

    Returns the value property of the UICommand. This is most often rendered as a label.

    public void setValue(Object value)

        UICommand value

            value

    setValue

    public void setValue(Object value)

        Sets the value property of the UICommand. This is most often rendered as a label.

        Parameters:
            value - the new value

    public MethodExpression getActionExpression()

    getActionExpression

    public MethodExpression getActionExpression()

        Description copied from interface: ActionSource2

        Return the MethodExpression pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.
Note that it's possible that the returned MethodExpression is just a wrapper around a MethodBinding instance which was set by a call to ActionSource.setAction(javax.faces.el.MethodBinding). This makes it possible for the default ActionListener to continue to work properly with older components.

Specified by:
getActionExpression in interface ActionSource2

public void setActionExpression(MethodExpression actionExpression)

setActionExpression

public void setActionExpression(MethodExpression actionExpression)

Description copied from interface: ActionSource2

Set the MethodExpression pointing at the application action to be invoked, if this UIComponent is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

Any method referenced by such an expression must be public, with a return type of String, and accept no parameters.

Specified by:
setActionExpression in interface ActionSource2

Parameters:
actionExpression - The new method expression

public void addActionListener(ActionListener listener)

Throws NullPointerException: NullPointerException

listener null
**addActionListener**

public void addActionListener(ActionListener listener)

**Description copied from interface: ActionSource**

Add a new ActionListener to the set of listeners interested in being notified when ActionEvent s occur.

**Specified by:**

addActionListener in interface ActionSource

**Parameters:**

listener - The ActionListener to be added

**Throws:**

NullPointerException - if listener is null

**public ActionListener[] getActionListeners()**

**getActionListeners**

public ActionListener[] getActionListeners()

**Description copied from interface: ActionSource**

Return the set of registered ActionListeners for this ActionSource instance. If there are no registered listeners, a zero-length array is returned.

**Specified by:**

getActionListeners in interface ActionSource

**public void removeActionListener(ActionListener listener)**

**Throws**

NullPointerException: NullPointerException

**removeActionListener**
public void removeActionListener(ActionListener listener)

Description copied from interface: ActionSource

Remove an existing ActionListener (if any) from the set of listeners interested in being notified when ActionEventS occur.

Specified by:
   removeActionListener in interface ActionSource

Parameters:
   listener - The ActionListener to be removed

Throws:
   NullPointerException - if listener is null

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);
component should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIComponentBase`

---

```java
public void restoreState(FacesContext context, Object state)
```

**restoreState**

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface:** `StateHolder`

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
- `restoreState` in interface `StateHolder`

**Overrides:**
- `restoreState` in class `UIComponentBase`

---

```java
public void broadcast(FacesEvent event) throws AbortProcessingException
```
**broadcast**

```java
public void broadcast(FacesEvent event)
throws AbortProcessingException
```

In addition to the default `UIComponent.broadcast(javax.faces.event.FacesEvent)` processing, pass the `ActionEvent` being broadcast to the method referenced by `actionListener` (if any), and to the default `ActionListener` registered on the `Application`.

**Overrides:**
- `broadcast` in class `UIComponentBase`

**Parameters:**
- `event` - `FacesEvent` to be broadcast

**Throws:**
- `AbortProcessingException` - Signal the JavaServer Faces implementation that no further processing on the current event should be performed
- `IllegalArgumentException` - if the implementation class of this `FacesEvent` is not supported by this component
- `NullPointerException` - if `event` is `null`

---

**queueEvent**

```java
public void queueEvent(FacesEvent e)
```

- `ActionEvent queueEvent immediate true phaseId
  PhaseId.APPLY_REQUEST_VALUES PhaseId.INVOKE_APPLICATION`
queueEvent

public void queueEvent(FacesEvent e)

    Intercept queueEvent and, for ActionEvents, mark the phaseId for the event to be PhaseId.APPLY_REQUEST_VALUES if the immediate flag is true, PhaseId.INVOKE_APPLICATION otherwise.

Overrides:
    queueEvent in class UIComponentBase

Parameters:
    e - FacesEvent to be queued
UIComponent is the base class for all user interface components in JavaServer Faces. The set of UIComponent instances associated with a particular request and response are organized into a component tree under a UIViewRoot that represents the entire content of the request or response.

For the convenience of component developers, UICOMPONENTBase provides the default behavior that is specified for a UIComponent, and is the base class for all of the concrete UIComponent "base" implementations. Component writers are encouraged to subclass UICOMPONENTBase, instead
of directly implementing this abstract class, to reduce the impact of any future changes to the method signatures.

### Field Summary

<table>
<thead>
<tr>
<th>Protected</th>
<th>bindings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Map&lt;String, ValueExpression&gt;</td>
<td></td>
</tr>
</tbody>
</table>

### Constructor Summary

**UIComponent()**

### Method Summary

<table>
<thead>
<tr>
<th>Protected abstract void</th>
<th>addFacesListener(FacesListener listener)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add the specified FacesListener to the set of listeners that receive event notifications from this UIComponent.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstract void</th>
<th>broadcast(FacesEvent event)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast the specified FacesEvent to all registered listeners who have expressed an interest in events of this type.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstract void</th>
<th>decode(FacesContext context)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Decode any new state of this UIComponent from the input source in the specified FacesContext, and store this state as needed.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Void</th>
<th>encodeAll(FacesContext context)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If this component returns true from isRendered and all its children that return true from isRendered of the value of the getRendersChildren() flag.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstract void</th>
<th>encodeBegin(FacesContext context)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If our rendered property is true, render the beginning state of this UIComponent to the response contained in FacesContext.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Abstract void</th>
<th>encodeChildren(FacesContext context)</th>
</tr>
</thead>
<tbody>
<tr>
<td>If our rendered property is true, render the children of this UIComponent.</td>
<td></td>
</tr>
</tbody>
</table>
abstract void encodeEnd(FacesContext context)

If our rendered property is true, render the end of this UIComponent.

abstract UIComponent findComponent(String expr)

Search for and return the UIComponent with an specified search expression (if any), according to the below.

abstract Map<String, Object> getAttributes()

Return a mutable Map representing the attribute below) associated with this UIComponent, keyed by attribute a String).

abstract int getChildCount()

Return the number of child UIComponent s that a UIComponent.

abstract List<UIComponent> getChildren()

Return a mutable List representing the child UIComponent with this component.

abstract String getClientId(FacesContext context)

Return a client-side identifier for this component, generating necessary.

abstract String getContainerClientId(FacesContext context)

Allow components that implement NamingConta disable prepending their clientId to their descendent's the prepending logic into a separately callable method.

protected abstract FacesContext getFacesContext()

Convenience method to return the current request.

protected abstract FacesListener[] getFacesListeners(Class clazz)

Return an array of registered FacesListener s that specified class.

abstract UIComponent getFacet(String name)

Convenience method to return the named face otherwise.

abstract int getFacetCount()

Return the number of facet UIComponent s that a UIComponent.
abstract Map<String, UIComponent> getFacets()  
Return a mutable Map representing the facet UIComponent with this UIComponent, keyed by facet name (which must be a String).

abstract Iterator<UIComponent> getFacetsAndChildren()  
Return an Iterator over the facet followed by children this UIComponent.

abstract String getFamily()  
Return the identifier of the component family to which this belongs.

abstract String getId()  
Return the component identifier of this UIComponent.

abstract UIComponent getParent()  
Return the parent UIComponent of this UIComponent.

protected abstract Renderer getRenderer(FacesContext context)  
Convenience method to return the Renderer instance this component, if any; otherwise, return null.

abstract String getRendererType()  
Return the Renderer type for this UIComponent (if any).

abstract boolean getRendersChildren()  
Return a flag indicating whether this component is responsible for rendering its child components.

abstract ValueBinding getValueBinding(String name)  
Deprecated. This has been replaced by
getValueExpression(java.lang.String).

abstract ValueExpression getValueExpression(String name)  
Return the ValueExpression used to calculate the specified attribute or property name, if any.

boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback)  
Starting at this component in the View hierarchy, search for a component with a clientId equal to the argument clientId, then call the
ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the
passing the current FacesContext and the found component.

boolean isRendered()
abstract boolean processDecodes(FacesContext context)
Return true if this component (and its children) should be "decoded" during the Render Response phase of the request processing lifecycle.

abstract void processDecodes(FacesContext context)
Perform the component tree processing required by the Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

abstract void processRestoreState(FacesContext context, Object state)
Perform the component tree processing required by the state portion of the Render Response phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

abstract Object processSaveState(FacesContext context)
Perform the component tree processing required by the Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

abstract void processUpdates(FacesContext context)
Perform the component tree processing required by the Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

abstract void processValidators(FacesContext context)
Perform the component tree processing required by the Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

abstract void queueEvent(FacesEvent event)
Queue an event for broadcast at the end of the request processing lifecycle phase.

protected abstract void removeFacesListener(FacesListener listener)
Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

abstract void setId(String id)
Set the component identifier of this UIComponent.

abstract void setParent(UIComponent parent)
Set the parent UIComponent of this UIComponent.
abstract void setRendered(boolean rendered)
    Set the rendered property of this UIComponent.

abstract void setRendererType(String rendererType)
    Set the Renderer type for this UIComponent, or null to
    render themselves.

abstract void setValueBinding(String name, ValueBinding binding)
    Deprecated. This has been replaced by
    setValueExpression(java.lang.String, javax.el.ValueExpression)

void setValueExpression(String name, ValueExpression binding)
    Set the ValueExpression used to calculate the value
    attribute or property name, if any.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

Methods inherited from interface javax.faces.component.StateHolder
isTransient, restoreState, saveState, setTransient

Field Detail

bindings

protected Map<String, ValueExpression> bindings

Constructor Detail

public UIComponent()
public UIComponent()

Method Detail

abstract public java.util.Map<K, V> getAttributes()

UIComponent Map String Map

- Map java.io.Serializable
- null NullPointerException
- String ClassCastException
- UIComponent
  - containsKey - false
  - get() - IllegalArgumentException
  - put() - null
    IllegalArgumentException
  - remove - IllegalArgumentException

getAttributes

public abstract Map<String, Object> getAttributes()

Return a mutable Map representing the attributes (and properties, see below) associated with this UIComponent, keyed by attribute name (which must be a String). The returned implementation must support all of the standard and optional Map methods, plus support the following additional requirements:

- The Map implementation must implement the java.io.Serializable interface.
- Any attempt to add a null key or value must throw a NullPointerException.
- Any attempt to add a key that is not a String must throw a
ClassCastException.

- If the attribute name specified as a key matches a property of this UIClassComponent's implementation class, the following methods will have special behavior:
  - containsKey - Return false.
  - get() - If the property is readable, call the getter method and return the returned value (wrapping primitive values in their corresponding wrapper classes); otherwise throw IllegalArgumentException.
  - put() - If the property is writeable, call the setter method to set the corresponding value (unwrapping primitive values in their corresponding wrapper classes). If the property is not writeable, or an attempt is made to set a property of primitive type to null, throw IllegalArgumentException.
  - remove - THROW IllegalArgumentException.

abstract public ValueBinding getValueBinding(String name)

<table>
<thead>
<tr>
<th>#getValueExpression</th>
<th>#setValueBinding</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueBinding</td>
<td>ValueBinding</td>
</tr>
</tbody>
</table>

name

<table>
<thead>
<tr>
<th>name</th>
<th>ValueBinding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>NullPointerException: name null</td>
</tr>
<tr>
<td>deprecated</td>
<td>#getValueExpression</td>
</tr>
</tbody>
</table>

getValueBinding

public abstract ValueBinding getValueBinding(String name)

**Deprecated. This has been replaced by** getValueExpression(java.lang.String).

Call through to getValueExpression(java.lang.String) and examine the result. If the result is an instance of the wrapper class mandated in setValueBinding(java.lang.String).
javax.faces.el.ValueBinding), extract the ValueBinding instance and return it. Otherwise, wrap the result in an implementation of ValueBinding, and return it.

Parameters:
- name - Name of the attribute or property for which to retrieve a ValueBinding

Throws:
- NullPointerException - if name is null

abstract public void setValueBinding(String name, ValueBinding binding)

Parameters:
- name - Name of the attribute or property for which to set a ValueBinding
- binding - The ValueBinding to set, or null to remove any

setValueBinding

public abstract void setValueBinding(String name, ValueBinding binding)

Deprecated. This has been replaced by setValueExpression(java.lang.String, javax.el.ValueExpression).

Wrap the argument binding in an implementation of ValueExpression and call through to setValueExpression(java.lang.String, javax.el.ValueExpression).

Parameters:
- name - Name of the attribute or property for which to set a ValueBinding
- binding - The ValueBinding to set, or null to remove any
currently set `ValueBinding`

**Throws:**
- `IllegalArgumentException` - if name is one of `id` or `parent`
- `NullPointerException` - if name is `null`

---

```java
public ValueExpression getValueExpression(String name)
```

**ValueExpression**

**JSF 1.2**

```java
since 1.2

name ValueExpression

Throws

NullPointerException: name null
```

---

```java
public ValueExpression getValueExpression(String name)
```

**Return the `ValueExpression` used to calculate the value for the specified attribute or property name, if any.**

This method must be overridden and implemented for components that comply with JSF 1.2 and later.

**Parameters:**
- `name` - Name of the attribute or property for which to retrieve a `ValueExpression`

**Throws:**
- `NullPointerException` - if name is `null`

**Since:**
- 1.2

---

```java
public void setValueExpression(String name, ValueExpression binding)
```
Set the `ValueExpression` used to calculate the value for the specified attribute or property name, if any.

The implementation must call `Expression.isLiteralText()` on the argument expression. If `isLiteralText()` returns true, invoke `ValueExpression.getValue(javax.el.ELContext)` on the argument expression and pass the result as the `value` parameter in a call to `this.#getAttributes().put(name, value)` where `name` is the argument name. If an exception is thrown as a result of calling `ValueExpression.getValue(javax.el.ELContext)`, wrap it in a `FacesException` and re-throw it. If `isLiteralText()` returns false, simply store the un-evaluated expression argument in the collection of `ValueExpression`s under the key given by the argument name.
This method must be overridden and implemented for components that comply with JSF 1.2 and later.

Parameters:
- name - Name of the attribute or property for which to set a ValueExpression
- binding - The ValueExpression to set, or null to remove any currently set ValueExpression

Throws:
- IllegalArgumentException - if name is one of id or parent
- NullPointerException - if name is null

Since: 1.2

```
abstract public String getClientId(FacesContext context)

Renderer clientId

id NamingContainer ID UIData
clientId

NamingContainer getContainerClientId() parentId
#if getID myId myId null
context.getViewRoot().createUniqueId() myId parentId
null myId parentId + NamingContainer.SEPARATOR_CHAR +
myId Renderer#convertClientId myId

context FacesContext
Throws
NullPointerExcep

getClientId

public abstract String getClientId(FacesContext context)

return a client-side identifier for this component, generating one if
necessary. The associated Renderer, if any, will be asked to convert the clientId to a form suitable for transmission to the client.

The return from this method must be the same value throughout the lifetime of the instance, unless the id property of the component is changed, or the component is placed in a NamingContainer whose client ID changes (for example, UIData). However, even in these cases, consecutive calls to this method must always return the same value. The implementation must follow these steps in determining the clientId:

Find the closest ancestor to this component in the view hierarchy that implements NamingContainer. Call getContainerClientId() on it and save the result as the parentId local variable. Call getId() on this component and save the result as the myId local variable. If myId is null, call context.getViewRoot().createUniqueId() and assign the result to myId. If parentId is non-null, let myId equal parentId + NamingContainer.SEPARATOR_CHAR + myId. Call Renderer.convertClientId(javax.faces.context.FacesContext, java.lang.String), passing myId, and return the result.

Parameters:
context - The FacesContext for the current request

Throws:
NullPointerException - if context is null

public String getContainerClientId(FacesContext context)

NamingContainer clientId clientId

#getClientId

since 1.2

Throws _NullPointerException: context null
**getContainerClientId**

```java
public String getContainerClientId(FacesContext context)
```

Allow components that implement `NamingContainer` to selectively disable prepending their clientId to their descendent's clientIds by breaking the prepending logic into a separately callable method. See `getClientId(javax.faces.context.FacesContext)` for usage.

By default, this method will call through to `getClientId(javax.faces.context.FacesContext)` and return the result.

**Throws:**
- `NullPointerException` - if context is null

**Since:**
1.2

---

**abstract public String getFamily()**

```java
rendererType Renderer
```

**getFamily**

```java
public abstract String getFamily()
```

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the `rendererType` property, may be used to select the appropriate `Renderer` for this component instance.

---

**abstract public String getId()**

```java
UIComponent
```
**getID**

```java
gpublic abstract String getId()
```

Return the component identifier of this UIComponent.

**abstract public void setId(String id)**

**UIComponent**

- 0 String
- ('_')
- ('_') ('-')

```java
setId()
```

- **facet** NamingContainer UIComponent

```java
NamingContainer
```

**Throws** IllegalArgumentException: id

**setID**

```java
public abstract void setId(String id)
```

Set the component identifier of this UIComponent (if any). Component identifiers must obey the following syntax restrictions:

- Must not be a zero-length String.
- First character must be a letter or an underscore ('_').
- Subsequent characters must be a letter, a digit, an underscore
Component identifiers must also obey the following semantic restrictions (note that this restriction is **NOT** enforced by the `setId()` implementation):

- The specified identifier must be unique among all the components (including facets) that are descendents of the nearest ancestor `UIComponent` that is a `NamingContainer`, or within the scope of the entire component tree if there is no such ancestor that is a `NamingContainer`.

**Parameters:**

- `id` - The new component identifier, or `null` to indicate that this `UIComponent` does not have a component identifier

**Throws:**

- `IllegalArgumentException` - if `id` is not syntactically valid

---

**abstract public** `UIComponent` `getParent()`

```java
UIComponent getParent()
```

null

**getParent**

**public abstract** `UIComponent` `getParent()`

Return the parent `UIComponent` of this `UIComponent`, if any. A component must allow child components to be added to and removed from the list of children of this component, even though the child component returns `null` from `getParent()`.  

---

**abstract public void** `setParent(UIComponent parent)`
UIComponent  UIComponent  List facet  Map

UIComponent

parent  null

setParent

public abstract void setParent(UIComponent parent)

Set the parent UIComponent of this UIComponent. This method must never be called by developers; a UIComponent's internal implementation will call it as components are added to or removed from a parent's child List or facet Map.

Parameters:
  parent - The new parent, or null for the root node of a component tree

-----------------------------------------------

abstract public boolean isRendered()

true

-----------------------------------------------

isRendered

public abstract boolean isRendered()

Return true if this component (and its children) should be rendered during the Render Response phase of the request processing lifecycle.

-----------------------------------------------

abstract public void setRendered(boolean rendered)

UIComponent  rendered
setRendered

public abstract void setRendered(boolean rendered)

Set the rendered property of this UIComponent.

Parameters:
rendered - If true render this component; otherwise, do not render this component

abstract public String getRendererType()

UIComponent      Renderer

getRendererType

public abstract String getRendererType()

Return the Renderer type for this UIComponent (if any).

abstract public void setRendererType(String rendererType)

UIComponent      Renderer      null

rendererType      Renderer      null

setRendererType

public abstract void setRendererType(String rendererType)
Set the Renderer type for this UIComponent, or null for components that render themselves.

Parameters:
 rendererType - Logical identifier of the type of Renderer to use, or null for components that render themselves.

abstract public boolean getRendersChildren()

Return a flag indicating whether this component is responsible for rendering its child components. The default implementation in UIComponentBase.getRendersChildren() tries to find the renderer for this component. If it does, it calls Renderer.getRendersChildren() and returns the result. If it doesn't, it returns false. As of version 1.2 of the JavaServer Faces Specification, component authors are encouraged to return true from this method and rely on UIComponentBase.encodeChildren(javax.faces.context.FacesContext).

abstract public java.util.List<E> getChildren()
getChildren

public abstract List<UIComponent> getChildren()

Return a mutable List representing the child UIComponents associated with this component. The returned implementation must support all of the standard and optional List methods, plus support the following additional requirements:

- The List implementation must implement the java.io.Serializable interface.
- Any attempt to add a null must throw a NullPointerException.
- Any attempt to add an object that does not implement UIComponent must throw a ClassCastException.
- Whenever a new child component is added, the parent property of the child must be set to this component instance. If the parent property of the child was already non-null, the child must first be removed from its previous parent (where it may have been either a child or a facet).
- Whenever an existing child component is removed, the parent property of the child must be set to null.

abstract public int getChildCount()

UIComponent UIComponent 0

getchildCount

public abstract int getChildCount()
Return the number of child UIComponents that are associated with this UIComponent. If there are no children, this method must return 0. The method must not cause the creation of a child component list.

abstract public UIComponent findComponent(String expr)

id   UIComponent

clientId   #invokeOnComponent

NamingContainer   NamingContainer
NamingContainer   NamingContainer

UIComponent   id
NamingContainer#SEPARATOR_CHAR

- UIComponent
  - ""   UIComponent"
  - UIComponent   NamingContainer
  - NamingContainer
  - NamingContainer   UIComponent

- ""
  - id   UIComponent   facet
  - NamingContainer   facet
  - NamingContainer   NamingContainer

findComponent()

expr   UIComponent
return   UIComponent   null

Throws IllegalArgumentException: NamingContainer
Throws NullPointerException: expr   null

findComponent
public abstract void findComponent(String expr)

    Search for and return the UIComponent with an id that matches the specified search expression (if any), according to the algorithm described below.

    For a method to find a component given a simple clientId, see invokeOnComponent(javax.faces.context.FacesContext, java.lang.String, javax.faces.component.ContextCallback).

    Component identifiers are required to be unique within the scope of the closest ancestor NamingContainer that encloses this component (which might be this component itself). If there are no NamingContainer components in the ancestry of this component, the root component in the tree is treated as if it were a NamingContainer, whether or not its class actually implements the NamingContainer interface.

    A search expression consists of either an identifier (which is matched exactly against the id property of a UIComponent, or a series of such identifiers linked by the NamingContainer.SEPARATOR_CHAR character value. The search algorithm should operate as follows, though alternate algorithms may be used as long as the end result is the same:

    - Identify the UIComponent that will be the base for searching, by stopping as soon as one of the following conditions is met:
      - If the search expression begins with the the separator character (called an "absolute" search expression), the base will be the root UIComponent of the component tree. The leading separator character will be stripped off, and the remainder of the search expression will be treated as a "relative" search expression as described below.
      - Otherwise, if this UIComponent is a NamingContainer it will serve as the basis.
      - Otherwise, search up the parents of this component. If a NamingContainer is encountered, it will be the base.
      - Otherwise (if no NamingContainer is encountered) the root UIComponent will be the base.
    - The search expression (possibly modified in the previous step)
is now a "relative" search expression that will be used to locate the component (if any) that has an id that matches, within the scope of the base component. The match is performed as follows:

- If the search expression is a simple identifier, this value is compared to the id property, and then recursively through the facets and children of the base UIComponent (except that if a descendant NamingContainer is found, its own facets and children are not searched).
- If the search expression includes more than one identifier separated by the separator character, the first identifier is used to locate a NamingContainer by the rules in the previous bullet point. Then, the findComponent() method of this NamingContainer will be called, passing the remainder of the search expression.

Parameters:
expr - Search expression identifying the UIComponent to be returned

Returns:
the found UIComponent, or null if the component was not found.

Throws:
IllegalArgumentException - if an intermediate identifier in a search expression identifies a UIComponent that is not a NamingContainer
NullPointerException - if expr is null

public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback) throws FacesException

clientId clientId callback
ContextCallback#invokeContextCallback FacesContext
#findComponent NamingContainer#SEPARATOR_CHAR

this.getClientId() clientId callback
private void doFind(FacesContext context, String clientId) {
    context.getViewRoot().invokeOnComponent(context, clientId,
        new ContextCallback() {
            public void invokeOnComponent(FacesContext context,
                UIComponent component)
            {
                found = component;
            }
        });
}
**invokeOnComponent**

public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback) throws FacesException

Starting at this component in the View hierarchy, search for a component with a clientId equal to the argument clientId and, if found, call the ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the argument callback, passing the current FacesContext and the found component as arguments. This method is similar to findComponent(java.lang.String) but it does not support the leading NamingContainer.SEPARATOR_CHAR syntax for searching from the root of the View.

The default implementation will first check if this.getClientId() is equal to the argument clientId. If so, call the ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the argument callback, passing through the FacesContext argument and passing this as the component argument. If an Exception is thrown by the callback, wrap it in a FacesException and re-throw it. Otherwise, return true.

Otherwise, for each component returned by getFacetsAndChildren(), call invokeOnComponent() passing the arguments to this method, in order. The first time invokeOnComponent() returns true, abort traversing the rest of the Iterator and return true.

When calling ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) the implementation of this method must guarantee that the state of the component passed to the callback correctly reflects the component's position in the View hierarchy with respect to any state found in the argument clientId. For example, an iterating component such as UIData will need to set its row index to correctly reflect the argument clientId before finding the appropriate child component backed by the correct row. When
the callback returns, either normally or by throwing an Exception the implementation of this method must restore the state of the view to the way it was before invoking the callback.

If none of the elements from `getFacetsAndChildren()` returned `true` from `invokeOnComponent()`, return `false`.

Simple usage example to find a component by `clientId`.

```java
private UIComponent found = null;

private void doFind(FacesContext context, String clientId) {
    context.getViewRoot().invokeOnComponent(context, clientId,
        new ContextCallback() {
            public void invokeOnComponent(FacesContext context,
                UIComponent component) {
                found = component;
            }
        });
}
```

**Parameters:**
- `context` - the `FacesContext` for the current request
- `clientId` - the client identifier of the component to be passed to the argument callback.
- `callback` - an implementation of the Callback interface.

**Returns:**
- `true` if the a component with the given `clientId` is found, the callback method was successfully invoked passing that component as an argument, and no Exception was thrown.
- `false` if no component with the given `clientId` is found.

**Throws:**
- `NullPointerException` - if any of the arguments are null
- `FacesException` - if the argument Callback throws an Exception, it is wrapped in a `FacesException` and re-thrown.

**Since:**
- 1.2

```java
abstract public java.util.Map<K, V> getFacets()
```
### getFacets

```java
public abstract Map<String, UIComponent> getFacets()
```

Return a mutable `Map` representing the `UIComponent` facets associated with this `UIComponent`, keyed by facet name (which must be a `String`). The returned implementation must support all of the standard and optional `Map` methods, plus support the following additional requirements:

- The `Map` implementation must implement the `java.io.Serializable` interface.
- Any attempt to add a `null` key or value must throw a `NullPointerException`.
- Any attempt to add a key that is not a `String` must throw a `ClassCastException`.
- Any attempt to add a value that is not a `UIComponent` must throw a `ClassCastException`.
- Whenever a new facet `UIComponent` is added:
  - The `parent` property of the component must be set to this component instance.
  - If the `parent` property of the component was already non-
null, the component must first be removed from its previous parent (where it may have been either a child or a facet).

- Whenever an existing facet `UIComponent` is removed:
  - The `parent` property of the facet must be set to `null`.

```java
public int getFacetCount()

UIComponent facet UIComponent facet 0
facet

UIComponent #getFacets Map size()
UIComponentBase#getFacetCount
since 1.2
```

**getFacetCount**

```java
public int getFacetCount()

Return the number of facet `UIComponent`s that are associated with this `UIComponent`. If there are no facets, this method must return 0. The method must not cause the creation of a facet component map.

For backwards compatibility with classes that extend `UIComponent` directly, a default implementation is provided that simply calls `getFacets()` and then calls the `size()` method on the returned `Map`. A more optimized version of this method is provided in `UIComponentBase.getFacetCount()`.

Since:
1.2
```

```java
abstract public UIComponent getFacet(String name)

facet facet null facet facet Map
```
getFacet

```java
public abstract UIComponent getFacet(String name)
```

Convenience method to return the named facet, if it exists, or null otherwise. If the requested facet does not exist, the facets Map must not be created.

**Parameters:**

- `name` - Name of the desired facet

---

abstract public java.util.Iterator<E> getFacetsAndChildren()

```java
facet   Iterator facet   UIComponent   UIComponent
facet   facet   facet

Iterator   remove()
```

---

getFacetsAndChildren

```java
public abstract Iterator<UIComponent> getFacetsAndChildren()
```

Return an `Iterator` over the facet followed by child `UIComponent`s of this `UIComponent`. Facets are returned in an undefined order, followed by all the children in the order they are stored in the child list. If this component has no facets or children, an empty `Iterator` is returned.

The returned `Iterator` must not support the `remove()` operation.

---

abstract public void broadcast(FacesEvent event) throws
AbortProcessingException

FacesEvent

event

Throws AbortProcessingException: JavaServer Face

Throws IllegalArgument Exception: FacesEvent

Throws NullPointerException: event null

broadcast

public abstract void broadcast(FacesEvent event)

throws AbortProcessingException

Broadcast the specified FacesEvent to all registered event listeners who have expressed an interest in events of this type. Listeners are called in the order in which they were added.

Parameters:

   event - The FacesEvent to be broadcast

Throws:

AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

IllegalArgumentException - if the implementation class of this FacesEvent is not supported by this component

NullPointerException - if event is null

abstract public void decode(FacesContext context)

FacesContext  UIComponent

queueEvent()

context

Throws

NullPointerException: context null
**decode**

```java
public abstract void decode(FacesContext context)
```

Decode any new state of this `UIComponent` from the request contained in the specified `FacesContext`, and store this state as needed.

During decoding, events may be enqueued for later processing (by event listeners who have registered an interest), by calling `queueEvent()`.

**Parameters:**
- `context` - `FacesContext` for the request we are processing

**Throws:**
- `NullPointerException` - if `context` is null

---

**abstract public void encodeBegin(FacesContext context) throws java.io.IOException**

```java
rendered true UIComponent FacesContext
```

`Renderer UIComponent encodeBegin(FacesContext, UIComponent)`

- `context` - `FacesContext`
- `Throws` - `java.io.IOException: /`
- `Throws` - `NullPointerException: context null`

---

**encodeBegin**

```java
public abstract void encodeBegin(FacesContext context) throws IOException
```
If our rendered property is true, render the beginning of the current state of this UIComponent to the response contained in the specified FacesContext.

If a Renderer is associated with this UIComponent, the actual encoding will be delegated to Renderer.encodeBegin(FacesContext, UIComponent).

Parameters:
- context - FacesContext for the response we are creating

Throws:
- IOException - if an input/output error occurs while rendering
- NullPointerException - if context is null

abstract public void encodeChildren(FacesContext context) throws java.io.IOException

rendered true UIComponent UIComponent
rendersChildren true

Renderer UIComponent
encodeChildren(FacesContext, UIComponent)

context FacesContext
Throws java.io.IOException: /
Throws NullPointerException: context null

encodeChildren

public abstract void encodeChildren(FacesContext context) throws IOException

If our rendered property is true, render the child UIComponents of this UIComponent. This method will only be called if the rendersChildren property is true.
If a `Renderer` is associated with this `UIComponent`, the actual encoding will be delegated to `Renderer.encodeChildren(FacesContext, UIComponent)`.

**Parameters:**
- `context` - `FacesContext` for the response we are creating

**Throws:**
- `IOException` - if an input/output error occurs while rendering
- `NullPointerException` - if `context` is null

```java
abstract public void encodeEnd(FacesContext context)
throws java.io.IOException
```

rendered true `UIComponent`

```java
Renderer UIComponent encodeEnd(FacesContext, UIComponent)
```

**context** `FacesContext`

**Throws**
- `java.io.IOException`: /
- `NullPointerException`: `context` null

**encodeEnd**

```java
public abstract void encodeEnd(FacesContext context)
throws IOException
```

If our `rendered` property is true, render the ending of the current state of this `UIComponent`.

If a `Renderer` is associated with this `UIComponent`, the actual encoding will be delegated to `Renderer.encodeEnd(FacesContext, UIComponent)`.

**Parameters:**
- `context` - `FacesContext` for the response we are creating
public void encodeAll(FacesContext context) throws IOException

#isRendered true isRendered() true
#getRendersChildren

since 1.2
Throws java.io.IOException: /
Throws NullPointerException: context null

encodeAll

public void encodeAll(FacesContext context) throws IOException

If this component returns true from isRendered(), render this component and all its children that return true from isRendered(), regardless of the value of the getRendersChildren() flag.

Throws:

IOException - if an input/output error occurs while rendering
NullPointerException - if context is null

Since:
1.2

abstract protected void addFacesListener(FacesListener listener)

FacesListener UICOMPONENT UICOMPONENT API
public class FooEvent extends FacesEvent {
    ...
}

public interface FooListener extends FacesListener {
    public void processFoo(FooEvent event);
}

public class FooComponent extends UIComponentBase {
    ...
    public void addFooListener(FooListener listener) {
        addFacesListener(listener);
    }
    public void removeFooListener(FooListener listener) {
        removeFacesListener(listener);
    }
    ...
}

**addFacesListener**

protected abstract void addFacesListener(FacesListener listener)

Add the specified FacesListener to the set of listeners registered to receive event notifications from this UIComponent. It is expected that UIComponent classes acting as event sources will have corresponding typesafe APIs for registering listeners of the required type, and the implementation of those registration methods will delegate to this method. For example:

```
public class FooEvent extends FacesEvent {
    ...
}

public interface FooListener extends FacesListener {
    public void processFoo(FooEvent event);
}

public class FooComponent extends UIComponentBase {
    ...
    public void addFooListener(FooListener listener) {
```
addFacesListener(listener);
}
public void removeFooListener(FooListener listener) {
    removeFacesListener(listener);
}
...

Parameters:
   listener - The FacesListener to be registered

Throws:
   NullPointerException - if listener is null

abstract protected FacesListener[] getFacesListeners(Class<T> clazz)

   FacesListener 0                     clazz
   clazz          FacesListener

   Throws IllegalArgumentException: clazz FacesListener
   Throws NullPointerException: clazz null

getFacesListeners

protected abstract FacesListener[] getFacesListeners(Class clazz)

Return an array of registered FacesListeners that are instances of the specified class. If there are no such registered listeners, a zero-length array is returned. The returned array can be safely be cast to an array strongly typed to an element type of clazz.

Parameters:
   clazz - Class that must be implemented by a FacesListener for it to be returned

Throws:
   IllegalArgumentException - if class is not, and does not
abstract protected void removeFacesListener(FacesListener listener)

Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

Parameters:
listener - The FacesListener to be deregistered

Throws:
NullPointerException - if listener is null

abstract public void queueEvent(FacesEvent event)

IllegalStateException: UIViewRoot

Throws
NullPointerException: event null

queueEvent
public abstract void queueEvent(FacesEvent event)

Queue an event for broadcast at the end of the current request processing lifecycle phase. The default implementation in UIComponentBase must delegate this call to the queueEvent() method of the parent UIComponent.

Parameters:
   event - FacesEvent to be queued

Throws:
   IllegalStateException - if this component is not a descendant of a UIViewRoot
   NullPointerException - if event is null

abstract public void processRestoreState(FacesContext context, Object state)

facet

- getFacetsAndChildren()  
  UIComponent facet
- processRestoreState()
- restoreState()

Throws

context FacesContext

Throws

NullPointerException: context null

processRestoreState

public abstract void processRestoreState(FacesContext context, Object state)

Perform the component tree processing required by the Restore View phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself,
as follows.

- Call the `processRestoreState()` method of all facets and children of this `UIComponent` in the order determined by a call to `getFacetsAndChildren()`.
- Call the `restoreState()` method of this component.

This method may not be called if the state saving method is set to server.

**Parameters:**
- `context` - `FacesContext` for the request we are processing

**Throws:**
- `NullPointerException` - if `context` is null

---

```java
abstract public void processDecodes(FacesContext context)
```

**facet**

- `UIComponent` rendered `false`
- `getFacetsAndChildren()` `UIComponent` facet
- `processDecodes()`
- `decode()`
- `RuntimeException` `FacesContext#renderResponse`

```java
context FacesContext
Throws NullPointerException: context null
```

**processDecodes**

```java
public abstract void processDecodes(FacesContext context)
```

Perform the component tree processing required by the *Apply Request Values* phase of the request processing lifecycle for all
facets of this component, all children of this component, and this component itself, as follows.

- If the rendered property of this UIComponent is false, skip further processing.
- Call the processDecodes() method of all facets and children of this UIComponent, in the order determined by a call to getFacetsAndChildren().
- Call the decode() method of this component.
- If a RuntimeException is thrown during decode processing, call FacesContext.renderResponse() and re-throw the exception.

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null

abstract public void processValidators(FacesContext context)

facet

- UIComponent rendered false
- getFacetsAndChildren() UIComponent facet
- processValidators()

context FacesContext

Throws

NullPointerException: context null

processValidators

public abstract void processValidators(FacesContext context)

Perform the component tree processing required by the Process Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component
abstract public void processUpdates(FacesContext context)

Perform the component tree processing required by the Update Model Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- If the rendered property of this UIComponent is false, skip further processing.

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null
Call the `processUpdates()` method of all facets and children of this `UIComponent`, in the order determined by a call to `getFacetsAndChildren()`.

**Parameters:**
- `context` - `FacesContext` for the request we are processing

**Throws:**
- `NullPointerException` - if `context` is null

### `abstract public Object processSaveState(FacesContext context)`

#### facet

- `transient true` null
- `getFacetsAndChildren()` `UIComponent` facet
- `processSaveState()` `facet`
- `saveState()`
- `Serializable` `Object`

**context** `FacesContext`

**Throws** `NullPointerException`: `context` null

### `processSaveState`

`public abstract Object processSaveState(FacesContext context)`

Perform the component tree processing required by the state saving portion of the `Render Response` phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- consult the `transient` property of this component. If true, just
return null.

- Call the `processSaveState()` method of all facets and children of this `UIComponent` in the order determined by a call to `getFacetsAndChildren()`, skipping children and facets that are transient.
- Call the `saveState()` method of this component.
- Encapsulate the child state and your state into a Serializable Object and return it.

This method may not be called if the state saving method is set to server.

**Parameters:**
- `context` - `FacesContext` for the request we are processing

**Throws:**
- `NullPointerException` - if `context` is `null`

---

**abstract protected `FacesContext` getFacesContext()**

`FacesContext`

**getFacesContext**

**protected abstract `FacesContext` getFacesContext()**

Convenience method to return the `FacesContext` instance for the current request.

---

**abstract protected `Renderer` getRenderer(`FacesContext` context)**

`Renderer`  
`null`  

`context`  
`FacesContext`
getRenderer

protected abstract Renderer getRenderer(FacesContext context)

Convenience method to return the Renderer instance associated with this component, if any; otherwise, return null.

Parameters:
context - FacesContext for the current request

PS:
javax.faces.component  **Class UIComponentBase**

java.lang.Object
   ├ javax.faces.component.UIComponent
   │   └ javax.faces.component.UIComponentBase

All Implemented Interfaces:
   StateHolder

Direct Known Subclasses:
   UIColumn, UICommand, UIData, UIForm, UIGraphic, UIMessage, UIMessages, UINamingContainer, UIOutput, UIPanel, UIParameter, UISelectItem, UISelectItems, UIViewRoot

---

public abstract class **UIComponentBase**

extends **UIComponent**

*Extends:* **UIComponent**

*Extended by:* **UIComponent**, UIColumn, UICommand, UIData, UIForm, UIGraphic, UIMessage, UIMessages, UINamingContainer, UIOutput, UIPanel, UIParameter, UISelectItem, UISelectItems, UIViewRoot

**UIComponentBase** is a convenience base class that implements the default concrete behavior of all methods defined by **UIComponent**.

By default, this class defines `getRendersChildren()` to find the renderer for this component and call its `getRendersChildren()` method. The default implementation on the Renderer returns `false`. As of version 1.2 of the
JavaServer Faces Specification, component authors are encouraged to return `true` from this method and rely on the implementation of `encodeChildren(javax.faces.context.FacesContext)` in this class and in the Renderer
`Renderer.encodeChildren(javax.faces.context.FacesContext, javax.faces.component.UIComponent`). Subclasses that wish to manage the rendering of their children should override this method to return `true` instead.

---

## Field Summary

### Fields inherited from class javax.faces.component.UIComponent

- `bindings`

## Constructor Summary

- `UIComponentBase()`

## Method Summary

- `addFacesListener(FacesListener listener)`

  Add the specified `FacesListener` to the set of listeners that receive event notifications from this `UIComponent`.

- `broadcast(FacesEvent event)`

  Broadcast the specified `FacesEvent` to all registered listeners who have expressed an interest in events of this type.

- `decode(FacesContext context)`

  Decode any new state of this `UIComponent` from the specified `FacesContext`, and store this state as needed.

- `encodeBegin(FacesContext context)`

  If our rendered property is `true`, render the beginning state of this `UIComponent` to the response contained in `FacesContext`.

- `encodeChildren(FacesContext context)`
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void</code></td>
<td>If our rendered property is true, render the child <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>void encodeEnd(FacesContext context)</code></td>
<td>If our rendered property is true, render the end of the current state of this <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>UIComponent</code> findComponent(String expr)`</td>
<td>Search for and return the <code>UIComponent</code> with an specified search expression (if any), according to the below.</td>
</tr>
<tr>
<td><code>Map&lt;String, Object&gt;</code> getAttributes()`</td>
<td>Return a mutable <code>Map</code> representing the attributes (below) associated with this <code>UIComponent</code>, keyed by attribute (which must be a <code>String</code>).</td>
</tr>
<tr>
<td><code>int getChildCount()</code></td>
<td>Return the number of child <code>UIComponent</code>s that a <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>List&lt;UIComponent&gt;</code> getChildren()`</td>
<td>Return a mutable <code>List</code> representing the child <code>UIComponent</code>s with this component.</td>
</tr>
<tr>
<td><code>String getClientId(FacesContext context)</code></td>
<td>Return a client-side identifier for this component, generating necessary.</td>
</tr>
<tr>
<td><code>protected FacesContext getFacesContext()</code></td>
<td>Convenience method to return the <code>FacesContext</code> current request.</td>
</tr>
<tr>
<td><code>protected FacesListener[] getFacesListeners(Class clazz)</code></td>
<td>Return an array of registered <code>FacesListener</code>s that are specified class.</td>
</tr>
<tr>
<td><code>UIComponent getFacet(String name)</code></td>
<td>Convenience method to return the named facet, otherwise.</td>
</tr>
<tr>
<td><code>int getFacetCount()</code></td>
<td>Return the number of facet <code>UIComponent</code>s that a <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>Map&lt;String, UIComponent&gt;</code> getFacets()`</td>
<td>Return a mutable <code>Map</code> representing the facet <code>UIComponent</code>s.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getFacetsAndChildren()</code></td>
<td>Return an Iterator over the facet followed by child this UIComponent.</td>
</tr>
<tr>
<td><code>getId()</code></td>
<td>Return the component identifier of this UIComponent.</td>
</tr>
<tr>
<td><code>getParent()</code></td>
<td>Return the parent UIComponent of this UIComponent.</td>
</tr>
<tr>
<td><code>getRenderer(FacesContext context)</code></td>
<td>Convenience method to return the Renderer instance this component, if any; otherwise, return null.</td>
</tr>
<tr>
<td><code>getRendererType()</code></td>
<td>Return the Renderer type for this UIComponent (if any).</td>
</tr>
<tr>
<td><code>getRendersChildren()</code></td>
<td>Return a flag indicating whether this component is responsible rendering its child components.</td>
</tr>
<tr>
<td><code>getValueBinding(String name)</code></td>
<td>Deprecated. This has been replaced by <code>UIComponent.getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td><code>invokeOnComponent(FacesContext context, String clientId, ContextCallback callback)</code></td>
<td>Starting at this component in the View hierarchy, search with a clientId equal to the argument clientId and, if found, call the ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the passing the current FacesContext and the found component as arguments.</td>
</tr>
<tr>
<td><code>isRendered()</code></td>
<td>Return true if this component (and its children) during the Render Response phase of the request processing.</td>
</tr>
<tr>
<td><code>isTransient()</code></td>
<td>If true, the Object implementing this interface must not state saving or restoring.</td>
</tr>
<tr>
<td><code>processDecodes(FacesContext context)</code></td>
<td></td>
</tr>
</tbody>
</table>
```java
void performComponentTreeProcessingRequiredByTheValuesPhaseOfTheRequestProcessingLifecycleForAllFacetsOfThisComponent,AllChildrenOfThisComponent,AndThisComponentItselfAsFollows.

void processRestoreState(FacesContext context, Object)
Perform the component tree processing required by the Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself as follows.

Object processSaveState(FacesContext context)
Perform the component tree processing required by the state portion of the Render Response phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

void processUpdates(FacesContext context)
Perform the component tree processing required by the Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

void processValidators(FacesContext context)
Perform the component tree processing required by the Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

void queueEvent(FacesEvent event)
Queue an event for broadcast at the end of the request processing lifecycle phase.

protected void removeFacesListener(FacesListener listener)
Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

static Object restoreAttachedState(FacesContext context, Object)
This method is called by UIComponent subclasses to restore the objects they saved using saveAttachedState(javax.faces.context.FacesContext, java.lang.Object).

void restoreState(FacesContext context, Object state)
Perform any processing required to restore the state in the state Object.
```
static Object saveAttachedState(FacesContext context, Object attachedObject)

This method is called by UIComponent subclasses for more attached objects.

Object saveState(FacesContext context)

Gets the state of the instance as a Serializable.

void setId(String id)

Set the component identifier of this UIComponent.

void setParent(UIComponent parent)

Set the parent UIComponent of this UIComponent.

void setRendered(boolean rendered)

Set the rendered property of this UIComponent.

void setRendererType(String rendererType)

Set the Renderer type for this UIComponent, or null to render themselves.

void setTransient(boolean transientFlag)

Denotes whether or not the Object implementing this interface must not participate in state saving or restoring.

void setValueBinding(String name, ValueBinding binding)

Deprecated. This has been replaced by UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression).

Methods inherited from class javax.faces.component.UIComponent

encodeAll, getContainerClientId, getFamily, getValueExpression, setValueExpression

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public UIComponentBase()
UIComponentBase

public UIComponentBase()

Method Detail

public java.util.Map<K, V> getAttributes()

getAttributes

public Map<String, Object> getAttributes()

Description copied from class: UIComponent

Return a mutable Map representing the attributes (and properties, see below) associated with this UIComponent, keyed by attribute name (which must be a String). The returned implementation must support all of the standard and optional Map methods, plus support the following additional requirements:

- The Map implementation must implement the java.io.Serializable interface.
- Any attempt to add a null key or value must throw a NullPointerException.
- Any attempt to add a key that is not a String must throw a ClassCastException.
- If the attribute name specified as a key matches a property of this UIComponent's implementation class, the following methods will have special behavior:
  - containsKey - Return false.
  - get() - If the property is readable, call the getter method and return the returned value (wrapping primitive values in their corresponding wrapper classes); otherwise throw IllegalArgumentException.
  - put() - If the property is writeable, call the setter method to set the corresponding value (unwrapping primitive values in their corresponding wrapper classes). If the property is not
writeable, or an attempt is made to set a property of primitive type to null, throw IllegalArgumentException.

- remove - Throw IllegalArgumentException.

Specified by:
getAttributes in class UICOMPONENT

```java
public ValueBinding getValueBinding(String name)
```

**NullPointerException**

Throws: NullPointeException: NullPointerException

```
#getValueExpression
```

**getvalueExpression**

**getvalueExpression**

```
public ValueBinding getValueBinding(String name)
```

**Deprecated. This has been replaced by**
UICOMPONENT.getValueExpression(java.lang.String).

Call through to UICOMPONENT.getValueExpression(java.lang.String) and examine the result. If the result is an instance of the wrapper class mandated in UICOMPONENT.setValueBinding(java.lang.String, javax.faces.el.ValueBinding), extract the ValueBinding instance and return it. Otherwise, wrap the result in an implementation of ValueBinding, and return it.

Specified by:
getValueBinding in class UICOMPONENT

Parameters:
name - Name of the attribute or property for which to retrieve a ValueBinding

Throws:
NullPointerException - if name is null

```
public void setValueBinding(String name, ValueBinding binding)
```

NullPointerException

```
ValueExpression binding #setValueExpression
```

Throws

- IllegalArgumentException: name id parent
- NullPointerException: name null

deprecated

```
setValueBinding
```

```
setValueBinding(String name, ValueBinding binding)
```

**Deprecated. This has been replaced by**

```
UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression)
```

Wrap the argument binding in an implementation of ValueExpression and call through to

```
UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression)
```

**Specified by:**

setValueBinding in class UIComponent

**Parameters:**

- name - Name of the attribute or property for which to set a ValueBinding
- binding - The ValueBinding to set, or null to remove any currently set ValueBinding

**Throws:**

- IllegalArgumentException - if name is one of id or parent
- NullPointerException - if name is null
public String getClientId(FacesContext context)

Throws  NullPointerException: NullPointerException

clientId

g getClientId

g getClientId(FacesContext context)

Description copied from class: UIComponent

Return a client-side identifier for this component, generating one if necessary. The associated Renderer, if any, will be asked to convert the clientId to a form suitable for transmission to the client.

The return from this method must be the same value throughout the lifetime of the instance, unless the id property of the component is changed, or the component is placed in a NamingContainer whose client ID changes (for example, UIData). However, even in these cases, consecutive calls to this method must always return the same value. The implementation must follow these steps in determining the clientId:

Find the closest ancestor to this component in the view hierarchy that implements NamingContainer. Call getContainerClientId() on it and save the result as the parentId local variable. Call

UIComponent.getId() on this component and save the result as the

myId local variable. If myId is null, call

context.getViewRoot().createUniqueId() and assign the result to

myId. If parentId is non-null, let myId equal parentId +

NamingContainer.SEPARATOR_CHAR + myId. Call

Renderer.convertClientId(javax.faces.context.FacesContext, java.lang.String), passing myId, and return the result.

Specified by:

g getClientId in class UIComponent

Parameters:

context - The FacesContext for the current request

Throws:
NullPointerException - if context is null

public String getId()

getId

public String getId()

Description copied from class: UICOMPONENT

Return the component identifier of this UICOMPONENT.

Specified by:

getID in class UICOMPONENT

public void setId(String id)

Throws IllegalArgumentException: NullPointerException id

Throws IllegalStateException: NullPointerException

setId

public void setId(String id)

Description copied from class: UICOMPONENT

Set the component identifier of this UICOMPONENT (if any). Component identifiers must obey the following syntax restrictions:

- Must not be a zero-length String.
- First character must be a letter or an underscore ('_').
- Subsequent characters must be a letter, a digit, an underscore ('_'), or a dash ('-').

Component identifiers must also obey the following semantic
restrictions (note that this restriction is **NOT** enforced by the `setId()` implementation):

- The specified identifier must be unique among all the components (including facets) that are descendents of the nearest ancestor `UIComponent` that is a `NamingContainer`, or within the scope of the entire component tree if there is no such ancestor that is a `NamingContainer`.

**Specified by:**
`setId` in class `UIComponent`

**Parameters:**
- `id` - The new component identifier, or `null` to indicate that this `UIComponent` does not have a component identifier

**Throws:**
- `IllegalArgumentException` - if `id` is not syntactically valid
- `IllegalStateException`

---

```java
public UIComponent getParent()
```

**getParent**

```java
public UIComponent getParent()
```

**Description copied from class:** `UIComponent`

Return the parent `UIComponent` of this `UIComponent`, if any. A component must allow child components to be added to and removed from the list of children of this component, even though the child component returns `null` from `getParent()`.

**Specified by:**
`getParent` in class `UIComponent`

---

```java
public void setParent(UIComponent parent)
```
setParent

public void.setParent(UIComponent parent)

Description copied from class: UIComponent

Set the parent UIComponent of this UIComponent. This method must never be called by developers; a UIComponent's internal implementation will call it as components are added to or removed from a parent's child List or facet Map.

Specified by:
  setParent in class UIComponent
Parameters:
  parent - The new parent, or null for the root node of a component tree

public boolean isRendered()
public void setRendered(boolean rendered)

**Description copied from class:** [UIComponent]

Set the rendered property of this [UIComponent].

**Specified by:**

`setRendered` in class [UIComponent]

**Parameters:**

rendered - If true render this component; otherwise, do not render this component

---

public String getRendererType()

**getRendererType**

public String getRendererType()

**Description copied from class:** [UIComponent]

Return the Renderer type for this [UIComponent] (if any).

**Specified by:**

`getRendererType` in class [UIComponent]

---

public void setRendererType(String rendererType)

**setRendererType**

public void setRendererType(String rendererType)

**Description copied from class:** [UIComponent]

Set the Renderer type for this [UIComponent], or null for components that render themselves.
public boolean getRendersChildren()

getRendersChildren

public boolean getRendersChildren()

Description copied from class: UIComponent

Return a flag indicating whether this component is responsible for rendering its child components. The default implementation in getRendersChildren() tries to find the renderer for this component. If it does, it calls Renderer.getRendersChildren() and returns the result. If it doesn't, it returns false. As of version 1.2 of the JavaServer Faces Specification, component authors are encouraged to return true from this method and rely on encodeChildren(javax.faces.context.FacesContext).

Specified by: getRendersChildren in class UIComponent

public java.util.List<E> getChildren()

getChildren

public List<UIComponent> getChildren()

Description copied from class: UIComponent

Return a mutable List representing the child UIComponents
associated with this component. The returned implementation must support all of the standard and optional List methods, plus support the following additional requirements:

- The List implementation must implement the java.io.Serializable interface.
- Any attempt to add a null must throw a NullPointerException
- Any attempt to add an object that does not implement UIComponent must throw a ClassCastException.
- Whenever a new child component is added, the parent property of the child must be set to this component instance. If the parent property of the child was already non-null, the child must first be removed from its previous parent (where it may have been either a child or a facet).
- Whenever an existing child component is removed, the parent property of the child must be set to null.

Specified by:

getChildren in class UICOMPONENT

---

public int getChildCount()

getChildCount

public int getChildCount()

Description copied from class: UICOMPONENT

Return the number of child UICOMPONENTs that are associated with this UICOMPONENT. If there are no children, this method must return 0. The method must not cause the creation of a child component list.

Specified by:

getChildCount in class UICOMPONENT

---

public UICOMPONENT findComponent(String expr)
findComponent

public UIComponent findComponent(String expr)

Description copied from class: UIComponent

Search for and return the UIComponent with an id that matches the specified search expression (if any), according to the algorithm described below.

For a method to find a component given a simple clientId, see UIComponent.invokeOnComponent(javax.faces.context.FacesContext, java.lang.String, javax.faces.component.ContextCallback).

Component identifiers are required to be unique within the scope of the closest ancestor NamingContainer that encloses this component (which might be this component itself). If there are no NamingContainer components in the ancestry of this component, the root component in the tree is treated as if it were a NamingContainer, whether or not its class actually implements the NamingContainer interface.

A search expression consists of either an identifier (which is matched exactly against the id property of a UIComponent, or a series of such identifiers linked by the NamingContainer.SEPARATOR_CHAR character value. The search algorithm should operate as follows, though alternate algorithms may be used as long as the end result is the same:

- Identify the UIComponent that will be the base for searching, by stopping as soon as one of the following conditions is met:
  - If the search expression begins with the separator character (called an "absolute" search expression), the base will be the root UIComponent of the component tree. The leading separator character will be stripped off, and the remainder of the search expression will be treated as a "relative" search expression as described below.
- Otherwise, if this UIComponent is a NamingContainer it will serve as the basis.
- Otherwise, search up the parents of this component. If a NamingContainer is encountered, it will be the base.
- Otherwise (if no NamingContainer is encountered) the root UIComponent will be the base.

- The search expression (possibly modified in the previous step) is now a "relative" search expression that will be used to locate the component (if any) that has an id that matches, within the scope of the base component. The match is performed as follows:
  - If the search expression is a simple identifier, this value is compared to the id property, and then recursively through the facets and children of the base UIComponent (except that if a descendant NamingContainer is found, its own facets and children are not searched).
  - If the search expression includes more than one identifier separated by the separator character, the first identifier is used to locate a NamingContainer by the rules in the previous bullet point. Then, the findComponent() method of this NamingContainer will be called, passing the remainder of the search expression.

Specified by:
findComponent in class UIComponent

Parameters:
expr - Search expression identifying the UIComponent to be returned

Returns:
the found UIComponent, or null if the component was not found.

Throws:
NullPointerException - if expr is null

public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback) throws FacesException
NullPointerException
private UIComponent found = null;

private void doFind(FacesContext context, String clientId) {
    context.getViewRoot().invokeOnComponent(context, clientId,
        new ContextCallback() {
            public void invokeOnComponent(FacesContext context,
                UIComponent component) {
                found = component;
            }
        });
}

since 1.2
Throws NullPointerException: NullPointerException null
Throws FacesException: NullPointerException Callback Exception FacesException
public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback) throws FacesException

Starting at this component in the View hierarchy, search for a component with a clientId equal to the argument clientId and, if found, call the ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the argument callback, passing the current FacesContext and the found component as arguments. This method is similar to UICOMPONENT.findComponent(java.lang.String) but it does not support the leading NamingContainer.SEPARATOR_CHAR syntax for searching from the root of the View.

The default implementation will first check if this.getClientId() is equal to the argument clientId. If so, call the ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the argument callback, passing through the FacesContext argument and passing this as the component argument. If an Exception is thrown by the callback, wrap it in a FacesException argument and re-throw it. Otherwise, return true.

Otherwise, for each component returned by UICOMPONENT.getFacetsAndChildren(), call invokeOnComponent() passing the arguments to this method, in order. The first time invokeOnComponent() returns true, abort traversing the rest of the iterator and return true.

When calling ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) the implementation of this method must guarantee that the state of the component passed to the callback correctly reflects the component's position in the View.
hierarchy with respect to any state found in the argument clientId. For example, an iterating component such as UIData will need to set its row index to correctly reflect the argument clientId before finding the appropriate child component backed by the correct row. When the callback returns, either normally or by throwing an Exception the implementation of this method must restore the state of the view to the way it was before invoking the callback.

If none of the elements from UIComponent.getFacetsAndChildren() returned true from invokeOnComponent(), return false.

Simple usage example to find a component by clientId.

```java
private UIComponent found = null;

private void doFind(FacesContext context, String clientId) {
    context.getViewRoot().invokeOnComponent(context, clientId,
        new ContextCallback() {
            public void invokeOnComponent(FacesContext context,
                UIComponent component) {
                found = component;
            }
        });
}
```

Overrides: invokeOnComponent in class UIComponent

Parameters:
context - the FacesContext for the current request
clientId - the client identifier of the component to be passed to the argument callback.
callback - an implementation of the Callback interface.

Returns:
true if the a component with the given clientId is found, the callback method was successfully invoked passing that component as an argument, and no Exception was thrown. Returns false if no component with the given clientId is found.

Throws:
NullPointerException - if any of the arguments are null
FacesException - if the argument Callback throws an Exception,
it is wrapped in a FacesException and re-thrown.

Since:
1.2

public java.util.Map<K, V> getFacets()

getFacets

public Map<String, UIComponent> getFacets()

Description copied from class: UIComponent

Return a mutable Map representing the facet UIComponent s associated with this UIComponent, keyed by facet name (which must be a String). The returned implementation must support all of the standard and optional Map methods, plus support the following additional requirements:

- The Map implementation must implement the java.io.Serializable interface.
- Any attempt to add a null key or value must throw a NullPointerException.
- Any attempt to add a key that is not a String must throw a ClassCastException.
- Any attempt to add a value that is not a UIComponent must throw a ClassCastException.
- Whenever a new facet UIComponent is added:
  - The parent property of the component must be set to this component instance.
  - If the parent property of the component was already non-null, the component must first be removed from its previous parent (where it may have been either a child or a facet).
- Whenever an existing facet UIComponent is removed:
  - The parent property of the facet must be set to null.

Specified by:
getFacets in class UIComponent
public int getFacetCount()

g FACET COUNT

getFacetCount

public int getFacetCount()

Description copied from class: UIC com ponent

Return the number of facet UIC omponents that are associated with this UIC omponent. If there are no facets, this method must return 0. The method must not cause the creation of a facet component map.

For backwards compatibility with classes that extend UIC omponent directly, a default implementation is provided that simply calls UIC omponent.getFacets() and then calls the size() method on the returned Map. A more optimized version of this method is provided in getFacetCount().

Overrides:

g FACET COUNT in class UIC omponent

public UIC omponent getFacet(String name)

g FACET

getFacet

public UIC omponent getFacet(String name)

Description copied from class: UIC omponent

Convenience method to return the named facet, if it exists, or null otherwise. If the requested facet does not exist, the facets Map must not be created.

Specified by:

g FACET in class UIC omponent
Parameters:
   name - Name of the desired facet

public java.util.Iterator<E> getFacetsAndChildren()

getFacetsAndChildren

public Iterator<UIComponent> getFacetsAndChildren()

   Description copied from class: UIComponent

   Return an Iterator over the facet followed by child UIComponents of this UIComponent. Facets are returned in an undefined order, followed by all the children in the order they are stored in the child list. If this component has no facets or children, an empty Iterator is returned.

   The returned Iterator must not support the remove() operation.

   Specified by:
      getFacetsAndChildren in class UIComponent

public void broadcast(FacesEvent event) throws AbortProcessingException

   Throws AbortProcessingException: NullPointerException JavaServer Face
   Throws IllegalStateException: NullPointerException
   Throws NullPointerException: NullPointerException event null

broadcast

public void broadcast(FacesEvent event)
   throws AbortProcessingException

   Description copied from class: UIComponent
Broadcast the specified FacesEvent to all registered event listeners who have expressed an interest in events of this type. Listeners are called in the order in which they were added.

**Specified by:**
*broadcast* in class UIComponent

**Parameters:**
event - The FacesEvent to be broadcast

**Throws:**
AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed
IllegalStateException
NullPointerException - if event is null

---

public void decode(FacesContext context)  
Throws NullPointerException: NullPointerException  
context  
null

decode

public void decode(FacesContext context)

**Description copied from class:** UIComponent

Decode any new state of this UIComponent from the request contained in the specified FacesContext, and store this state as needed.

During decoding, events may be enqueued for later processing (by event listeners who have registered an interest), by calling queueEvent().

**Specified by:**
decode in class UIComponent

**Parameters:**
context - FacesContext for the request we are processing

**Throws:**
public void encodeBegin(FacesContext context) throws IOException

Throws: NullPointerException: NullPointerException context null

Description copied from class: UIComponent

If our rendered property is true, render the beginning of the current state of this UIComponent to the response contained in the specified FacesContext.

If a Renderer is associated with this UIComponent, the actual encoding will be delegated to Renderer.encodeBegin(FacesContext, UIComponent).

Specified by: encodeBegin in class UIComponent

Parameters:
context - FacesContext for the response we are creating

Throws:
NullPointerException - if context is null
IOException - if an input/output error occurs while rendering

public void encodeChildren(FacesContext context) throws IOException

Throws: NullPointerException: NullPointerException context null

encodeChildren
public void encodeChildren(FacesContext context) throws IOException

Description copied from class: UICOMPONENT

If our rendered property is true, render the child UICOMPONENTs of this UICOMPONENT. This method will only be called if the rendersChildren property is true.

If a Renderer is associated with this UICOMPONENT, the actual encoding will be delegated to Renderer.encodeChildren(FacesContext, UICOMPONENT).

Specified by:
encodeChildren in class UICOMPONENT
Parameters:
context - FacesContext for the response we are creating
Throws:
NullPointerException - if context is null
IOException - if an input/output error occurs while rendering

public void encodeEnd(FacesContext context) throws java.io.IOException

Throws java.io.IOException: NullPointerException /
Throws NullPointerException: NullPointerException context null

codeEnd

public void encodeEnd(FacesContext context) throws IOException

Description copied from class: UICOMPONENT

If our rendered property is true, render the ending of the current state of this UICOMPONENT.

If a Renderer is associated with this UICOMPONENT, the actual encoding
will be delegated to `Renderer.encodeEnd(FacesContext, UIComponent)`.

**Specified by:**
`encodeEnd` in class `UIComponent`  

**Parameters:**
> context - `FacesContext` for the response we are creating 

**Throws:**
> `IOException` - if an input/output error occurs while rendering  
> `NullPointerException` - if context is null 

---

protected void addFacesListener(`FacesListener` listener)

`FacesListener` `UIComponent` `UIComponent` API

```java
public class FooEvent extends FacesEvent {
    ...
    protected boolean isAppropriateListener(FacesListener listener) {
        return (listener instanceof FooListener);
    }
    protected void processListener(FacesListener listener) {
        ((FooListener) listener).processFoo(this);
    }
    ...
}

public interface FooListener extends FacesListener {
    public void processFoo(FooEvent event);
}

public class FooComponent extends UIComponentBase {
    ...
    public void addFooListener(FooListener listener) {
        addFacesListener(listener);
    }
    public void removeFooListener(FooListener listener) {
        removeFacesListener(listener);
    }
```
addFacesListener

protected void addFacesListener(FacesListener listener)

Add the specified FacesListener to the set of listeners registered to receive event notifications from this UIComponent. It is expected that UIComponent classes acting as event sources will have corresponding typesafe APIs for registering listeners of the required type, and the implementation of those registration methods will delegate to this method. For example:

public class FooEvent extends FacesEvent {
    ...
    protected boolean isAppropriateListener(FacesListener listener) {
        return (listener instanceof FooListener);
    }
    protected void processListener(FacesListener listener) {
        ((FooListener) listener).processFoo(this);
    }
    ...
}

public interface FooListener extends FacesListener {
    public void processFoo(FooEvent event);
}

public class FooComponent extends UIComponentBase {
    ...
    public void addFooListener(FooListener listener) {
        addFacesListener(listener);
    }
    public void removeFooListener(FooListener listener) {
        removeFacesListener(listener);
    }
    ...
}
Specified by:
   addFacesListener in class UICOMPONENT
Parameters:
   listener - The FacesListener to be registered
Throws:
   NullPointerException - if listener is null

protected FacesListener[] getFacesListeners(Class<T> clazz)

Throws
   IllegalArgumentException: NullPointerException
   FacesListener
   FacesListener
Throws
   NullPointerException: NullPointerException
   clazz
   null

getFacesListeners

protected FacesListener[] getFacesListeners(Class clazz)

Description copied from class: UICOMPONENT

Return an array of registered FacesListener s that are instances of the specified class. If there are no such registered listeners, a zero-length array is returned. The returned array can be safely be cast to an array strongly typed to an element type of clazz.

Specified by:
   getFacesListeners in class UICOMPONENT
Parameters:
   clazz - Class that must be implemented by a FacesListener for it to be returned
Throws:
   IllegalArgumentException - if clazz is not, and does not implement, FacesListener
   NullPointerException - if clazz is null

protected void removeFacesListener(FacesListener
removeFacesListener

protected void removeFacesListener(FacesListener listener)

Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

Specified by:
removeFacesListener in class UIComponent

Parameters:
listener - The FacesListener to be deregistered

Throws:
NullPointerException - if listener is null

queueEvent

public void queueEvent(FacesEvent event)

Description copied from class: UIComponent

Queue an event for broadcast at the end of the current request processing lifecycle phase. The default implementation in UIComponentBase must delegate this call to the queueEvent() method of the parent UIComponent.
public void processDecodes(FacesContext context)

Arguments copied from class: UIComponent

Perform the component tree processing required by the Apply Request Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- If the rendered property of this UIComponent is false, skip further processing.
- Call the processDecodes() method of all facets and children of this UIComponent, in the order determined by a call to getFacetsAndChildren().
- Call the decode() method of this component.
- If a RuntimeException is thrown during decode processing, call FacesContext.renderResponse() and re-throw the exception.
NullPointerException - if context is null

public void processValidators(FacesContext context)

Throws  NullPointerException: NullPointerException  context  null

processValidators

public void processValidators(FacesContext context)

Description copied from class: UIComponent

Perform the component tree processing required by the Process Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- If the rendered property of this UIComponent is false, skip further processing.
- Call the processValidators() method of all facets and children of this UIComponent, in the order determined by a call to getFacetsAndChildren().

Specified by:
processValidators in class UIComponent

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null

public void processUpdates(FacesContext context)

Throws  NullPointerException: NullPointerException  context  null

processUpdates
public void processUpdates(FacesContext context)

Description copied from class: UIComponent

Perform the component tree processing required by the Update Model Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- If the rendered property of this UIComponent is false, skip further processing.
- Call the processUpdates() method of all facets and children of this UIComponent, in the order determined by a call to getFacetsAndChildren().

Specified by:
processUpdates in class UIComponent

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null

public Object processSaveState(FacesContext context)

Throws NullPointerException: NullPointerException context null

processSaveState

public Object processSaveState(FacesContext context)

Description copied from class: UIComponent

Perform the component tree processing required by the state saving portion of the Render Response phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- consult the transient property of this component. If true, just
return null.

- Call the `processSaveState()` method of all facets and children of this `UIComponent` in the order determined by a call to `getFacetsAndChildren()`, skipping children and facets that are transient.
- Call the `saveState()` method of this component.
- Encapsulate the child state and your state into a `Serializable` Object and return it.

This method may not be called if the state saving method is set to server.

**Specified by:**
`processSaveState` in class `UIComponent`

**Parameters:**
- `context` - `FacesContext` for the request we are processing

**Throws:**
- `NullPointerException` - if `context` is `null`

```java
public void processRestoreState(FacesContext context, Object state)
    throws NullPointerException
```

**processRestoreState**

```java
public void processRestoreState(FacesContext context, Object state)
```

**Description copied from class:** `UIComponent`

Perform the component tree processing required by the Restore View phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

- Call the `processRestoreState()` method of all facets and children of this `UIComponent` in the order determined by a call to
getFacetsAndChildren().

- Call the restoreState() method of this component.

This method may not be called if the state saving method is set to server.

Specified by:
  - processRestoreState in class UIComponent

Parameters:
  context - FacesContext for the request we are processing

Throws:
  - NullPointerException - if context is null

protected FacesContext getFacesContext()

getFacesContext

protected FacesContext getFacesContext()

Description copied from class: UIComponent

Convenience method to return the FacesContext instance for the current request.

Specified by:
  - getFacesContext in class UIComponent

protected Renderer getRenderer(FacesContext context)

getRenderer

protected Renderer getRenderer(FacesContext context)

Description copied from class: UIComponent
Convenience method to return the Renderer instance associated with this component, if any; otherwise, return null.

**Specified by:**
```
getRenderer in class UICOMPONENT
```

**Parameters:**
```
context - FacesContext for the current request
```

```java
public Object saveState(FacesContext context)
```

**saveState**

```java
public Object saveState(FacesContext context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UICOMPONENT with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`

```java
public void restoreState(FacesContext context, Object state)
```

**restoreState**
public void **restoreState**(FacesContext context, Object state)

Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the **StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)** method on all those instances as well.

---

**public boolean isTransient()**

**isTransient**

public boolean isTransient()

Description copied from interface: **StateHolder**

If true, the Object implementing this interface must not participate in state saving or restoring.

---

**public void setTransient(boolean transientFlag)**

**setTransient**

public void setTransient(boolean transientFlag)
Description copied from interface: **StateHolder**

Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**Parameters:**
- `transientFlag` - boolean pass `true` if this Object will participate in state saving or restoring, otherwise pass `false`.

```
public static Object saveAttachedState(FacesContext context, Object attachedObject)
```

**UICOMPONENT**

`StateHolder`

`#restoreAttachedState`

**Object**

`null`

`Lists`

`Lists`

`StateHolder`

**context**

`FacesContext`

`attachedObject`

`List`

`Object`

`attachedObject`

`StateHolder`

**Throws**

`NullPointerException: null`

**saveAttachedState**

```
public static Object saveAttachedState(FacesContext context, Object attachedObject)
```

This method is called by **UICOMPONENT** subclasses that want to save one or more attached objects. It is a convenience method that does the work of saving attached objects that may or may not implement the **StateHolder** interface. Using this method implies the use of `restoreAttachedState(javax.faces.context.FacesContext, java.lang.Object)` to restore the attached objects.

This method supports saving attached objects of the following type: `Objects`, `null` values, and `Lists` of these objects. If any contained
objects are not Lists and do not implement StateHolder, they must have zero-argument public constructors. The exact structure of the returned object is undefined and opaque, but will be serializable.

**Parameters:**
- context - the FacesContext for this request.
- attachedObject - the object, which may be a List instance, or an Object. The attachedObject (or the elements that comprise attachedObject) may implement StateHolder.

**Throws:**
- NullPointerException - if the context argument is null.

```java
public static Object restoreAttachedState(FacesContext context, Object stateObj) throws IllegalStateException

UIComponent #saveAttachedState
#saveAttachedState

#saveAttachedState

context FacesContext
stateObj #saveAttachedState
Throws NullPointerException: context null
Throws IllegalArgumentException: context null

restoreAttachedState

public static Object restoreAttachedState(FacesContext context, Object stateObj) throws IllegalStateException

This method is called by UIComponent subclasses that need to restore the objects they saved using saveAttachedState(javax.faces.context.FacesContext, java.lang.Object). This method is tightly coupled with saveAttachedState(javax.faces.context.FacesContext, java.lang.Object).```
This method supports restoring all attached objects types supported by `saveAttachedState(javax.faces.context.FacesContext, java.lang.Object)`.

**Parameters:**
- `context` - the `FacesContext` for this request
- `stateObj` - the opaque object returned from `saveAttachedState(javax.faces.context.FacesContext, java.lang.Object)`

**Throws:**
- `NullPointerException` - if context is null.
- `IllegalStateException` - if the object is not previously returned by `saveAttachedState(javax.faces.context.FacesContext, java.lang.Object)`.
javax.faces.webapp Class UIComponentBodyTag

java.lang.Object  
  ▼ javax.faces.webapp.UIComponentTagBase  
    ▼ javax.faces.webapp.UIComponentClassicTagBase  
      ▼ javax.faces.webapp.UIComponentTag  
        ▼ javax.faces.webapp.UIComponentBodyTag

All Implemented Interfaces:
  BodyTag, IterationTag, JspIdConsumer, JspTag, Tag

---

Deprecated. All component tags now implement BodyTag. This class has been replaced by UIComponentELTag.

Extends: UIComponentTagBase > UIComponentClassicTagBase > UIComponentTag

UIComponentBodyTag  UIComponent  JSP

deprecated

BodyTag  UIComponentELTag

public abstract class UIComponentBodyTag
extends UIComponentTag

UIComponentBodyTag is a base class for all JSP custom actions, related to a UIComponent, that need to process their tag bodies.

---

Field Summary

Fields inherited from class javax.faces.webapp.UIComponentClassicTagBase
bodyContent, pageContext, UNIQUE_ID_PREFIX
<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UIComponentBodyTag()</code></td>
</tr>
<tr>
<td>Deprecated.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class javax.faces.webapp.UIComponentTag</td>
</tr>
<tr>
<td><code>addComponent</code>, <code>getParentUIComponentTag</code>, <code>hasBinding</code>, <code>isSuppressed</code>, <code>isValueReference</code>, <code>release</code>, <code>setBinding</code>, <code>setProperties</code>, <code>setRendered</code></td>
</tr>
</tbody>
</table>

| Methods inherited from class javax.faces.webapp.UIComponentClassicTagBase|
| `addChild`, `addFacet`, `addVerbatimAfterComponent`, `addVerbatimBeforeComponent`, `createVerbatimComponent`, `createVerbatimComponentFromBodyContent`, `doAfterBody`, `doEndTag`, `doInitBody`, `doStartTag`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getBodyContent`, `getComponentInstance`, `getCreated`, `getCreatedComponents`, `getDoAfterBodyValue`, `getDoEndValue`, `getDoStartValue`, `getFacesContext`, `getFacesJspId`, `getFacetName`, `getId`, `getIdxOfNextChildTag`, `getJspId`, `getParent`, `getParentUIComponentClassicTagBase`, `getPreviousOut`, `getPreviousOut` |
Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from class javax.servlet.jsp.tagext.Tag
doEndTag, doStartTag, getParent, setPageContext, setParent

Constructor Detail

public UICOMPONENTBODYTAG()

UICOMPONENTBODYTAG

public UICOMPONENTBODYTAG()

   Deprecated.
PS:
javax.faces.webapp  **Class UIComponentClassicTagBase**

`java.lang.Object`  
  - `javax.faces.webapp.UIComponentTagBase`  
  - `javax.faces.webapp.UIComponentClassicTagBase`

**All Implemented Interfaces:**  
  - `BodyTag`, `IterationTag`, `JspIdConsumer`, `JspTag`, `Tag`

**Direct Known Subclasses:**  
  - `UIComponentELTag`, `UIComponentTag`

---

```java
public abstract class UIComponentClassicTagBase
    extends UIComponentTagBase
    implements JspIdConsumer, BodyTag
```

**Extends:** `UIComponentTagBase`  
**Implements:** `javax.servlet.jsp.tagext.JspIdConsumer`, `BodyTag`  
**Extended by:** `UIComponentELTag`, `UIComponentTag`

---

`UIComponentTagBase`  
**JSP**  
  "classic" JSP

`javax.faces.component.UIComponent`  
**Faces 1.2**  
**BodyTag**

---

`UIComponentTag`  
**UIComponentClassicTagBase**  
**Faces 1.1**  
**EL**

---

`UIComponentELTag`  
**UIComponentClassicTagBase**  
**EL API**

---

**Faces JSP  Faces**

**CASE**

- **CASE 1 /**
- **CASE 2 /**
- **CASE 3  <f:verbatim> /**
• CASE 4 /

<h:panelGrid style="color:red" border="4" columns="2">
CASE 1
<h:outputText value="component 1"/>
CASE 2
<h:outputText value="component 2"/>
<f:verbatim>CASE 3</f:verbatim>
<c:out value="${pageScope.CASE4}" />
</h:panelGrid>

faces  ${pageScope.CASE4}  "  CASE 4"
<table border="4" style="color:red">
<tbody>
<tr><td>CASE 1</td></tr>
<tr><td>component 1</td></tr>
<tr><td>CASE 2</td>
<tr><td>component 2</td></tr>
<tr><td>CASE 3</td>
<td>CASE 4</td></tr>
</tbody>
</table>

UIComponentTagBase is the base class for all JSP tags that use the "classic" JSP tag interface that correspond to a UIComponent instance in the view. In Faces 1.2, all component tags are BodyTag instances to allow for the execution of the page to build the component tree, but not render it. Rendering happens only after the component tree is completely built.

UIComponentTag extends UIComponentClassicTagBase to add support for properties that conform to the Faces 1.1 EL.

UIComponentELTag extends UIComponentClassicTagBase class to add support for properties that conform to the EL API.
The default implementation allows the proper interweaving of template text, non-Faces JSP tag output, and Faces component tag output in the same page, as expected by the page author.

The CASE markers in the following example will be cited in the method descriptions of this class.

- **CASE 1** describes template text and/or non-component custom tag output occurring as the child of a component tag, but before the first component tag child of that component tag.

- **CASE 2** describes template text and/or non-component custom tag output occurring between two sibling component tags.

- **CASE 3** describes template text and/or non-component custom tag output occurring as the child content of an `<f:verbatim>` tag at any point in the page.

- **CASE 4** describes template text and/or non-component custom tag output occurring between the last child component tag and its enclosing parent component tag's end tag.

```xml
<h:panelGrid style="color:red" border="4" columns="2">
  CASE 1
  <h:outputText value="component 1"/>
  CASE 2
  <h:outputText value="component 2"/>
  <f:verbatim>CASE 3</f:verbatim>
  <c:out value="${pageScope.CASE4}" />
</h:panelGrid>
```

The preceding arrangement of faces component tags, must yield markup that will render identically to the following (assuming that `${pageScope.CASE4}` evaluates to "CASE 4" without the quotes).

```xml
<table border="4" style="color:red">
  <tbody>
    <tr><td>CASE 1</td></tr>
    <tr><td>component 1</td></tr>
  </tbody>
</table>
```
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected BodyContent</td>
<td>bodyContent: The bodyContent for this tag handler.</td>
</tr>
<tr>
<td>protected PageContext</td>
<td>pageContext: The JSP PageContext for the page we are embedded in.</td>
</tr>
<tr>
<td>protected static String</td>
<td>UNIQUE_ID_PREFIX: Used as the prefix for ids.</td>
</tr>
</tbody>
</table>

### Fields inherited from class

docs.javax.faces.webapp.UIComponentTagBase

- log

### Fields inherited from interface

docs.javax.servlet.jsp.tagext.BodyTag

- EVAL_BODY_BUFFERED, EVAL_BODY_TAG

### Fields inherited from interface

docs.javax.servlet.jsp.tagext.IterationTag

- EVAL_BODY_AGAIN

### Fields inherited from interface

docs.javax.servlet.jsp.tagext.Tag

- EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE

### Constructor Summary

UIComponentClassicTagBase()
## Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addChild(UIComponent child)</code></td>
<td>Add the component identifier of the specified component to the list of component identifiers created or located by nested <code>UIComponentTag</code> processing this request.</td>
</tr>
<tr>
<td><code>addFacet(String name)</code></td>
<td>Add the facet name of the specified facet to the list of names created or located by nested <code>UIComponentTag</code> this request.</td>
</tr>
<tr>
<td><code>addVerbatimAfterComponent(UIComponent verbatim, UIComponent component)</code></td>
<td>Add <code>verbatim</code> as a sibling of <code>component</code> in parent's child list.</td>
</tr>
<tr>
<td><code>addVerbatimBeforeComponent(UIComponent verbatim, UIComponent component)</code></td>
<td>Add <code>verbatim</code> as a sibling of <code>component</code> in parent's child list.</td>
</tr>
<tr>
<td><code>createComponent(FacesContext context, String)</code></td>
<td>Create and return a new child component by calling <code>getComponentType()</code>.</td>
</tr>
<tr>
<td><code>createVerbatimComponent()</code></td>
<td>Use the <code>Application</code> instance to create a <code>UIComponent</code> with the following characteristics.</td>
</tr>
<tr>
<td><code>createVerbatimComponentFromBodyContent()</code></td>
<td>Create a transient <code>UIOutput</code> component from the body content, of this tag instance or return null if the body content is whitespace, or the body content is empty.</td>
</tr>
<tr>
<td><code>doAfterBody()</code></td>
<td>Perform any processing necessary to handle the implications of CASE 4 in the class description.</td>
</tr>
<tr>
<td><code>doEndTag()</code></td>
<td>Perform any processing necessary to handle the implications of CASE 3 in the class description.</td>
</tr>
<tr>
<td><code>doInitBody()</code></td>
<td>Prepare for evaluation of the body.</td>
</tr>
</tbody>
</table>
Perform any processing necessary to find the UIComponent instance in the page corresponding to this tag instance in the page and, if and only if a component was not found, the tree at the proper location as expected by the page author.

**encodeBegin()**

*Deprecated.* No encoding is done during JSP page execution. Encoding is deferred until the page is executing to allow the entire tree to be built before any encoding occurs.

**encodeChildren()**

*Deprecated.* No encoding is done during JSP page execution. Encoding is deferred until the page is executing to allow the entire tree to be built before any encoding occurs.

**encodeEnd()**

*Deprecated.* No encoding is done during JSP page execution. Encoding is deferred until the page is executing to allow the entire tree to be built before any encoding occurs.

**findComponent(FacesContext context)**

Find and return the UIComponent, from the component that corresponds to this tag handler instance.

**getBodyContent()**

**getComponentInstance()**

Return the UIComponent instance that is associated with tag instance.

**getCreated()**

Return true if we dynamically created a component instance during execution of this tag.

**getCreatedComponents()**

Returns the List of UIComponent ids created or nested UIComponentTagS while processing the current tag instance.

**getDoAfterBodyValue()**

Return the flag value that should be returned from the child tag handler of this tag handler.
doAfterBody() method when it is called.

protected int 

getDoEndValue()

Return the flag value that should be returned when it is called.

protected int

getDoStartValue()

Return the flag value that should be returned when doStart() method is called.

protected FacesContext

getFacesContext()

Return the FacesContext instance for the current tag.

protected String

getFacesJspId()

If this method has been called before on the tag's lifetime (before release() was called), return the value.

protected String

getFacetName()

Return the facet name that we should be stored under, if any; otherwise, return null (indicating that we will be a child component).

protected String

getId()

Return the id value assigned by the page author.

protected int

getIndexOfNextChildTag()

Return the index of the next child to be added as a child of this tag.

String

getJspId()

Tag

getParent()

Return the Tag that is the parent of this instance.

static UICOMPONENTCLASSICTAGBASE

getParentUICOMPONENTCLASSICTAGBASE(PageContext)

Locate and return the nearest enclosing UICOMPONENTCLASSICTAGBASE if any; otherwise, return null.

JspWriter

getPreviousOut()

Get the JspWriter from our BodyContent.

protected abstract boolean

hasBinding()

Return true if this component has a non-attribute.

void

release()

Release any resources allocated during
void setBodyContent(BodyContent bodyContent)
Set the bodyContent for this tag handler.

void setId(String id)
Set the component identifier for our component.

void setJspId(String id)
Defined on JspIdConsumer.

void setPageContext(PageContext pageContext)
Set the PageContext of the page containing this component.

void setParent(Tag parent)
Set the Tag that is the parent of this instance.

protected abstract void setProperties(UIComponent component)
Override properties and attributes of the component if the corresponding properties of this tag handler were explicitly set.

protected void setupResponseWriter()
Set up the ResponseWriter for the current response, if this has not been done already.

Methods inherited from class javax.faces.webapp.UIComponentTagBase
getComponentType, getELContext, getRendererType

Methods inherited from class java.lang.Object
cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Field Detail

UNIQUE_ID_PREFIX
protected static final String UNIQUE_ID_PREFIX
Used as the prefix for ids. This is necessary to avoid uniqueness conflicts with the transient verbatim components.

See Also:
 Constant Field Values

bodyContent

protected BodyContent bodyContent

The bodyContent for this tag handler.

pageContext

protected PageContext pageContext

The JSP PageContext for the page we are embedded in.

Constructor Detail

public UIComponentClassicTagBase()

UIComponentClassicTagBase

public UIComponentClassicTagBase()

Method Detail

protected int getDoStartValue() throws JspException
doStart()

Throws

JspException: doStart()

getDoStartValue

protected int getDoStartValue()

throws JspException

Return the flag value that should be returned from the doStart() method when it is called. Subclasses may override this method to return the appropriate value.

Throws:

JspException - to cause doStart() to throw an exception

protected int getDoEndValue() throws JspException

doEnd()

Throws

JspException: doEnd()

getDoEndValue

protected int getDoEndValue()

throws JspException

Return the flag value that should be returned from the doEnd() method when it is called. Subclasses may override this method to return the appropriate value.

Throws:

JspException - to cause doEnd() to throw an exception
protected void encodeBegin() throws java.io.IOException

UIComponent encodeBegin() doStartTag()

Throws java.io.IOException: /
deprecated JSP

encodeBegin

protected void encodeBegin()
throws IOException

Deprecated. No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.

Delegate to the encodeBegin() method of our corresponding UIComponent. This method is called from doStartTag(). Normally, delegation occurs unconditionally; however, this method is abstracted out so that advanced tags can conditionally perform this call.

Throws: IOException - if an input/output error occurs

protected void encodeChildren() throws java.io.IOException

UIComponent encodeChildren() doStartTag()

Throws java.io.IOException: /
deprecated JSP
**encodeChildren**

```java
protected void encodeChildren()
    throws IOException
```

**Deprecated.** No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.

Delegate to the `encodeChildren()` method of our corresponding `UIComponent`. This method is called from `doStartTag()`. Normally, delegation occurs unconditionally; however, this method is abstracted out so that advanced tags can conditionally perform this call.

**Throws:**

- `IOException` - if an input/output error occurs

---

**protected void encodeEnd() throws java.io.IOException**

```java
UIComponent encodeEnd() doStartTag()
```

Throws: `java.io.IOException: /
deprecated JSP`

---

**encodeEnd**

```java
protected void encodeEnd()
    throws IOException
```

**Deprecated.** No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.

Delegate to the `encodeEnd()` method of our corresponding `UIComponent`. This method is called from `doStartTag()`. Normally,
delegation occurs unconditionally; however, this method is abstracted out so that advanced tags can conditionally perform this call.

**Throws:**

- `IOException` - if an input/output error occurs

--

**public void setPageContext(PageContext pageContext)**

Set the PageContext of the page containing this tag instance.

**Specified by:**

`setPageContext` in interface `Tag`

**Parameters:**

- `pageContext` - The enclosing `PageContext`

--

**public Tag getParent()**

Return the `Tag` that is the parent of this instance.
public void setParent(Tag parent)

Tag

```java

parent
```

setParent

public void setParent(Tag parent)

Set the Tag that is the parent of this instance.

Specified by:  
setParent in interface Tag

Parameters:  
parent - The new parent Tag

protected void setupResponseWriter()

javax.faces.context.ResponseWriter

setupResponseWriter

protected void setupResponseWriter()
Set up the ResponseWriter for the current response, if this has not been done already.

@deprecated.
ViewHandler.renderView(javax.faces.context.FacesContext, javax.faces.component.UIViewRoot) is now responsible for setting up the response writer. This method is now a no-op.

protected UIComponent findComponent(FacesContext context) throws JspException

1. UIComponentUIComponentUIComponent facet

2. UIComponentTagUIComponentTag Tree

3. UIComponentTag facetName UIComponent facet setProperties

4. ID UIComponentTag id

5. UIComponent id UIComponentTag

6. UIComponent setProperties()

UIComponent

1. UIComponentTag#getComponentType

2. binding FacesContext

   Application#createComponent "binding"

3. Application#createComponent

4. setProperties()

5.facet
findComponent

protected UICOMPONENT findComponent(FacesContext context) throws JspException

Find and return the UICOMPONENT, from the component tree, that corresponds to this tag handler instance. If there is no such UICOMPONENT, create one and add it as a child or facet of the UICOMPONENT associated with our nearest enclosing UICOMPONENTTag.

The process for locating or creating the component is:

1. If we have previously located this component, return it.
2. Locate the parent component by looking for a parent UICOMPONENTTag instance, and ask it for its component. If there is no parent UICOMPONENTTag instance, this tag represents the root component, so get it from the current Tree and return it.
3. If this UICOMPONENTTag instance has the facetName attribute set, ask the parent UICOMPONENT for a facet with this name. If not found, create one, call setProperties() with the new component as a parameter, and register it under this name. Return the found or created facet UICOMPONENT.
4. Determine the component id to be assigned to the new component, as follows: if this UICOMPONENTTag has an id attribute set, use that value; otherwise, generate an identifier that is guaranteed to be the same for this UICOMPONENT every time this page is processed (i.e. one based on the location of all UICOMPONENTTag instances without an id attribute set).
5. Ask the parent UICOMPONENT for a child with this identifier. If not found, create one, call setProperties() with the new component as a parameter, and register it as a child with this identifier. Return the found or created child UICOMPONENT.

When creating a component, the process is:

1. Retrieve the component type by calling UICOMPONENTTagBase.getComponentType().
2. If the component has a binding attribute, create an expression from it, and call Application.createComponent(java.lang.String) with that
expression, the `FacesContext`, and the component type. Store the expression using the key "binding".

3. Otherwise, call `Application.createComponent(java.lang.String)` with only the component type.
5. Add the new component as a child or facet of its parent

Throws: `JspException`

```java
public static UIComponentClassicTagBase getParentUIComponentClassicTagBase(PageContext context)

UIComponentClassicTagBase null
context PageContext

getParentUIComponentClassicTagBase

public static UIComponentClassicTagBase getParentUIComponentClassicTagBase(PageContext context)

    Locate and return the nearest enclosing `UIComponentClassicTagBase` if any; otherwise, return null.

Parameters:
    context - PageContext for the current page

protected int getIndexOfNextChildTag()

getIndexOfNextChildTag

protected int getIndexOfNextChildTag()
```
Description copied from class: `UIComponentTagBase`

Return the index of the next child to be added as a child of this tag. The default implementation maintains a list of created components and returns the size of the list.

Specified by:
`getIndexOfNextChildTag` in class `UIComponentTagBase`

protected void `addChild(UIComponent child)`

`addChild`

protected void `addChild(UIComponent child)`

Description copied from class: `UIComponentTagBase`

Add the component identifier of the specified `UIComponent` to the list of component identifiers created or located by nested `UIComponentTags` processing this request.

Specified by:
`addChild` in class `UIComponentTagBase`

Parameters:
child - New child whose identifier should be added

protected void `addFacet(String name)`

`addFacet`

protected void `addFacet(String name)`

Description copied from class: `UIComponentTagBase`

Add the facet name of the specified facet to the list of facet names
specified by: addFacet in class UIComponentTagBase

Parameters:
name - Facet name to be added

protected UIComponent createVerbatimComponentFromBodyContent()

UIOutput null

createVerbatimComponentFromBodyContent

protected UIComponent createVerbatimComponentFromBodyContent()

Create a transient UIOutput component from the body content, of this tag instance or return null if there is no body content, the body content is whitespace, or the body content is a comment.

protected UIOutput createVerbatimComponent()

Application

componentType javax.faces.HtmlOutputText

transient true

escape false

id FacesContext.getViewRoot().createUniqueId()
createVerbatimComponent

protected UIOutput createVerbatimComponent()

Use the Application instance to create a new component with the following characteristics.

- `componentType` is `javax.faces.HtmlOutputText`.
- `transient` is `true`.
- `escape` is `false`.
- `id` is `FacesContext.getViewRoot().createUniqueId()`.

addVerbatimBeforeComponent

protected void addVerbatimBeforeComponent(UIComponentClassicTagBase parentTag, UIComponent verbatim, UIComponent component)

Add `verbatim` as a sibling of `component` in `component` in the parent's child list. `verbatim` is added to the list at the position immediately preceding `component`.

addVerbatimBeforeComponent

protected void addVerbatimBeforeComponent(UIComponentClassicTagBase parentTag, UIComponent verbatim, UIComponent component)
protected void addVerbatimAfterComponent(UIComponentClassicTagBase parentTag, UIComponent verbatim, UIComponent component)

component verbatim component verbatim component

addVerbatimAfterComponent

protected void addVerbatimAfterComponent(UIComponentClassicTagBase p, UIComponent verbatim, UIComponent component)

Add verbatim as a sibling of component in component in the parent's child list. verbatim is added to the list at the position immediately following component.

public int doStartTag() throws JspException

(UIComponent UIOutput)
(UIComponent UIOutput CASE 1 CASE 2)

#getParentUIComponentClassicTagBase
#createVerbatimComponentFromBodyContent
#adVerbatimBeforeComponent

#getParentUIComponentClassicTagBase

getDoStartValue()

Throws JspException:
doStartTag

public int doStartTag()
    throws JspException

Perform any processing necessary to find (or create) the UIComponent instance in the view corresponding to this tag instance in the page and, if and only if a component was created, insert it into the tree at the proper location as expected by the page author. Secondarily, cause a transient UIOutput component to be created and placed in the tree before the UIComponent instance for this tag. The value of this UIOutput component must include anything covered by CASE 1 OR CASE 2 in the class description.

The default implementation, which is intended to be sufficient for most components, implements this secondary requirement by calling getParentUIComponentClassicTagBase(javax.servlet.jsp.PageContext) and calling createVerbatimComponentFromBodyContent() on the result. It then adds the returned component to the tree before the actual component for this tag instance instance by calling addVerbatimBeforeComponent(javax.faces.webapp.UIComponentClassicTagBase, javax.faces.component.UIComponent, javax.faces.component.UIComponent).

Before returning, the component is pushed onto the component stack for this response to enable the getParentUIComponentClassicTagBase(javax.servlet.jsp.PageContext) method to work properly.

The flag value to be returned is acquired by calling the getDoStartValue() method, which tag subclasses may override if they do not want the default value.

Specified by:
    doStartTag in interface Tag

Returns:
    EVAL_BODY_INCLUDE if the tag wants to process body, SKIP_BODY if it does not want to process it.

Throws:
    JspException - if an error occurs
public int doEndTag() throws JspException

CASE 3

#createVerbatimComponentFromBodyContent

- UIComponent UIComponentTag ID
- UIComponent UIComponentTag facet facet facet facet
- getDoEndValue()

Throws JspException:

doEndTag

public int doEndTag() throws JspException

Perform any processing necessary to handle the content implications of CASE 3 in the class description.

The default implementation, which is intended to be sufficient for most components, calls createVerbatimComponentFromBodyContent() on this instance and adds it as a child of the component for this tag's component at the end of the child list. In addition, the following housekeeping steps are taken.

- Retrieve from the UIComponent the set of component ids of child components created by UIComponentTag instances the last time this page was processed (if any). Compare it to the list of
children created during this page processing pass, and remove all children present in the old list but not the new. Save the new list as a component attribute so that it gets saved as part of the component's state.

- Retrieve from the UIComponent the set of facet names of facets created by UIComponentTag instances the last time this page was processed (if any). Compare it to the list of facets created during this page processing pass, and remove all facets present in the old list but not the new. Save the new list as a component attribute so that it gets saved as part of the component's state.
- Release all references to the component, and pop it from the component stack for this response, removing the stack if this was the outermost component.

The flag value to be returned is acquired by calling the getDoEndValue() method, which tag subclasses may override if they do not want the default value.

Specified by:  
doEndTag in interface Tag  
Returns:  
indication of whether to continue evaluating the JSP page.  
Throws:  
jspException - if an error occurs

public void release()

release

public void release()

Release any resources allocated during the execution of this tag handler.
protected int getDoAfterBodyValue() throws JspException

doAfterBody()

getDoAfterBodyValue

protected int getDoAfterBodyValue() throws JspException

    Return the flag value that should be returned from the doAfterBody() method when it is called. Subclasses may override this method to return the appropriate value.

    Throws: JspException

public void setBodyContent(BodyContent bodyContent)

    bodyContent JSP doInitiBody() doStartTag()
    SKIP_BODY EVAL_BODY_INCLUDE

    bodyContent BodyContent

setBodyContent

public void setBodyContent(BodyContent bodyContent)

    Set the bodyContent for this tag handler. This method is invoked by the JSP page implementation object at most once per action invocation, before doInitiBody(). This method will not be invoked for
empty tags or for non-empty tags whose doStartTag() method returns SKIP_BODY OR EVAL_BODY_INCLUDE.

Specified by:
  setBodyContent in interface BodyTag

Parameters:
  bodyContent - The new BodyContent for this tag

See Also:
  BodyTag.doInitBody(), IterationTag.doAfterBody()

public JspWriter getPreviousOut()

getBodyContent

public BodyContent getBodyContent()

public void doInitBody() throws JspException

Throws JspException
doInitBody

public void doInitBody() throws JspException

Prepare for evaluation of the body. This method is invoked by the JSP page implementation object after setBodyContent() and before the first time the body is to be evaluated. This method will not be invoked for empty tags or for non-empty tags whose doStartTag() method returns SKIP_BODY OR EVAL_BODY_INCLUDE.

Specified by: doInitBody in interface BodyTag

Throws: JspException - if an error is encountered

See Also: IterationTag.doAfterBody()

doAfterBody

public int doAfterBody() throws JspException

CASE 4

#getDoAfterBodyValue

Throws JspException:

public int doAfterBody() throws JspException

Perform any processing necessary to handle the content implications of CASE 4 in the class description.

Return result from getDoAfterBodyValue()
public void setId(String id)

setId

protected String get(Id())
getId

protected String getId()

Return the id value assigned by the page author.

getFacesJspId

protected String getFacesJspId()

#release #getJspId null #UNIQUE_ID_PREFIX jspId

getFacesJspId

protected String getFacesJspId()

If this method has been called before on this tag's useful lifetime (before release() was called), return the previously returned value. Otherwise, if getJspId() returns non-null, prepend UNIQUE_ID_PREFIX to the jspId and return the result.

getCreatedComponents

protected java.util.List<E> getCreatedComponents()

Returns the List of UIComponent ids created or located by nested
UIComponentTag while processing the current request.

```java
public void setJspId(String id)
```

Defined on JspIdConsumer. This method is called by the container before doStartTag(). The argument is guaranteed to be unique within the page.

IMPLEMENTATION NOTE: This method will detect where we are in an include and assign a unique ID for each include in a particular 'logical page'. This allows us to avoid possible duplicate ID situations for included pages that have components without explicit IDs.

Specified by:

```java
public String getJspId()
```

Specified by:

```java
public String getJspId()
```
abstract protected void setProperties(UIComponent component)

    protected void setProperties(UIComponent component)
    {
        super.setProperties(component);
        if (foo != null) {
            component.setAttribute("foo", foo);
        }
        if (bar != null) {
            component.setAttribute("bar", bar);
        }
    }

setProperties

protected abstract void setProperties(UIComponent component)

    Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set. This method must be called **ONLY** if the specified UIComponent was in fact created during the execution of this tag handler instance, and this call will occur **BEFORE** the UIComponent is added to the view.
Tag subclasses that want to support additional set properties must ensure that the base class `setProperties()` method is still called. A typical implementation that supports extra properties `foo` and `bar` would look something like this:

```java
protected void setProperties(UIComponent component) {
    super.setProperties(component);
    if (foo != null) {
        component.setAttribute("foo", foo);
    }
    if (bar != null) {
        component.setAttribute("bar", bar);
    }
}
```

The default implementation overrides the following properties:

- `rendered` - Set if a value for the `rendered` property is specified for this tag handler instance.
- `rendererType` - Set if the `getRendererType()` method returns a non-null value.

**Parameters:**

- `component` - `UIComponent` whose properties are to be overridden

```java
abstract protected UIComponent createComponent(FacesContext context, String newId)
throws JspException
```

```java
getComponentType() UIComponentTag null
binding binding Application#createComponent
Application#createComponent ID
context FacesContext
newId ID
```

`createComponent`
protected abstract UIComponent createComponent(FacesContext context, String newId) throws JspException

Create and return a new child component of the type returned by calling getComponentType(). If this UIComponentTag has a non-null binding attribute, this is done by call
Application.createComponent(java.lang.String) with the expression created for the binding attribute, and the expression will be stored on the component. Otherwise, Application.createComponent(java.lang.String) is called with only the component type. Finally, initialize the components id and other properties.

Parameters:
context - FacesContext for the current request
newId - id of the component

Throws:
JspException

abstract protected boolean hasBinding()

null binding true binding Faces 1.1 EL
EL API

hasBinding

protected abstract boolean hasBinding()

Return true if this component has a non-null binding attribute. This method is necessary to allow subclasses that expose the binding property as an Faces 1.1 style EL property as well as subclasses that expose it as an EL API property.

public UIComponent getComponentInstance()
**UIComponent**  
doStartTag()  
doEndTag()

### getComponentInstance

```java
public UIComponent getComponentInstance()
```

Return the `UIComponent` instance that is associated with this tag instance. This method is designed to be used by tags nested within this tag, and only returns useful results between the execution of `doStartTag()` and `doEndTag()` on this tag instance.

**Specified by:**  
`getComponentInstance` in class `UIComponentTagBase`

### public boolean getCreated()

```java
true doStartTag() doEndTag()
```

### getCreated

```java
public boolean getCreated()
```

Return `true` if we dynamically created a new component instance during execution of this tag. This method is designed to be used by tags nested within this tag, and only returns useful results between the execution of `doStartTag()` and `doEndTag()` on this tag instance.

**Specified by:**  
`getCreated` in class `UIComponentTagBase`

### protected FacesContext getFacesContext()

```java
protected FacesContext getFacesContext()
```
getFacesContext

protected FacesContext getFacesContext()

Description copied from class: UIComponentTagBase

Return the FacesContext instance for the current request. This value will be non-null only from the beginning of doStartTag() through the end of doEndTag() for each tag instance.

Specified by:
    getFacesContext in class UIComponentTagBase

protected String getFacetName()

facet null

getFacetName

protected String getFacetName()

    Return the facet name that we should be stored under, if any; otherwise, return null (indicating that we will be a child component).
javax.faces.webapp  Class UIComponentELTag

java.lang.Object  
  \ javax.faces.webapp.UIComponentTagBase  
  \ javax.faces.webapp.UIComponentClassicTagBase  
  \ javax.faces.webapp.UIComponentELTag

All Implemented Interfaces:
  BodyTag, IterationTag, JspIdConsumer, JspTag, Tag

public abstract class UIComponentELTag
extends UIComponentClassicTagBase
implements Tag

Extends: UIComponentTagBase > UIComponentClassicTagBase
Implements: Tag

UIComponentELTag  EL API

Faces 1.2  JSP 2.1

UIComponentELTag specializes its superclass to allow for properties that take their values from EL API expressions.

This tag is designed for use with Faces version 1.2 and JSP version 2.1 containers.

Field Summary

Fields inherited from class javax.faces.webapp.UIComponentClassicTagBase
bodyContent, pageContext, UNIQUE_ID_PREFIX
Constructor Summary

UICOMPONENTELTAG()
protected void setProperties(UIComponent component)

Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set.

void setRendered(ValueExpression rendered)

Set an override for the rendered attribute.

Methods inherited from class javax.faces.webapp.UIComponentClassicTagBase

addChild, addFacet, addVerbatimAfterComponent, addVerbatimBeforeComponent, createVerbatimComponent, createVerbatimComponentFromBodyContent, doAfterBody, doEndTag, doInitBody, doStartTag, encodeBegin, encodeChildren, encodeEnd, findComponent, getBodyContent, getComponentInstance, getCreated, getCreatedComponents, getDoAfterBodyValue, getDoEndValue, getDoStartValue, getFacesContext, getFacesJspId, getFacetName, getId, getIndexOfNextChildTag, getJspId, getParent, getParentUIComponentClassicTagBase, getPreviousOut, setBodyContent, setId, setJspId, setPageContext, setParent, setupResponseWriter

Methods inherited from class javax.faces.webapp.UIComponentTagBase

getComponentType, getRendererType

Methods inherited from class java.lang.Object

close, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.jsp.tagext.Tag

doEndTag, doStartTag, getParent, setPageContext, setParent

Constructor Detail

public UIComponentELTag()
UIComponentELTag

public UIComponentELTag()

Method Detail

public void setBinding(ValueExpression binding) throws JspException

Parameter:

binding - The new value expression

Throws:

JspException - if an error occurs

protected boolean hasBinding()

hasBinding

protected boolean hasBinding()
Return true if this component has a non-null binding attribute. This method is necessary to allow subclasses that expose the binding property as an Faces 1.1 style EL property as well as subclasses that expose it as an EL API property.

**Specified by:**
- hasBinding in class UICOMPONENTCLASSICTAGBASE

```java
public void setRendered(ValueExpression rendered)
```

*rendered*

**setRendered**

```java
public void setRendered(ValueExpression rendered)
```

Set an override for the rendered attribute.

**Parameters:**
- rendered - The new value for rendered attribute

```java
protected ELContext getELContext()
```

**getELContext**

```java
protected ELContext getELContext()
```
Return the `ELContext` for the `FacesContext` for this request.

This is a convenience for `getFacesContext().getELContext()`.

**Overrides:**
- `getELContext` in class `UIComponentTagBase`

---

```java
public void release()
```

**release**

```java
public void release()
```

Release any resources allocated during the execution of this tag handler.

**Specified by:**
- `release` in interface `Tag`

**Overrides:**
- `release` in class `UIComponentClassicTagBase`

---

```java
protected void setProperties(UIComponent component)
```

```java
protected void setProperties(UIComponent component) {
    super.setProperties(component);
    if (foo != null) {
        component.setAttribute("foo", foo);
    }
    if (bar != null) {
```
protected void setProperties(UIComponent component) {
    super.setProperties(component);
    if (foo != null) {
        component.setAttribute("foo", foo);
    }
    if (bar != null) {
        component.setAttribute("bar", bar);
    }
}

The default implementation overrides the following properties:

- rendered - rendered
- rendererType - getRendererType() null
- rendered - Set if a value for the rendered property is specified for this tag handler instance.
- rendererType - Set if the getRendererType() method returns a non-null value.

Specified by:

setProperties in class UIComponentClassicTagBase

Parameters:

component - UIComponent whose properties are to be overridden

protected UIComponent.createComponent(FacesContext context, String newId) throws JspException

```
getComponentType()  UIComponentELTag null
binding  binding  ValueExpression
Application#createComponent  ValueExpression
Application#createComponent ID

context  FacesContext
newId  ID
```

createComponent

protected UIComponent createComponent(FacesContext context, String newId) throws JspException

Create and return a new child component of the type returned by calling getComponentType(). If this UIComponentELTag has a non-null binding attribute, this is done by call Application.createComponent(java.lang.String) with the ValueExpression created for the binding attribute, and the ValueExpression will be stored on the component. Otherwise, Application.createComponent(java.lang.String) is called with only the component type. Finally, initialize the components id and other properties.
Specified by:
createComponent in class UIComponentClassicTagBase

Parameters:
context - FacesContext for the current request
newId - id of the component

Throws:
JspException

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
public abstract class `UIComponentTag`

extends `UIComponentClassicTagBase`

implements `Tag`

`UIComponentTag` is the base class for all JSP custom actions that correspond to user interface components in a page that is rendered by JavaServer Faces.
In this version of the specification, UIComponentTag extends UIComponentClassicTagBase to add properties that use the Faces 1.1 Expression Language.

### Field Summary

**Fields inherited from class javax.faces.webapp.UIComponentClassicTagBase**
- `bodyContent`, `pageContext`, `UNIQUE_ID_PREFIX`

**Fields inherited from class javax.faces.webapp.UIComponentTagBase**
- `log`

**Fields inherited from interface javax.servlet.jsp.tagext.Tag**
- `EVAL_BODY_INCLUDE`, `EVAL_PAGE`, `SKIP_BODY`, `SKIP_PAGE`

**Fields inherited from interface javax.servlet.jsp.tagext.BodyTag**
- `EVAL_BODY_BUFFERED`, `EVAL_BODY_TAG`

**Fields inherited from interface javax.servlet.jsp.tagext.IterationTag**
- `EVAL_BODY_AGAIN`

### Constructor Summary

- `UIComponentTag()`  
  `Deprecated.`

### Method Summary

```java
protected UIComponent createComponent(FacesContext context, String newId)
```

* Deprecated. Implement `createComponent` using Faces 1.1 EL API.*
### Methods

**Deprecated.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getParentUIComponentTag(PageContext context)</code></td>
<td>Locate and return the nearest enclosing <code>UIComponentTag</code> if any; otherwise, return null.</td>
</tr>
<tr>
<td><strong>hasBinding()</strong></td>
<td>Return true if this component has a non-null binding attribute.</td>
</tr>
<tr>
<td><strong>isSuppressed()</strong></td>
<td><strong>Deprecated.</strong></td>
</tr>
<tr>
<td><strong>isValueReference(String value)</strong></td>
<td>Return true if the specified value conforms to the syntax requirements of a value binding expression.</td>
</tr>
<tr>
<td><strong>release()</strong></td>
<td><strong>Deprecated.</strong> Release any resources allocated during the execution of this tag handler.</td>
</tr>
<tr>
<td><strong>setBinding(String binding)</strong></td>
<td><strong>Deprecated.</strong> Set the value binding expression for our component.</td>
</tr>
<tr>
<td><strong>setProperties(UIComponent component)</strong></td>
<td><strong>Deprecated.</strong> Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set.</td>
</tr>
<tr>
<td><strong>setRendered(String rendered)</strong></td>
<td><strong>Deprecated.</strong> Set an override for the rendered attribute.</td>
</tr>
</tbody>
</table>

**Methods inherited from class javax.faces.webapp.UIComponentClassicTagBase**

- `addChild`, `addFacet`, `addVerbatimAfterComponent`,
- `addVerbatimBeforeComponent`, `createVerbatimComponent`,
- `createVerbatimComponentFromBodyContent`, `doAfterBody`, `doEndTag`,
- `doInitBody`, `doStartTag`, `encodeBegin`, `encodeChildren`, `encodeEnd`,
- `findComponent`, `getBodyContent`, `getComponentInstance`, `getCreated`,
- `getCreatedComponents`, `getDoAfterBodyValue`, `getDoEndValue`,
- `getDoStartValue`, `getFacesContext`, `getFacesJspId`, `getFacetName`,
- `getId`, `getIndexOfNextChildTag`, `getJspId`, `getParent`,
- ...
getParentUIComponentClassicTagBase, getPreviousOut, setBodyContent, setId, setJspId, setPageContext, setParent, setupResponseWriter

Methods inherited from class javax.faces.webapp.UIComponentTagBase
getComponentType, getELContext, getRendererType

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.servlet.jsp.tagext.Tag
doEndTag, doStartTag, getParent, setPageContext, setParent

Constructor Detail

public UIComponentTag()

UIComponentTag

public UIComponentTag()

    Deprecated.

Method Detail

public void setBinding(String binding) throws JspException

    binding
Throws IllegalArgumentException:

**setBinding**

```java
public void setBinding(String binding) throws JspException
```

Deprecated.

Set the value binding expression for our component.

**Parameters:**
- binding - The new value binding expression

**Throws:**
- IllegalArgumentException - if the specified binding is not a valid value binding expression.
- JspException

---

**protected boolean hasBinding()**

**hasBinding**

```java
protected boolean hasBinding()
```

**Deprecated.**

Description copied from class: `UIComponentClassicTagBase`

Return true if this component has a non-null binding attribute. This method is necessary to allow subclasses that expose the binding property as an Faces 1.1 style EL property as well as subclasses that expose it as an EL API property.

**Specified by:**
- `hasBinding` in class `UIComponentClassicTagBase`
public void setRendered(String rendered)

rendered

setRendered

public void setRendered(String rendered)

 Deprecated.

 Set an override for the rendered attribute.

 Parameters:
 rendered - The new value for rendered attribute

protected boolean isSuppressed()

isSuppressed

protected boolean isSuppressed()

 Deprecated.

public static boolean isValueReference(String value)

 true ` *

 value

 Throws

 NullPointerException: value null
public static boolean isValueReference(String value)

Deprecated.

Return true if the specified value conforms to the syntax requirements of a value binding expression. Such expressions `*` may be used on most component tag attributes to signal a desire for deferred evaluation of the attribute or property value to be set on the underlying UIComponent.

Parameters:
   value - The value to evaluate

Throws:
   NullPointerException - if value is null

public void release()

release

public void release()

Deprecated.

Release any resources allocated during the execution of this tag handler.

Specified by:
   release in interface Tag

Overrides:
   release in class UIComponentClassicTagBase
protected void setProperties(UIComponent component)

Description copied from class: UIComponentClassicTagBase

Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set. This method must be called ONLY if the specified UIComponent was in fact created during the execution of this tag handler instance, and this call will occur BEFORE the UIComponent is added to the view.

Tag subclasses that want to support additional set properties must ensure that the base class setProperties() method is still called. A typical implementation that supports extra properties foo and bar would look something like this:

protected void setProperties(UIComponent component) {
    super.setProperties(component);
    if (foo != null) {
        component.setAttribute("foo", foo);
    }
    if (bar != null) {
        component.setAttribute("bar", bar);
    }
}

The default implementation overrides the following properties:

- rendered - Set if a value for the rendered property is specified for this tag handler instance.
- rendererType - Set if the getRendererType() method returns a non-null value.
protected **UIComponent** `createComponent`(FacesContext context, String newId)

Faces 1.1 EL API

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Exception</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createComponent</code></td>
<td><code>FacesContext context, String newId</code></td>
<td>NullPointerException</td>
</tr>
</tbody>
</table>

**createComponent**

protected **UIComponent** `createComponent`(FacesContext context, String newId)

**Deprecated.**

Implement `createComponent` using Faces 1.1 EL API.

Specified by: `createComponent` in class **UIComponentClassicTagBase**

Parameters:
- `context` - FacesContext for the current request
- `newId` - id of the component

public static **UIComponentTag** `getParentUIComponentTag`(PageContext context)

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getParentUIComponentTag</code></td>
<td><code>PageContext context</code></td>
<td>null</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>context</code></td>
<td></td>
<td>PageContext</td>
</tr>
</tbody>
</table>
getParentUIComponentTag

public static UIComponentTag getParentUIComponentTag(PageContext context)

Deprecated.

Locate and return the nearest enclosing UIComponentTag if any; otherwise, return null.

Parameters:
context - PageContext for the current page

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
javax.faces.webapp  Class UIComponentTagBase

java.lang.Object
   ↓ javax.faces.webapp.UIComponentTagBase

All Implemented Interfaces:
   JspTag

Direct Known Subclasses:
   UIComponentClassicTagBase

public abstract class UIComponentTagBase
  extends Object
  implements JspTag

Implements: JspTag
Extended by: UIComponentClassicTagBase

UIComponentTagBase  JSP  javax.faces.component.UIComponent
                      UIComponentELTag  UIComponentTag  JSP

UIComponentTagBase is the base class for all JSP tags that correspond to a
UIComponent instance in the view. This base class allows a single view to
be described in a JSP page consisting of both UIComponentELTag and
UIComponentTag instances.

Field Summary

| protected static Logger log |

Constructor Summary
**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>addChild(UIComponent child)</code></td>
<td>Add the component identifier of the specified <code>UIComponent</code> to the list of component identifiers created or located by nested <code>UIComponentTag</code> processing this request.</td>
</tr>
<tr>
<td><code>addFacet(String name)</code></td>
<td>Add the facet name of the specified facet to the list of facet names created or located by nested <code>UIComponentTag</code> processing this request.</td>
</tr>
<tr>
<td><code>getComponentInstance()</code></td>
<td>Return the <code>UIComponent</code> instance that is associated with this tag instance.</td>
</tr>
<tr>
<td><code>getComponentType()</code></td>
<td>Return the component type for the component that is or will be bound to this tag.</td>
</tr>
<tr>
<td><code>getCreated()</code></td>
<td>Return <code>true</code> if we dynamically created a new component instance during execution of this tag.</td>
</tr>
<tr>
<td><code>getELContext()</code></td>
<td>Return the <code>ELContext</code> for the <code>FacesContext</code> for this request.</td>
</tr>
<tr>
<td><code>getFacesContext()</code></td>
<td>Return the <code>FacesContext</code> instance for the current request.</td>
</tr>
<tr>
<td><code>getIndexOfNextChildTag()</code></td>
<td>Return the index of the next child to be added as a child of this tag.</td>
</tr>
<tr>
<td><code>getRendererType()</code></td>
<td>Return the <code>rendererType</code> property that selects the <code>Renderer</code> to be used for encoding this component, or <code>null</code> to ask the component to render itself directly.</td>
</tr>
<tr>
<td><code>setId(String id)</code></td>
<td></td>
</tr>
</tbody>
</table>
abstract void Set the component identifier for the component corresponding to this tag instance.

### Methods inherited from class java.lang.Object

*clone*, *equals*, *finalize*, *getClass*, *hashCode*, *notify*, *notifyAll*, *toString*, *wait*, *wait*, *wait*

### Field Detail

**log**

protected static Logger log

### Constructor Detail

**public UIComponentTagBase()**

**UIComponentTagBase**

**public UIComponentTagBase()**

### Method Detail

abstract protected FacesContext getFacesContext()

*FacesContext* doStartTag() doEndTag() null

getFacesContext
protected abstract FacesContext getFacesContext()

Return the FacesContext instance for the current request. This value will be non-null only from the beginning of doStartTag() through the end of doEndTag() for each tag instance.

protected ELContext getELContext()

FacesContext ELContext

g getFacesContext().getELContext()

getELContext

protected ELContext getELContext()

Return the ELContext for the FacesContext for this request.

This is a convenience for getFacesContext().getELContext().

abstract protected void addChild(UIComponent child)

UIComponent UICOMPONENTTAG

child

addChild

protected abstract void addChild(UIComponent child)

Add the component identifier of the specified UIComponent to the list of component identifiers created or located by nested UICOMPONENTTAGs processing this request.
abstract protected void addFacet(String name)

Add the facet name of the specified facet to the list of facet names created or located by nested UIComponentTag processing this request.

Parameters:
name - Facet name to be added

abstract public void setId(String id)

Throws
IllegalArgumentException

setId

public abstract void setId(String id)

Set the component identifier for the component corresponding to this
tag instance. If the argument begins with `UIViewRoot.UNIQUE_ID_PREFIX` throw an `IllegalArgumentException`.

**Parameters:**
- `id` - The new component identifier. This may not start with `UIViewRoot.UNIQUE_ID_PREFIX`.

**Throws:**
- `IllegalArgumentException` - if the argument is non-null and starts with `UIViewRoot.UNIQUE_ID_PREFIX`.

---

```java
abstract public String getComponentType()
```

**getComponentType**

```java
public abstract String getComponentType()
```

Return the component type for the component that is or will be bound to this tag. This value can be passed to `javax.faces.application.Application#createComponent(UIComponent)` to create the `UIComponent` instance for this tag. Subclasses must override this method to return the appropriate value.

---

```java
abstract public String getRendererType()
```

**getRendererType**

```java
public abstract String getRendererType()
```

```java
rendererType Renderer null
```

getRendererType
Return the `rendererType` property that selects the `Renderer` to be used for encoding this component, or `null` to ask the component to render itself directly. Subclasses must override this method to return the appropriate value.

```
abstract public UIComponent getComponentInstance()
```

```
UIComponent doStartTag() doEndTag()
```

**getComponentInstance**

```
public abstract UIComponent getComponentInstance()
```

Return the `UIComponent` instance that is associated with this tag instance. This method is designed to be used by tags nested within this tag, and only returns useful results between the execution of `doStartTag()` and `doEndTag()` on this tag instance.

```
abstract public boolean getCreated()
```

```
true doStartTag() doEndTag()
```

**getCreated**

```
public abstract boolean getCreated()
```

Return `true` if we dynamically created a new component instance during execution of this tag. This method is designed to be used by tags nested within this tag, and only returns useful results between the execution of `doStartTag()` and `doEndTag()` on this tag instance.
abstract protected int getIndexOfNextChildTag()

getIndexOfNextChildTag

protected abstract int getIndexOfNextChildTag()

Return the index of the next child to be added as a child of this tag. The default implementation maintains a list of created components and returns the size of the list.
javax.faces.component  **Class UIData**

java.lang.Object  
  └ javax.faces.component.UIComponent  
      └ javax.faces.component.UIComponentBase  
      └ javax.faces.component.UIData

**All Implemented Interfaces:**  
NamingContainer, StateHolder

**Direct Known Subclasses:**  
HtmlDataTable

---

class **UIData**

extends UIComponentBase
implements NamingContainer

Extends: UIComponent > UIComponentBase  
Implements: NamingContainer  
Extended by: HtmlDataTable

**UIData**  
DataModel   Collection   UIColumn   renderer

rendererType   javax.faces.Table   setRendererType()

**UIData** is a **UIComponent** that supports data binding to a collection of data objects represented by a **DataModel** instance, which is the current value of this component itself (typically established via a **ValueExpression**). During iterative processing over the rows of data in the data model, the object for the current row is exposed as a request attribute under the key specified by the **var** property.
Only children of type `UIColumn` should be processed by renderers associated with this component.

By default, the `rendererType` property is set to `javax.faces.Table`. This value can be changed by calling the `setRendererType()` method.

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <code>String</code> COMPONENT_FAMILY</td>
<td>The standard component family for this component.</td>
</tr>
<tr>
<td>static <code>String</code> COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class `javax.faces.component.UIComponent` bindings

### Fields inherited from interface `javax.faces.component.NamingContainer` SEPARATOR_CHAR

### Constructor Summary

**UIData()**  
Create a new `UIData` instance with default property values.

### Method Summary

**void broadcast(FacesEvent event)**
Override the default `UIComponentBase.broadcast(javax.faces.event.FacesEvent)` procedure to unwrap any wrapped `FacesEvent` and reset the current row index before the event is actually broadcast.

**void encodeBegin(FacesContext context)**
In addition to the default behavior, ensure that any saved state for our child input components is discarded unless it is needed.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getClientId(FacesContext context)</code></td>
<td>Return a client identifier for this component that includes current value of the <code>rowIndex</code> property, if it is not set to -1.</td>
</tr>
<tr>
<td><code>getDataModel()</code></td>
<td>Return the internal <code>DataModel</code> object representing the data that we will iterate over in this component's rendering.</td>
</tr>
<tr>
<td><code>getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>getFirst()</code></td>
<td>Return the zero-relative row number of the first row to be displayed.</td>
</tr>
<tr>
<td><code>getFooter()</code></td>
<td>Return the footer facet of this component (if any).</td>
</tr>
<tr>
<td><code>getHeader()</code></td>
<td>Return the header facet of this component (if any).</td>
</tr>
<tr>
<td><code>getRowCount()</code></td>
<td>Return the number of rows in the underlying data model.</td>
</tr>
<tr>
<td><code>getRowData()</code></td>
<td>Return the data object representing the data for the currently selected row index, if any.</td>
</tr>
<tr>
<td><code>getRowIndex()</code></td>
<td>Return the zero-relative index of the currently selected row.</td>
</tr>
<tr>
<td><code>getRows()</code></td>
<td>Return the number of rows to be displayed, or zero for all remaining rows in the table.</td>
</tr>
<tr>
<td><code>getValue()</code></td>
<td>Return the value of the UIData.</td>
</tr>
<tr>
<td><code>getVar()</code></td>
<td>Return the request-scope attribute under which the data for the current row will be exposed when iterating.</td>
</tr>
<tr>
<td><code>invokeOnComponent(FacesContext context, String clientId, ContextCallback callback)</code></td>
<td>Override behavior from <code>UIComponentBase.invokeOnComponent(javax.faces.context.FacesContext, String clientId, ContextCallback callback)</code></td>
</tr>
</tbody>
</table>
special care for positioning the data properly before finding the component and invoking the callback on it.

`boolean isRowAvailable()`  
Return a flag indicating whether there is `rowData` available at the current `rowIndex`.

`void processDecodes(FacesContext context)`  
Override the default `UIComponentBase.processDecodes(javax.faces.context.FacesContext)` processing to perform the following steps.

`void processUpdates(FacesContext context)`  
Override the default `UIComponentBase.processUpdates(javax.faces.context.FacesContext)` processing to perform the following steps.

`void processValidators(FacesContext context)`  
Override the default `UIComponentBase.processValidators(javax.faces.context.FacesContext)` processing to perform the following steps.

`void queueEvent(FacesEvent event)`  
Override the default `UIComponentBase.queueEvent(javax.faces.event.FacesEvent)` processing to wrap any queued events in a wrapper so that we can reset the row index in `broadcast()`.

`void restoreState(FacesContext context, Object state)`  
Perform any processing required to restore the state from the entries in the state `Object`.

`Object saveState(FacesContext context)`  
Gets the state of the instance as a `Serializable Object`.

`protected void setDataModel(DataModel dataModel)`  
Set the internal `DataModel`.

`void setFirst(int first)`  
Set the zero-relative row number of the first row to be displayed.

`void setFooter(UIComponent footer)`  
Set the footer facet of this component.

`void setHeader(UIComponent header)`  
Set the header facet of this component.
void **setRowIndex**(int rowIndex)
Set the zero relative index of the current row, or -1 to indicate no row is currently selected, by implementing the following algorithm.

void **setRows**(int rows)
Set the number of rows to be displayed, or zero for all remaining rows in the table.

void **setValue**(Object value)
Set the value of the UIData.

void **setValueBinding**(String name, ValueBinding binding)
 Deprecated. This has been replaced by setValueExpression(java.lang.String, javax.el.ValueExpression)

void **setValueExpression**(String name, ValueExpression binding)
Set the ValueExpression used to calculate the value for the specified attribute or property name, if any.

void **setVar**(String var)
Set the request-scope attribute under which the data object for the current row will be exposed when iterating.

**Methods inherited from class javax.faces.component.UIComponentBase**
addFacesListener, decode, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, isRendered, isTransient, processRestoreState, processSaveState, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient

**Methods inherited from class javax.faces.component.UIComponent**
encodeAll, getContainerClientId, getValueExpression

**Methods inherited from class java.lang.Object**
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

- [Constant Field Values](#)

**COMPONENT_FAMILY**

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**

- [Constant Field Values](#)

### Constructor Detail

**public UIData()**

- [UIData](#)

**UIData**

```java
public UIData()
```
Create a new `UIData` instance with default property values.

### Method Detail

**public String getFamily()**

**getFamily**

**public String getFamily()**

**Description copied from class:** `UIComponent`

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the `rendererType` property, may be used to select the appropriate `Renderer` for this component instance.

**Specified by:**

`getFamily` in class `UIComponent`

---

**public int getFirst()**

**0**

**getFirst**

**public int getFirst()**

Return the zero-relative row number of the first row to be displayed.

---

**public void setFirst(int first)**
**setFirst**

```java
public void setFirst(int first)
```

Set the zero-relative row number of the first row to be displayed.

**Parameters:**
- `first` - New first row number

**Throws:**
- `IllegalArgumentException` - if `first` is negative

**getFooter**

```java
public UIComponent getFooter()
```

```java
facet getFacet("footer")
```

**getFooter**

```java
public UIComponent getFooter()
```

Return the footer facet of this component (if any). A convenience method for `getFacet("footer")`.

**setFooter**

```java
public void setFooter(UIComponent footer)
```

```java
facet getFacets().put("footer", footer)
```

** Throws:**
- `NullPointerException` - if `footer` is null
**setFooter**

```java
public void setFooter(UIComponent footer)
```

Set the footer facet of this component. A convenience method for `getFacets().put("footer", footer)`.

**Parameters:**
- `footer` - the new footer facet

**Throws:**
- `NullPointerException` - if `footer` is null

**getHeader**

```java
public UIComponent getHeader()
```

```java
facet getFacet("header")
```

**getHeader**

```java
public UIComponent getHeader()
```

Return the header facet of this component (if any). A convenience method for `getFacet("header")`.

**setHeader**

```java
public void setHeader(UIComponent header)
```

```java
facet getFacets().put("header", header)
```

**setHeader**

- `header` - The new header facet
- `getFacets().put("header", header)`

**Throws:**
- `NullPointerException: header` - if header is null

---

**setFooter**

```java
public void setFooter(UIComponent footer)
```

Set the footer facet of this component. A convenience method for `getFacets().put("footer", footer)`.

**Parameters:**
- `footer` - the new footer facet

**Throws:**
- `NullPointerException` - if `footer` is null

**getHeader**

```java
public UIComponent getHeader()
```

```java
facet getFacet("header")
```

**getHeader**

```java
public UIComponent getHeader()
```

Return the header facet of this component (if any). A convenience method for `getFacet("header")`.

**setHeader**

```java
public void setHeader(UIComponent header)
```

```java
facet getFacets().put("header", header)
```

**setHeader**

- `header` - The new header facet
- `getFacets().put("header", header)`

**Throws:**
- `NullPointerException: header` - if header is null

---

**setFooter**

```java
public void setFooter(UIComponent footer)
```

Set the footer facet of this component. A convenience method for `getFacets().put("footer", footer)`.

**Parameters:**
- `footer` - the new footer facet

**Throws:**
- `NullPointerException` - if `footer` is null

**getHeader**

```java
public UIComponent getHeader()
```

```java
facet getFacet("header")
```

**getHeader**

```java
public UIComponent getHeader()
```

Return the header facet of this component (if any). A convenience method for `getFacet("header")`.

**setHeader**

```java
public void setHeader(UIComponent header)
```

```java
facet getFacets().put("header", header)
```

**setHeader**

- `header` - The new header facet
- `getFacets().put("header", header)`

**Throws:**
- `NullPointerException: header` - if header is null
public void setHeader(UIComponent header)

Set the header facet of this component. A convenience method for
getFacets().put("header", header).

Parameters:
header - the new header facet

Throws:
NullPointerException - if header is null

public boolean isRowAvailable()

rowIndex rowData wrappedData false

Throws         FacesException:

isRowAvailable

public boolean isRowAvailable()

Return a flag indicating whether there is rowData available at the
current rowIndex. If no wrappedData is available, return false.

Throws:
FacesException - if an error occurs getting the row availability

public int getRowCount()

-1

Throws         FacesException:

getRowCount
public int getRowCount()

    Return the number of rows in the underlying data model. If the number of available rows is unknown, return -1.

    Throws:  
    FacesException - if an error occurs getting the row count

public Object getRowData()

    Throws  FacesException:
    Throws  IllegalArgumentException:

getRowData

public Object getRowData()

    Return the data object representing the data for the currently selected row index, if any.

    Throws:  
    FacesException - if an error occurs getting the row data
    IllegalArgumentException - if now row data is available at the currently specified row index

public int getRowIndex()

    0 -1

    Throws  FacesException:

getRowIndex
public int getRowIndex()

    Return the zero-relative index of the currently selected row. If we are not currently positioned on a row, return -1. This property is not enabled for value binding expressions.

    Throws:
        FacesException - if an error occurs getting the row index

public void setRowIndex(int rowIndex)

0 -1 Collection
getRowData() isRowAvailable()
Throws **FacesException**:

Throws **IllegalArgumentException**: rowIndex -1

**setRowIndex**

```java
class UIData {

    public void setRowIndex(int rowIndex) {

        Set the zero relative index of the current row, or -1 to indicate that no row is currently selected, by implementing the following algorithm. It is possible to set the row index at a value for which the underlying data collection does not contain any row data. Therefore, callers may use the **isRowAvailable()** method to detect whether row data will be available for use by the **getRowData()** method.

        - Save current state information for all descendant components (as described below).
        - Store the new row index, and pass it on to the **DataModel** associated with this **UIData** instance.
        - If the new **rowIndex** value is -1:
            - If the **var** property is not null, remove the corresponding request scope attribute (if any).
            - Reset the state information for all descendant components (as described below).
        - If the new **rowIndex** value is not -1:
            - If the **var** property is not null, call **getRowData()** and expose the resulting data object as a request scope attribute whose key is the **var** property value.
            - Reset the state information for all descendant components (as described below).

To save current state information for all descendant components, **UIData** must maintain per-row information for each descendant as follows:

    - If the descendant is an instance of **EditableValueHolder**, save the state of its **localValue** property.
    - If the descendant is an instance of **EditableValueHolder**, save the state of the **localValueSet** property.
    - If the descendant is an instance of **EditableValueHolder**, save the
state of the valid property.

- If the descendant is an instance of EditableValueHolder, save the state of the submittedValue property.

To restore current state information for all descendant components, **UIData** must reference its previously stored information for the current rowIndex and call setters for each descendant as follows:

- If the descendant is an instance of EditableValueHolder, restore the value property.
- If the descendant is an instance of EditableValueHolder, restore the state of the localValueSet property.
- If the descendant is an instance of EditableValueHolder, restore the state of the valid property.
- If the descendant is an instance of EditableValueHolder, restore the state of the submittedValue property.

**Parameters:**

- rowIndex - The new row index value, or -1 for no associated row

**Throws:**

- FacesException - if an error occurs setting the row index
- IllegalArgumentException - if rowIndex is less than -1

---

**public int getRows()**

0 0

**getRows**

**public int getRows()**

Return the number of rows to be displayed, or zero for all remaining rows in the table. The default value of this property is zero.

---

**public void setRows(int rows)**
setRows

public void setRows(int rows)

Set the number of rows to be displayed, or zero for all remaining rows in the table.

Parameters:
rows - New number of rows

Throws:
IllegalArgumentException - if rows is negative

public String getVar()

gVar

public String getVar()

Return the request-scope attribute under which the data object for the current row will be exposed when iterating. This property is not enabled for value binding expressions.

public void setVar(String var)
setVar

public void setVar(String var)

Set the request-scope attribute under which the data object for the current row will be exposed when iterating.

Parameters:
  var - The new request-scope attribute name

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.
The return from this method must be `Serializable`.

**Specified by:**
- `saveState` in interface `StateHolder`

**Overrides:**
- `saveState` in class `UIComponentBase`

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface: `StateHolder`**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement `StateHolder` (such as a `UIComponent` with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
- `restoreState` in interface `StateHolder`

**Overrides:**
- `restoreState` in class `UIComponentBase`

```java
public Object getValue()
```

**UI Data**
- `DataModel`
- `DataModel`
- `UIData`

-
- java.util.List
- java.sql.ResultSet
- javax.servlet.jsp.jstl.sql.Result

**ScalarDataModel**

**getValue**

```java
generic public Object getValue()
```

Return the value of the UIData. This value must either be of type `DataModel`, or a type that can be adapted into a `DataModel`. UIData will automatically adapt the following types:

- Arrays
- java.util.List
- java.sql.ResultSet
- javax.servlet.jsp.jstl.sql.Result

All other types will be adapted using the `ScalarDataModel` class, which will treat the object as a single row of data.

**public void setValue(Object value)**

**setValue**

```java
void setValue(Object value)
```

Set the value of the UIData. This value must either be of type `DataModel`, or a type that can be adapted into a `DataModel`. 
public void setValueBinding(String name, ValueBinding binding)

"name" "value" "var" "rowIndex"

ValueBinding ValueExpression

name ValueBinding
binding ValueExpression null ValueBinding

Throws IllegalArgumentException: name id parent var rowIndex

Throws NullPointerException: name null

deprecated

#setValueExpression(java.lang.String, javax.el.ValueExpression)

setValueBinding

public void setValueBinding(String name, ValueBinding binding)

Deprecated. This has been replaced by


If "name" is something other than "value", "var", or "rowIndex", rely on the superclass conversion from ValueBinding to ValueExpression.

Overrides:

setValueBinding in class UICOMPONENTBASE

Parameters:

name - Name of the attribute or property for which to set a ValueBinding
binding - The ValueBinding to set, or null to remove any currently set ValueBinding

Throws:
public void setValueExpression(String name, ValueExpression binding)

<table>
<thead>
<tr>
<th>DataModel</th>
<th>ValueExpression</th>
<th>value</th>
<th>ValueExpression</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>ValueExpression</td>
<td>null</td>
<td>ValueExpression</td>
</tr>
<tr>
<td>binding</td>
<td>ValueExpression</td>
<td>null</td>
<td>ValueExpression</td>
</tr>
</tbody>
</table>

Throws: 
IllegalArgumentException: name id parent var var

Throws: 
NullPointerException: name null

since 1.2

setValueExpression

public void setValueExpression(String name, ValueExpression binding)

Set the ValueExpression used to calculate the value for the specified attribute or property name, if any. In addition, if a ValueExpression is set for the value property, remove any synthesized DataModel for the data previously bound to this component.

Overrides: 
setValueExpression in class UIComponent

Parameters: 
name - Name of the attribute or property for which to set a ValueExpression
binding - The ValueExpression to set, or null to remove any currently set ValueExpression

Throws: 
IllegalArgumentException - if name is one of id, parent, var, or rowIndex
NullPointerException - if name is null

Since:
1.2

public String getClientId(FacesContext context)

Throws
Null POINTER EXCEPTION: context null

getClientId

public String getClientId(FacesContext context)

Return a client identifier for this component that includes the current value of the rowIndex property, if it is not set to -1. This implies that multiple calls to getClientId() may return different results, but ensures that child components can themselves generate row-specific client identifiers (since UIData is a NamingContainer).

Overrrides:
getClientId in class UIComponentBase

Parameters:
context - The FacesContext for the current request

Throws:
Null POINTER EXCEPTION - if context is null

public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback) throws FacesException

this.getClientId() contextCallback context this
public boolean invokeOnComponent(FacesContext context, String clientId, ContextCallback callback)
    throws FacesException

Override behavior from
UIComponentBase.invokeOnComponent(javax.faces.context.FacesContext,
java.lang.String, javax.faces.component.ContextCallback) to
provide special care for positioning the data properly before finding
the component and invoking the callback on it. If the argument
clientId is equal to this.getClientId() simply invoke the
callback, passing the context argument and this as
arguments, and return true. Otherwise, attempt to extract a
rowIndex from the clientId. For example, if the argument clientId
was form:data:3:customerHeader the rowIndex would be 3. Let this
value be called newIndex. The current rowIndex of this instance must
be saved aside and restored before returning in all cases, regardless
of the outcome of the search or if any exceptions are thrown in the
process.

The implementation of this method must never return true if setting
the rowIndex of this instance to be equal to newIndex causes this
instance to return false from isRowAvailable().
Overrides:  
  invokeOnComponent in class UICOMPONENTBASE

Parameters:
  context - the FacesContext for the current request
  clientId - the client identifier of the component to be passed to
  the argument callback.
  callback - an implementation of the Callback interface.

Returns:
  true if the a component with the given clientId is found, the
  callback method was successfully invoked passing that
  component as an argument, and no Exception was thrown.
  Returns false if no component with the given clientId is found.

Throws:
  NullPointerException - if any of the arguments are null
  FacesException - if the argument Callback throws an Exception, it is
  wrapped in a FacesException and re-thrown. Also throws
  FacesException if any exception is thrown when deriving the
  rowIndex from the argument clientId.

Since:
  1.2

---

public void queueEvent(FacesEvent event)

<table>
<thead>
<tr>
<th>UICOMPONENTBASE#queueEvent</th>
<th>broadcast()</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
<td>FacesEvent</td>
</tr>
<tr>
<td>Throws</td>
<td>IllegalStateException: UIViewRoot</td>
</tr>
<tr>
<td>Throws</td>
<td>NullPointerException: event null</td>
</tr>
</tbody>
</table>

queueEvent

public void queueEvent(FacesEvent event)

Override the default
UICOMPONENTBASE.queueEvent(javax.faces.event.FacesEvent)
processing to wrap any queued events in a wrapper so that we can
reset the current row index in broadcast().

**Overrides:**
- `queueEvent` in class `UIComponentBase`

**Parameters:**
- `event` - `FacesEvent` to be queued

**Throws:**
- `IllegalStateException` - if this component is not a descendant of a `UIViewRoot`
- `NullPointerException` - if `event` is null

---

**public void broadcast(FacesEvent event) throws AbortProcessingException**

**UIComponentBase#broadcast**

**queueEvent()**

- `event` - `FacesEvent`

**Throws:**
- `AbortProcessingException`: JavaServer Face
- `IllegalArgumentException`: `FacesEvent`
- `NullPointerException`: `event null`

---

**broadcast**

**public void broadcast(FacesEvent event)**

**throws** `AbortProcessingException`

Override the default `UIComponentBase.broadcast(javax.faces.event.FacesEvent)` processing to unwrap any wrapped `FacesEvent` and reset the current row index, before the event is actually broadcast. For events that we did not wrap (in `queueEvent()`), default processing will occur.

**Overrides:**
- `broadcast` in class `UIComponentBase`

**Parameters:**
event - The FacesEvent to be broadcast

Throws:

AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed
IllegalArgumentException - if the implementation class of this FacesEvent is not supported by this component
NullPointerException - if event is null

public void encodeBegin(FacesContext context) throws java.io.IOException

context | FacesContext
Throws | java.io.IOException: /
Throws | NullPointerException: context null

encodeBegin

public void encodeBegin(FacesContext context)
throws IOException

In addition to the default behavior, ensure that any saved per-row state for our child input components is discarded unless it is needed to rerender the current page with errors.

Overrides:
encodeBegin in class UIComponentBase

Parameters:
context - FacesContext for the current request

Throws:
IOException - if an input/output error occurs while rendering
NullPointerException - if context is null
public void processDecodes(FacesContext context)

Override the default UIComponentBase.processDecodes method to perform the following steps.

- If the rendered property of this UIComponent is false, skip further processing.
- Set the current rowIndex to -1.
- Call the processDecodes() method of all facets of this UIData, in the order determined by a call to getFacets().keySet().iterator().
- Call the processDecodes() method of all facets of the UIColumn.
Iterate over the set of rows that were included when this component was rendered (i.e. those defined by the first and rows properties), performing the following processing for each row:

- Set the current rowIndex to the appropriate value for this row.
- If isRowAvailable() returns true, iterate over the children components of each UIColumn child of this UIData component, calling the processDecodes() method for each such child.
- Set the current rowIndex to -1.
- Call the decode() method of this component.
- If a RuntimeException is thrown during decode processing, call FacesContext.renderResponse() and re-throw the exception.

Override:

processDecodes in class UICOMPONENTBASE

Parameters:
context - FacesContext for the current request

Throws:
NullPointerException - if context is null

public void processValidators(FacesContext context)

UICOMPONENTBASE#processValidators
processValidators

public void processValidators(FacesContext context)

Override the default UICOMPONENTBase.processValidators(javax.faces.context.FacesContext) processing to perform the following steps.

- If the rendered property of this UICOMPONENT is false, skip further processing.
- Set the current rowIndex to -1.
- Call the processValidators() method of all facets of this UIDATA, in the order determined by a call to getFacets().keySet().iterator().
- Call the processValidators() method of all facets of the UICOLUMN children of this UIDATA.
- Iterate over the set of rows that were included when this component was rendered (i.e. those defined by the first and rows properties), performing the following processing for each row:
  - Set the current rowIndex to the appropriate value for this row.
  - If isRowAvailable() returns true, iterate over the children components of each UICOLUMN child of this UIDATA component, calling the processValidators() method for each such child.
- Set the current rowIndex to -1.

Overrides:
  processValidators in class UICOMPONENTBase

Parameters:
  context - FacesContext for the current request

Throws:
public void processUpdates(FacesContext context)

UIComponentBase#processUpdates

- **UIComponent** rendered false
- rowIndex -1
- getFacets().keySet().iterator() UIData facet processUpdates()
- **UIData** **UIColumn** facet processUpdates()
- first rows
  - rowIndex
  - isRowAvailable() true **UIData** **UIColumn** processUpdates()
- rowIndex -1

context FacesContext
Throws NullPointerException: context null

processUpdates

public void processUpdates(FacesContext context)

Override the default
UIComponentBase.processUpdates(javax.faces.context.FacesContext) processing to perform the following steps.

- If the rendered property of this **UIComponent** is false, skip further processing.
- Set the current rowIndex to -1.
- Call the processUpdates() method of all facets of this **UIData**, in the order determined by a call to getFacets().keySet().iterator().
- Call the processUpdates() method of all facets of the **UIColumn**
Iterate over the set of rows that were included when this component was rendered (i.e. those defined by the first and rows properties), performing the following processing for each row:

- Set the current rowIndex to the appropriate value for this row.
- If isRowAvailable() returns true, iterate over the children components of each UIColumn child of this UIData component, calling the processUpdates() method for each such child.
- Set the current rowIndex to -1.

Overrides:

   processUpdates in class UICOMPONENTBASE

Parameters:

   context - FacesContext for the current request

Throws:

   NullPointerException - if context IS null

```
protected DataModel getDataModel()
```

```
#setDataModel null #getValue ListDataModel
```

#### getDataModel

protected DataModel getDataModel()

Return the internal DataModel object representing the data objects that we will iterate over in this component's rendering.

If the model has been cached by a previous call to

children of this UIData.
setDataModel(javax.faces.model.DataModel), return it. Otherwise call getValue(). If the result is null, create an empty ListDataModel and return it. If the result is an instance of DataModel, return it. Otherwise, adapt the result as described in getValue() and return it.

protected void setDataModel(DataModel dataModel)

<table>
<thead>
<tr>
<th>DataModel</th>
<th>UIData</th>
<th>DataModel</th>
<th>setDataModel</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataModel</td>
<td>null</td>
<td>DataModel</td>
<td>#getDataModel</td>
</tr>
<tr>
<td>DataModel</td>
<td>DataModel</td>
<td>null</td>
<td></td>
</tr>
</tbody>
</table>

dataModel

setDataModel

protected void setDataModel(DataModel dataModel)

Set the internal DataModel. This UIData instance must use the given DataModel as its internal value representation from now until the next call to setDataModel. If the given DataModel is null, the internal DataModel must be reset in a manner so that the next call to getDataModel() causes lazy instantiation of a newly refreshed DataModel.

Subclasses might call this method if they either want to restore the internal DataModel during the Restore View phase or if they want to explicitly refresh the current DataModel for the Render Response phase.

Parameters:

dataModel - the new DataModel or null to cause the model to be refreshed.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.mail Interface UIDFolder

public interface UIDFolder

Inner classes: UIDFolder.FetchProfileItem

UIDFolder "disconnected" Folder id IMAP disconnected

(UID) long

RFC 2060 http://www.ietf.org/rfc/rfc2060.txt

The UIDFolder interface is implemented by Folders that can support the "disconnected" mode of operation, by providing unique-ids for messages in the folder. This interface is based on the IMAP model for supporting disconnected operation.

A Unique identifier (UID) is a positive long value, assigned to each message in a specific folder. Unique identifiers are assigned in a strictly ascending fashion in the mailbox. That is, as each message is added to the mailbox it is assigned a higher UID than the message(s) which were added previously. Unique identifiers persist across sessions. This permits a client to resynchronize its state from a previous session with the server.

Associated with every mailbox is a unique identifier validity value. If unique identifiers from an earlier session fail to persist to this session, the unique identifier validity value must be greater than the one used in the earlier session.

Refer to RFC 2060 http://www.ietf.org/rfc/rfc2060.txt for more information.

Author: John Mani
Nested Class Summary

| static class | UIDFolder.FetchProfileItem | A fetch profile item for fetching UIDs. |

Field Summary

| static long | LASTUID | This is a special value that can be used as the end parameter in `getMessagesByUID(start, end)`, to denote the UID of the last message in the folder. |

Method Summary

| Message | getMessageByUID(long uid) | Get the Message corresponding to the given UID. |
| Message[] | getMessagesByUID(long[] uids) | Get the Messages specified by the given array of UIDs. |
| Message[] | getMessagesByUID(long start, long end) | Get the Messages specified by the given range. |
| long | getUID(Message message) | Get the UID for the specified message. |
| long | getUIDValidity() | Returns the UIDValidity value associated with this folder. |

Field Detail

LASTUID

static final long LASTUID

This is a special value that can be used as the end parameter in
getMessagesByUID(start, end), to denote the UID of the last message in the folder.

See Also:
- getMessagesByUID(long, long), Constant Field Values

### Method Detail

```java
public long getUIDValidity() throws MessagingException
```

**UIDValidity**

```java
UIDValidity
```

```java
return
```

```java
UIDValidity
```

### getUIDValidity

```java
long getUIDValidity() throws MessagingException
```

Returns the UIDValidity value associated with this folder.

Clients typically compare this value against a UIDValidity value saved from a previous session to insure that any cached UIDs are not stale.

**Returns:**
- UIDValidity

**Throws:**
- MessagingException

```java
public Message getMessageByUID(long uid) throws MessagingException
```

**UID**

```java
Message
```

```java
uid
```

**UID**

```java
null
```
getMessageByUID

getMessageByUID(Message getMessagesByUID(long uid)
                     throws MessagingException

  Get the Message corresponding to the given UID. If no such message exists, null is returned.

  Parameters:
  uid - UID for the desired message

  Returns:
  the Message object. null is returned if no message corresponding to this UID is obtained.

  Throws:
  MessagingException

public Message[] getMessagesByUID(long start, long end)
                     throws MessagingException

  Get the Messages specified by the given range. The special value LASTUID can be used for the end parameter to indicate the UID of
the last message in the folder.

**Parameters:**
- start - start UID
- end - end UID

**Returns:**
- array of Message objects

**Throws:**
- MessagingException

**See Also:**
- LASTUID

---

```java
public Message[] getMessagesByUID(long[] uids) throws MessagingException
```

```java
UID    Message UID
null   UID
```

**getMessagesByUID**

Get the Messages specified by the given array of UIDs. If any UID is invalid, null is returned for that entry.

Note that the returned array will be of the same size as the specified array of UIDs, and null entries may be present in the array to indicate invalid UIDs.

**Parameters:**
- uids - array of UIDs

**Returns:**
public long getUID(Message message) throws MessagingException

Get the UID for the specified message. Note that the message must belong to this folder. Otherwise java.util.NoSuchElementException is thrown.

Parameters:
message - Message from this folder

Returns:
UID for this message

Throws:
java.util.NoSuchElementException - if the given Message is not in this Folder.
MessagingException
PS:
javax.mail  **Class UIDFolder.FetchProfileItem**

**java.lang.Object**
  ↓  **javax.mail.FetchProfile.Item**
    ↓  **javax.mail.UIDFolder.FetchProfileItem**

Enclosing interface:
  **UIDFolder**

```java
class UIDFolder.FetchProfileItem extends FetchProfile.Item
```

**Enclosing interface:**  **UIDFolder**

**public static class**  **UIDFolder.FetchProfileItem**

**Extends:**  **FetchProfile.Item**

**Contained within:**  **UIDFolder**

**UID fetch**

**FetchProfile Item**

**UID**

**See also**  **javax.mail.FetchProfile**

A fetch profile item for fetching UIDs. This inner class extends the FetchProfile.Item class to add new FetchProfile item types, specific to UIDFolders. The only item currently defined here is the **UID** item.

See Also:
  **FetchProfile**

---

**Field Summary**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <strong>UIDFolder.FetchProfileItem</strong></td>
<td><strong>UID</strong> is a fetch profile item that can be included in a FetchProfile during a fetch request to a Folder.</td>
</tr>
</tbody>
</table>

**Fields inherited from class javax.mail.FetchProfile.Item**
UID

public static final UIDFolder.FetchProfileItem UID

UID is a fetch profile item that can be included in a FetchProfile during a fetch request to a Folder. This item indicates that the UIDs for messages in the specified range are desired to be prefetched.

An example of how a client uses this is below:

```java
FetchProfile fp = new FetchProfile();
fp.add(UIDFolder.FetchProfileItem.UID);
folder.fetch(msgs, fp);
```
protected UIDFolder.FetchProfileItem(String name)

UIDFolder.FetchProfileItem

protected UIDFolder.FetchProfileItem(String name)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.component Class UIForm

java.lang.Object  
   javax.faces.component.UIComponent  
      javax.faces.component.UIComponentBase  
         javax.faces.component.UIForm

All Implemented Interfaces:
   NamingContainer, StateHolder

Direct Known Subclasses:
   HtmlForm

public class UIForm
extends UIComponentBase
implements NamingContainer

Extends: UIComponent > UIComponentBase
Implements: NamingContainer
Extended by: HtmlForm

UIForm UIComponent

   rendererType "javax.faces.Form"   setRendererType()

UIForm is a UIComponent that represents an input form to be presented to the user, and whose child components represent (among other things) the input fields to be included when the form is submitted.

By default, the rendererType property must be set to "javax.faces.Form". This value can be changed by calling the setRendererType() method.

Field Summary
### Constructor Summary

**UIForm()**  
Create a new `UIForm` instance with default property values.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getContainerClientId(FacesContext context)</code></td>
<td>Override the <code>UIComponent.getContainerClientId(javax.faces.context.FacesContext)</code> to allow users to disable this form from prepending its <code>clientId</code> to descendant's <code>clientIds</code> depending on the value of this form's <code>isPrependId()</code> property.</td>
</tr>
<tr>
<td><code>getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>isPrependId()</code></td>
<td>Returns the current value of the <code>submitted</code> property.</td>
</tr>
<tr>
<td><code>processDecodes(FacesContext context)</code></td>
<td>Override <code>UIComponent.processDecodes(javax.faces.context.FacesContext)</code> to ensure that the form is decoded <em>before</em> its children.</td>
</tr>
</tbody>
</table>
void processUpdates(FacesContext context)
  Override

    UIComponent.processUpdates(javax.faces.context.FacesContext) to ensure that the children of this UIForm instance are only processed if isSubmitted() returns true.

void processValidators(FacesContext context)
  Override

    UIComponent.processValidators(javax.faces.context.FacesContext) to ensure that the children of this UIForm instance are only processed if isSubmitted() returns true.

void setPrependId(boolean prependId)

void setSubmitted(boolean submitted)

  If this UIForm instance (as opposed to other forms in the page) is experiencing a submit during this request processing lifecycle, this method must be called, with true as the argument, during the UIComponent.decode(javax.faces.context.FacesContext) for this UIForm instance.

Methods inherited from class javax.faces.component.UIComponentBase
addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildren, getClientId, getFacesContext, getFacets, getFacet, getFacetCount, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRenderersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, restoreState, saveAttachedState, saveState, setId, setParent, setRendered, setRendererType, setId, setParent
### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**
- [Constant Field Values](#)

---

**COMPONENT_FAMILY**

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**
- [Constant Field Values](#)

### Constructor Detail

```java
public UIForm()
```

- **UIForm**
public UIForm()

    Create a new UIForm instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

    Description copied from class: UIComponent

    Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

    Specified by:
        getFamily in class UIComponent

public boolean isSubmitted()

    submitted false #setSubmitted

isSubmitted

public boolean isSubmitted()

    Returns the current value of the submitted property. The default value is false. See setSubmitted(boolean) for details.
public void setSubmitted(boolean submitted)

If this UIForm instance (as opposed to other forms in the page) is experiencing a submit during this request processing lifecycle, this method must be called, with true as the argument, during the UIComponent.decode(javax.faces.context.FacesContext) for this UIForm instance. If this UIForm instance is not experiencing a submit, this method must be called, with false as the argument, during the UIComponent.decode(javax.faces.context.FacesContext) for this UIForm instance.

The value of a UIForm's submitted property must not be saved as part of its state.

public boolean isPrependId()

public void setPrependId(boolean prependId)
setPrependId

public void setPrependId(boolean prependId)

public void processDecodes(FacesContext context)

Throws

NullPointerException: NullPointerException
NullPointerException context null

processDecodes

public void processDecodes(FacesContext context)

Override

UIComponent.processDecodes(javax.faces.context.FacesContext) to ensure that the form is decoded before its children. This is necessary to allow the submitted property to be correctly set.

Overrides:

processDecodes in class UIComponentBase

Parameters:

context - FacesContext for the request we are processing

Throws:

NullPointerException - if context is null

public void processValidators(FacesContext context)

throws

NullPointerException: NullPointerException
NullPointerException context null
processValidators

public void processValidators(FacesContext context)

Override
UIComponent.processValidators(javax.faces.context.FacesContext)
to ensure that the children of this UIForm instance are only processed if isSubmitted() returns true.

Overrides:  
processValidators in class UICOMPONENTBASE

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null

processUpdates

public void processUpdates(FacesContext context)

Override
UIComponent#processUpdates #isSubmitted true
UIForm

Throws
NullPointerException: NullPointerException
NullPointerException context null

processUpdates

public void processUpdates(FacesContext context)

Override
UIComponent.processUpdates(javax.faces.context.FacesContext) to ensure that the children of this UIForm instance are only processed if isSubmitted() returns true.

Overrides:  
processUpdates in class UICOMPONENTBASE

Parameters:
context - FacesContext for the request we are processing
Throws:

`NullPointerException` - if context is null

```java
public String getContainerClientId(FacesContext context)
```

**UIComponent#getContainerClientId**  **#isPrependId**

`clientId`  `clientIds`

**getContainerClientId**

```java
public String getContainerClientId(FacesContext context)
```

Override the `UIComponent.getContainerClientId(javax.faces.context.FacesContext)` to allow users to disable this form from prepending its `clientId` to its descendent's `clientIds` depending on the value of this form's `isPrependId()` property.

**Overrides:**

`getContainerClientId` in class `UIComponent`
javax.faces.component Class UIGraphic

java.lang.Object
    ▼ javax.faces.component.UIComponent
        ▼ javax.faces.component.UIComponentBase
            ▼ javax.faces.component.UIGraphic

All Implemented Interfaces:
    StateHolder

Direct Known Subclasses:
    HtmlGraphicImage

public class UIGraphic
extends UIComponentBase

Extends: UIComponent > UIComponentBase
Extended by: HtmlGraphicImage

UIGraphic  UIComponent

    rendererType "javax.faces.Image"  setRendererType()

UIGraphic is a UIComponent that displays a graphical image to the user. The user cannot manipulate this component; it is for display purposes only.

By default, the rendererType property must be set to "javax.faces.Image". This value can be changed by calling the setRendererType() method.

---

Field Summary

<table>
<thead>
<tr>
<th>static</th>
<th>String</th>
<th>COMPONENT_FAMILY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The standard component family for this component.

<table>
<thead>
<tr>
<th>static String COMPONENT_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIComponent

<table>
<thead>
<tr>
<th>bindings</th>
</tr>
</thead>
</table>

## Constructor Summary

**UIGraphic()**

Create a new UIGraphic instance with default property values.

## Method Summary

<table>
<thead>
<tr>
<th>Fields</th>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getFamily()</td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td>String</td>
<td>getUrl()</td>
<td>Return the image URL for this UIGraphic.</td>
</tr>
<tr>
<td>Object</td>
<td>getValue()</td>
<td>Returns the value property of the UIGraphic.</td>
</tr>
<tr>
<td>ValueBinding</td>
<td>getValueBinding(String name)</td>
<td><strong>Deprecated.</strong> This has been replaced by getValueExpression(java.lang.String).**</td>
</tr>
<tr>
<td>ValueExpression</td>
<td>getValueExpression(String name)</td>
<td>Return any ValueExpression set for value if a ValueExpression for url is requested; otherwise, perform the default superclass processing for this method.</td>
</tr>
<tr>
<td>void</td>
<td>restoreState(FacesContext context, Object state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td>saveState(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td>setUrl(String url)</td>
<td></td>
</tr>
</tbody>
</table>
### Set the image URL for this `UIGraphic`.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong><code>setValue(Object value)</code></strong></td>
<td>Sets the value property of the <code>UIGraphic</code>.</td>
</tr>
<tr>
<td><strong><code>setValueBinding(String name, ValueBinding binding)</code></strong></td>
<td>Deprecated. This has been replaced by <code>setValueExpression(java.lang.String, javax.el.ValueExpression)</code>.</td>
</tr>
<tr>
<td><strong><code>setValueExpression(String name, ValueExpression binding)</code></strong></td>
<td>Store any <code>ValueExpression</code> specified for <code>url</code> under <code>value</code> instead; otherwise, perform the default superclass processing for this method.</td>
</tr>
</tbody>
</table>

**Methods inherited from class `javax.faces.component.UIComponentBase`**

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`

**Methods inherited from class `javax.faces.component.UIComponent`**

- `encodeAll`, `getContainerClientId`

**Methods inherited from class `java.lang.Object`**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
COMPONENT_TYPE

public static final String COMPONENT_TYPE

   The standard component type for this component.

See Also:
   Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

   The standard component family for this component.

See Also:
   Constant Field Values

### Constructor Detail

public UIGraphic()

   UIGraphic

UIGraphic

public UIGraphic()

   Create a new UIGraphic instance with default property values.

### Method Detail
public String getFamily()

getFamily

public String getFamily()

Description copied from class: UICOMPONENT

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:
   getFamily in class UICOMPONENT

public String getUrl()

getValue()

getUrl()

public String getUrl()

Return the image URL for this UIGraphic. This method is a typesafe alias for getValue().

public void setUrl(String url)

setValue()
**setUrl**

```java
public void setUrl(String url)
```

Set the image URL for this `UIImage`. This method is a typesafe alias for `setValue()`.

**Parameters:**
- `url` - The new image URL

---

**public Object getValue()**

```java
public Object getValue()
```

Returns the `value` property of the `UIImage`. This will typically be rendered as an URL.

---

**public void setValue(Object value)**

```java
public void setValue(Object value)
```

Sets the `value` property of the `UIImage`. This will typically be rendered as an URL.
public **ValueBinding** getValueBinding(String name)

url   ValueBinding   value   ValueBinding

name   ValueBinding

Declares name - Name of the attribute or property for which to retrieve a **ValueBinding**

Throws:  
**NullPointerException** - if name is null

**Deprecated.** *This has been replaced by getValueExpression(java.lang.String).*

Return any **ValueBinding** set for value if a **ValueBinding** for url is requested; otherwise, perform the default superclass processing for this method.

Override:getValueBinding in class UICOMPONENTBase

Parameters:

Throws:

**NumberFormatException** - if name is null

public void setValueBinding(String name, **ValueBinding** binding)

value   url   ValueBinding   ValueBinding

ValueExpression
setValueBinding

```java
public void setValueBinding(String name, ValueBinding binding)
```

**Deprecated.** *This has been replaced by*

```java
setValueExpression(java.lang.String, javax.el.ValueExpression)
```

Store any `ValueBinding` specified for `url` under `value` instead; otherwise, perform the default superclass processing for this method. In all cases, the superclass is relied on to convert the `ValueBinding` to a `ValueExpression`.

**Overrides:**

`setValueBinding` in class `UIComponentBase`

**Parameters:**

- `name` - Name of the attribute or property for which to set a `ValueBinding`
- `binding` - The `ValueBinding` to set, or `null` to remove any currently set `ValueBinding`

**Throws:**

- `NullPointerException` - if `name` is `null`

------------------------

```java
public ValueExpression getValueExpression(String name)
```

```java
url ValueExpression value ValueExpression
```

**Name**

```java
ValueExpression
```

**Throws**

`NullPointerException: name null`

**since**

1.2
getValueExpression

public ValueExpression getValueExpression(String name)

Return any ValueExpression set for value if a ValueExpression for url is requested; otherwise, perform the default superclass processing for this method.

Overrides:
getValueExpression in class UIComponent

Parameters:
name - Name of the attribute or property for which to retrieve a ValueExpression

Throws:
NullPointerException - if name is null

Since:
1.2

public void setValueExpression(String name, ValueExpression binding)

value url ValueExpression

name ValueExpression
binding ValueExpression null ValueExpression

Throws
NullPointerException - name null

since
1.2

setValueExpression

public void setValueExpression(String name, ValueExpression binding)

Store any ValueExpression specified for url under value instead; otherwise, perform the default superclass processing for this method.
Overrides:

```
setValueExpression in class UIComponent
```

Parameters:

```
name - Name of the attribute or property for which to set a ValueExpression
binding - The ValueExpression to set, or null to remove any currently set ValueExpression
```

Throws:

```
NullPointerException - if name is null
```

Since:

```
1.2
```

```java
public Object saveState(FacesContext context)
```

**saveState**

```java
public Object saveState(FacesContext context)
```

**Description copied from interface:** StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UICOMPONENT with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the `StateManager`

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`
public void restoreState(FacesContext context, Object state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by: restoreState in interface StateHolder
Overrides: restoreState in class UIComponentBase
PS:
javax.faces.component  Class UIInput

java.lang.Object
  └─javax.faces.component.UIComponent
      └─javax.faces.component.UIComponentBase
          └─javax.faces.component.UIOutput
              └─javax.faces.component.UIInput

All Implemented Interfaces:
   EditableValueHolder, StateHolder, ValueHolder

Direct Known Subclasses:
   HtmlInputHidden, HtmlInputSecret, HtmlInputText,
   HtmlInputTextarea, UISelectBoolean, UISelectMany, UISelectOne

public class UIInput
extends UIOutput
implements EditableValueHolder

Extends: UIComponent > UIComponentBase > UIOutput
Implements: EditableValueHolder
Extended by: HtmlInputHidden, HtmlInputSecret, HtmlInputText,
            HtmlInputTextarea, UISelectBoolean, UISelectMany, UISelectOne

UIInput  UIComponent  UIOutput
javax.faces.render.Renderer

    setSubmittedValue() String  null
    setValue()        immediate  true

UIOutput

UIInput  validate()  ValueChangeEvent
broadcast()  immediate  true

rendererType "Text"  setRendererType()
**UIInput** is a **UIComponent** that represents a component that both displays output to the user (like **UIOutput** components do) and processes request parameters on the subsequent request that need to be decoded. There are no restrictions on the data type of the local value, or the object referenced by the value binding expression (if any); however, individual **Renderer**s will generally impose restrictions on the type of data they know how to display.

During the **Apply Request Values** phase of the request processing lifecycle, the decoded value of this component, usually but not necessarily a String, must be stored - but not yet converted - using `setSubmittedValue()`. If the component wishes to indicate that no particular value was submitted, it can either do nothing, or set the submitted value to `null`.

By default, during the **Process Validators** phase of the request processing lifecycle, the submitted value will be converted to a typesafe object, and, if validation succeeds, stored as a local value using `setValue()`. However, if the `immediate` property is set to `true`, this processing will occur instead at the end of the **Apply Request Values** phase.

During the **Render Response** phase of the request processing lifecycle, conversion for output occurs as for **UIOutput**.

When the `validate()` method of this **UIInput** detects that a value change has actually occurred, and that all validations have been successfully passed, it will queue a **ValueChangeEvent**. Later on, the `broadcast()` method will ensure that this event is broadcast to all interested listeners. This event will be delivered by default in the **Process Validators** phase, but can be delivered instead during **Apply Request Values** if the `immediate` property is set to `true`.

By default, the `rendererType` property must be set to "Text". This value can be changed by calling the `setRendererType()` method.
**Field Summary**

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_FAMILY</td>
<td>The standard component family for this component.</td>
</tr>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
<tr>
<td>static String</td>
<td>CONVERSION_MESSAGE_ID</td>
<td>The message identifier of the FacesMessage to be created if a conversion error occurs, and neither the page author nor the ConverterException provides a message.</td>
</tr>
<tr>
<td>static String</td>
<td>REQUIRED_MESSAGE_ID</td>
<td>The message identifier of the FacesMessage to be created if a required check fails.</td>
</tr>
<tr>
<td>static String</td>
<td>UPDATE_MESSAGE_ID</td>
<td>The message identifier of the FacesMessage to be created if a model update error occurs, and the thrown exception has no message.</td>
</tr>
</tbody>
</table>

**Fields inherited from class javax.faces.component.UIComponent**

**bindings**

**Constructor Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIInput()</td>
<td>Create a new UIInput instance with default property values.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void addValidator(Validator validator)</td>
<td>Add a Validator instance to the set associated with this component.</td>
</tr>
<tr>
<td>void setValueChangeListener(ValueChangeListener listener)</td>
<td>Add a new ValueChangeListener to the set of listeners being notified when ValueChangeEvent occurs.</td>
</tr>
<tr>
<td>protected boolean compareValues(Object previous, Object value)</td>
<td>Return true if the new value is different from the previous value.</td>
</tr>
<tr>
<td>void decode(FacesContext context)</td>
<td>Decode the component using the provided FacesContext.</td>
</tr>
</tbody>
</table>
void Decode any new state of this UIComponent from the specified FacesContext, and store this state as needed.

protected Object getConvertedValue(FacesContext context, Object newSubmittedValue)
Convert the submitted value into a "local value" of the type, if necessary.

String getConverterMessage()
If there has been a call to setConverterMessage(), this instance, return the message.

String getFamily()
Return the identifier of the component family to which this belongs.

String getRequiredMessage()
If there has been a call to setRequiredMessage(), this instance, return the message.

Object getSubmittedValue()
Return the submittedValue value of this UIInput component.

MethodBinding getValidator()
Deprecated. getValidators() should be used instead.

String getValidatorMessage()
If there has been a call to setRequiredMessage(), this instance, return the message.

Validator[] getValidators()
Return the set of registered Validators for this UIInput component.

MethodBinding getValueChangeListener()
If EditableValueHolder.setValueChangeListener() was not previously called for this instance, this method

ValueChangeListener[] getValueChangeListeners()
Return the set of registered ValueChangeListener instance.

boolean isImmediate()
Return the "immediate" state for this component.

boolean isLocalValueSet()
Return the "local value set" state for this component.

boolean isRequired()
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>isValid()</code></td>
<td>Return a flag indicating whether the local value is valid (no conversion error has occurred).</td>
</tr>
<tr>
<td><code>processDecodes(FacesContext context)</code></td>
<td>Specialized decode behavior on top of that provided by the superclass.</td>
</tr>
<tr>
<td><code>processUpdates(FacesContext context)</code></td>
<td>In addition to the standard <code>processUpdates</code> behavior of <code>UIComponentBase</code>, calls <code>updateModel()</code>.</td>
</tr>
<tr>
<td><code>processValidators(FacesContext context)</code></td>
<td>In addition to the standard <code>processValidators</code> behavior of <code>UIComponentBase</code>, calls <code>validate()</code> if the <code>immediate</code> property is false (the default); if the component is invalid afterwards, calls <code>FacesContext.renderResponse()</code>.</td>
</tr>
<tr>
<td><code>removeValidator(Validator validator)</code></td>
<td>Remove a <code>Validator</code> instance from the set associated with the component. If it was previously associated.</td>
</tr>
<tr>
<td><code>removeValueChangeListener(ValueChangeListener listener)</code></td>
<td>Remove an existing <code>ValueChangeListener</code> (if any) from the listeners interested in being notified when <code>ValueChangeEvent</code> fired.</td>
</tr>
<tr>
<td><code>resetValue()</code></td>
<td>Convenience method to reset this component's value to the initialized state.</td>
</tr>
<tr>
<td><code>restoreState(FacesContext context, Object state)</code></td>
<td>Perform any processing required to restore the state from the state Object.</td>
</tr>
<tr>
<td><code>saveState(FacesContext context)</code></td>
<td>Gets the state of the instance as a <code>Serializable</code>.</td>
</tr>
<tr>
<td><code>setConverterMessage(String message)</code></td>
<td>Override any <code>ValueExpression</code> set for the &quot;converterMessage&quot; literal argument provided to this method.</td>
</tr>
<tr>
<td><code>setImmediate(boolean immediate)</code></td>
<td>Set the &quot;immediate&quot; state for this component.</td>
</tr>
<tr>
<td><code>setLocalValueSet(boolean localValueSet)</code></td>
<td>Sets the &quot;local value set&quot; state for this component.</td>
</tr>
</tbody>
</table>
### Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void <code>setRequired</code> (boolean required)</td>
<td>Set the &quot;required field&quot; state for this component.</td>
</tr>
<tr>
<td>void <code>setRequiredMessage</code> (String message)</td>
<td>Override any <code>ValueExpression</code> set for the &quot;requiredMessage&quot; literal argument provided to this method.</td>
</tr>
<tr>
<td>void <code>setSubmittedValue</code> (Object submittedValue)</td>
<td>Set the submittedValue value of this <code>UIInput</code> component.</td>
</tr>
<tr>
<td>void <code>setValid</code> (boolean valid)</td>
<td>Set a flag indicating whether the local value of this <code>UIInput</code> component is valid (no conversion error has occurred).</td>
</tr>
<tr>
<td>void <code>setValidator</code> (MethodBinding validatorBinding)</td>
<td>Deprecated. Use <code>addValidator</code> instead, obtaining the argument <code>Validator</code> by creating <code>MethodExpressionValidator</code>.</td>
</tr>
<tr>
<td>void <code>setValidatorMessage</code> (String message)</td>
<td>Override any <code>ValueExpression</code> set for the &quot;validatorMessage&quot; literal argument provided to this method.</td>
</tr>
<tr>
<td>void <code>setValue</code> (Object value)</td>
<td>Set the value of this <code>UIComponent</code> (if any).</td>
</tr>
<tr>
<td>void <code>setValueChangeListener</code> (MethodBinding valueChangeListener)</td>
<td>Deprecated. Use <code>addValueChangeListener</code> obtaining the argument <code>ValueChangeListener</code> by creating <code>MethodExpressionValueChangeListener</code>.</td>
</tr>
<tr>
<td>void <code>updateModel</code> (FacesContext context)</td>
<td>Perform the following algorithm to update the model with this <code>UIInput</code>, if any, as appropriate.</td>
</tr>
<tr>
<td>void <code>validate</code> (FacesContext context)</td>
<td>Perform the following algorithm to validate the local value of this <code>UIInput</code>.</td>
</tr>
<tr>
<td>protected void <code>validateValue</code> (FacesContext context, Object newValue)</td>
<td>Set the &quot;valid&quot; property according to the below algorithm.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.component.UIOutput`:

- `getConverter`, `getLocalValue`, `getValue`, `setConverter`
Methods inherited from class javax.faces.component.UICOMPONENTBASE

- addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

Methods inherited from class javax.faces.component. UICOMPONENT

- encodeAll, getContainerClientId, getValueExpression, setValueExpression

Methods inherited from class java.lang. OBJECT

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Methods inherited from interface javax.faces.component. VALUEHOLDER

- getConverter, getLocalValue, getValue, setConverter

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:

Constant Field Values
COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

CONVERSION_MESSAGE_ID

public static final String CONVERSION_MESSAGE_ID

The message identifier of the FacesMessage to be created if a conversion error occurs, and neither the page author nor the ConverterException provides a message.

See Also:
Constant Field Values

REQUIRED_MESSAGE_ID

public static final String REQUIRED_MESSAGE_ID

The message identifier of the FacesMessage to be created if a required check fails.

See Also:
Constant Field Values
**UPDATE_MESSAGE_ID**

public static final String UPDATE_MESSAGE_ID

The message identifier of the FacesMessage to be created if a model update error occurs, and the thrown exception has no message.

See Also:
Constant Field Values

---

**Constructor Detail**

public UIInput()

UIInput

UIInput

public UIInput()

Create a new UIInput instance with default property values.

---

**Method Detail**

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UICOMPONENT
Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Overrides:

\texttt{getFamily} in class \texttt{UIOutput}

---

\textbf{public Object getSubmittedValue()}

\underline{UIInput} \texttt{submittedValue} \hspace{1cm} \underline{Renderer} \texttt{decode()}

\texttt{validate()}

---

\textbf{getSubmittedValue}

\textbf{public Object getSubmittedValue()}

Return the submittedValue value of this \texttt{UIInput} component. This method should only be used by the \texttt{decode()} and \texttt{validate()} method of this component, or its corresponding \texttt{Renderer}.

\textbf{Specified by:}

\texttt{getSubmittedValue} in interface \texttt{EditableValueHolder}

---

\textbf{public void setSubmittedValue(Object submittedValue)}

\underline{UIInput} \texttt{submittedValue} \hspace{1cm} \underline{Renderer} \texttt{decode()}

\texttt{validate()}

\textit{submittedValue}

---

\textbf{setSubmittedValue}
public void `setSubmittedValue`(`Object` submittedValue)

Set the submittedValue value of this `UIInput` component. This method should only be used by the `decode()` and `validate()` method of this component, or its corresponding `Renderer`.

**Specified by:**  
`setSubmittedValue` in interface `EditableValueHolder`  
**Parameters:**  
submittedValue - The new submitted value

---

public void `setValue`(`Object` value)

**setValue**

public void `setValue`(`Object` value)

Description copied from interface: `ValueHolder`

Set the value of this `UIComponent` (if any).

**Specified by:**  
`setValue` in interface `ValueHolder`  
**Overrides:**  
`setValue` in class `UIOutput`  
**Parameters:**  
value - The new local value

---

public void `resetValue`()

#`setValue` null

#`setSubmittedValue` null
resetValue

public void resetValue()

    Convenience method to reset this component's value to the uninitialized state. This method does the following:

    Call setValue(java.lang.Object) passing null.
    Call setSubmittedValue(java.lang.Object) passing null.
    Call setLocalValueSet(boolean) passing false.
    Call setValid(boolean) passing true.

    Upon return from this call if the instance had a ValueBinding associated with it for the "value" property, this binding is evaluated when UIOutput.getValue() is called. Otherwise, null is returned from getValue().

public boolean isLocalValueSet()
"local value set" setValue() true

isLocalValueSet

public boolean isLocalValueSet()
Return the "local value set" state for this component. Calls to setValue() automatically reset this property to true.

**Specified by:**
`isLocalValueSet` in interface `EditableValueHolder`

```java
public void setLocalValueSet(boolean localValueSet)

"local value set"
```

**setLocalValueSet**

```java
public void setLocalValueSet(boolean localValueSet)

Sets the "local value set" state for this component.

**Specified by:**
`setLocalValueSet` in interface `EditableValueHolder`
```

```java
public boolean isRequired()

"required field"
```

**isRequired**

```java
public boolean isRequired()

Return the "required field" state for this component.

**Specified by:**
`isRequired` in interface `EditableValueHolder`
```

```java
public String getRequiredMessage()
```
getRequiredMessage

public String getRequiredMessage()

If there has been a call to setRequiredMessage(java.lang.String) on this instance, return the message. Otherwise, call UIComponent.getValueExpression(java.lang.String) passing the key "requiredMessage", get the result of the expression, and return it. Any ELExceptions thrown during the call to getValue() must be wrapped in a FacesException and rethrown.

setRequiredMessage

public void setRequiredMessage(String message)

Override any ValueExpression set for the "requiredMessage" with the literal argument provided to this method. Subsequent calls to getRequiredMessage() will return this value;

Parameters:
message - the literal message value to be displayed in the event the user hasn't supplied a value and one is required.
public String getConverterMessage()

    #setConverterMessage #getValueExpression
    "converterMessage" getVa] FacesException

getConverterMessage

public String getConverterMessage()

    If there has been a call to setConverterMessage(java.lang.String) on this instance, return the message. Otherwise, call UIComponent.getValueExpression(java.lang.String) passing the key "converterMessage", get the result of the expression, and return it. Any ELExceptions thrown during the call to getValue() must be wrapped in a FacesException and rethrown.

public void setConverterMessage(String message)

    "converterMessage" ValueExpression #getConverterMessage

    message

setConverterMessage

public void setConverterMessage(String message)

    Override any ValueExpression set for the "converterMessage" with the literal argument provided to this method. Subsequent calls to getConverterMessage() will return this value;

    Parameters:
        message - the literal message value to be displayed in the event
public String getValidatorMessage()

    #setRequiredMessage  #getValueExpression
"requiredMessage"  getValue()
FacesException

getValidatorMessage

public String getValidatorMessage()

    If there has been a call to setRequiredMessage(java.lang.String) on this instance, return the message. Otherwise, call UICOMPONENT.getVALUEEXPRESSION(java.lang.String) passing the key "requiredMessage", get the result of the expression, and return it. Any ELException s thrown during the call to getValue() must be wrapped in a FacesException and rethrown.

public void setValidatorMessage(String message)

    "validatorMessage"  valueExpression
#getValueExpression
message

setValidatorMessage

public void setValidatorMessage(String message)

    Override any ValueExpression set for the "validatorMessage" with the literal argument provided to this method. Subsequent calls to
getValidatorMessage() will return this value;

**Parameters:**
message - the literal message value to be displayed in the event validation fails.

```
public boolean isValid()
```

**isValid**

```
public boolean isValid()
```

**Description copied from interface:** EditableValueHolder

Return a flag indicating whether the local value of this component is valid (no conversion error has occurred).

**Specified by:**
isValid in interface EditableValueHolder

```
public void setValid(boolean valid)
```

**setValid**

```
public void setValid(boolean valid)
```

**Description copied from interface:** EditableValueHolder

Set a flag indicating whether the local value of this component is valid (no conversion error has occurred).

**Specified by:**
setValid in interface EditableValueHolder

**Parameters:**
valid - The new valid flag
public void setRequired(boolean required)

"required field"

required "required field"

setRequired

class

class

public void setRequired(boolean required)

Set the "required field" state for this component.

Specified by:

setRequired in interface EditableValueHolder

Parameters:

required - The new "required field" state

Description copied from interface: EditableValueHolder

Return the "immediate" state for this component.

Specified by:

isImmediate in interface EditableValueHolder

isImmediate

public boolean isImmediate()

public void setImmediate(boolean immediate)
**setImmediate**

```java
public void setImmediate(boolean immediate)
```

**Description copied from interface:** [EditableValueHolder](#)

Set the "immediate" state for this component. When set to true, the component's value will be converted and validated immediately in the Apply Request Values phase, and [ValueChangeEvent](#)s will be delivered in that phase as well. The default value for this property must be false.

**Specified by:**
- [setImmediate](#) in interface [EditableValueHolder](#)

**Parameters:**
- immediate - The new "immediate" state

---

**public MethodBinding getValidator()**

```java
MethodBinding getValidator()
```

**MethodBinding**

```java
deprecated
```

**getValidator**

```java
public MethodBinding getValidator()
```

**Deprecated. getValidators() should be used instead.**

Return a MethodBinding pointing at a method that will be called during Process Validations phase of the request processing lifecycle, to validate the current value of this component.

**Specified by:**
- [getValidator](#) in interface [EditableValueHolder](#)
public void setValidator(MethodBinding validatorBinding)

MethodBinding

void FacesContext UIComponent Object

validatorBinding MethodBinding

deprecated javax.faces.validator.MethodExpressionValidator

Validator

setValidator

public void setValidator(MethodBinding validatorBinding)

Deprecated. Use addValidator(javax.faces.validator.Validator) instead, obtaining the argument Validator by creating an instance of MethodExpressionValidator.

Set a MethodBinding pointing at a method that will be called during Process Validations phase of the request processing lifecycle, to validate the current value of this component.

Any method referenced by such an expression must be public, with a return type of void, and accept parameters of type FacesContext, UIComponent, and Object.

Specified by:
setValidator in interface EditableValueHolder

Parameters:
validatorBinding - The new MethodBinding instance

public MethodBinding getValueChangeListener()

g getValueChangeListe
public MethodBinding getValueChangeListener()

Description copied from interface: EditableValueHolder

If EditableValueHolder.setValueChangeListener(javax.faces.el.MethodBinding) was not previously called for this instance, this method must return null. If it was called, this method must return the exact MethodBinding instance that was passed to EditableValueHolder.setValueChangeListener(javax.faces.el.MethodBinding)

Specified by:
   getValueChangeListener in interface EditableValueHolder

public void setValueChangeListener(MethodBinding valueChangeListener)

NullPointerException

ValueChangeListener valueChangeMethod

#getValueChangeListeners setValueChangeListener

immediate

void javax.faces.event.ValueChangeEvent

deprecated javax.faces.event.MethodExpressionValueChangeListener

setValueChangeListener

public void setValueChangeListener(MethodBinding valueChangeListener)

Deprecated. Use addValueChangeListener(javax.faces.event.ValueChangeListener) instead, obtaining the argument ValueChangeListener by creating an instance of MethodExpressionValueChangeListener.
Wrap the argument valueChangeMethod in an implementation of ValueChangeListener and store it in the internal data structure that backs the EditableValueHolder.getValueChangeListeners() method, taking care to over-write any instance that was stored by a previous call to setValueChangeListener.

This argument method will be called during the Process Validations or Apply Request Values phases (depending on the value of the immediate property).

Any method referenced by such an expression must be public, with a return type of void, and accept a parameter of type ValueChangeEvent.

Specified by:
setValueChangeListener in interface EditableValueHolder

Parameters:
valueChangeListener - The new method binding instance

```
public void processDecodes(FacesContext context)

setErrorComponentProperty(UIComponentBase component, String property, String newValue, UIComponent originalComponent)

processDecodes(FacesContext context)

UIComponentBase processDecodes immediate
true validate() RuntimeException
FacesContext#renderResponse

throws

NullPointerException: NullPointerException

processDecodes

public void processDecodes(FacesContext context)

Specialized decode behavior on top of that provided by the superclass. In addition to the standard processDecodes behavior inherited from UIComponentBase, calls validate() if the immediate property is true; if the component is invalid afterwards or a
 RuntimeException is thrown, calls FacesContext.renderResponse().

Overrides: processDecodes in class UIComponentBase

Parameters: context - FacesContext for the request we are processing

Throws: NullPointerException - if context is null

public void processValidators(FacesContext context)

 In addition to the standard processValidators behavior inherited from UIComponentBase, calls validate() if the immediate property is false (which is the default); if the component is invalid afterwards, calls FacesContext.renderResponse(). If a RuntimeException is thrown during validation processing, calls FacesContext.renderResponse() and re-throw the exception.

Overrides: processValidators in class UIComponentBase

Parameters: context - FacesContext for the request we are processing

Throws: NullPointerException - if context is null
public void processUpdates(FacesContext context)

    public void processUpdates(FacesContext context)

    In addition to the standard processUpdates behavior inherited from UIComponentBase, calls updateModel(). If the component is invalid afterwards, calls FacesContext.renderResponse(). If a RuntimeException is thrown during update processing, calls FacesContext.renderResponse() and re-throw the exception.

    Overrides:
        processUpdates in class UIComponentBase

    Parameters:
        context - FacesContext for the request we are processing

    Throws:
        NullPointerException - if context is null

public void decode(FacesContext context)

    public void decode(FacesContext context)
Description copied from class: `UIComponent`

Decode any new state of this `UIComponent` from the request contained in the specified `FacesContext`, and store this state as needed.

During decoding, events may be enqueued for later processing (by event listeners who have registered an interest), by calling `queueEvent()`.

**Overrides:**

`decode` in class `UIComponentBase`

**Parameters:**

- `context` - `FacesContext` for the request we are processing

**Throws:**

- `NullPointerException` - if `context` is `null`

---

```java
public void updateModel(FacesContext context)
```

- `UIInput`

  - `valid` false
  - `localValueSet` false
  - `value` `ValueExpression`
  - `ValueExpression` `setValue()` `ValueExpression`
  - `setValue()`
    - `UIInput`
    - `UIInput` `localValueSet` false
  - `setValue()`
    - `FacesContext` `addMessage()`
    - `UIInput` `valid` false

**Throws**

`NullPointerException: context null`
updateModel

public void updateModel(FacesContext context)

Perform the following algorithm to update the model data associated with this UIInput, if any, as appropriate.

- If the valid property of this component is false, take no further action.
- If the localValueSet property of this component is false, take no further action.
- If no ValueExpression for value exists, take no further action.
- Call setValue() method of the ValueExpression to update the value that the ValueExpression points at.
- If the setValue() method returns successfully:
  - Clear the local value of this UIInput.
  - Set the localValueSet property of this UIInput to false.
- If the setValue() method call fails:
  - Enqueue an error message by calling addMessage() on the specified FacesContext instance.
  - Set the valid property of this UIInput to false.

Parameters:
context - FacesContext for the request we are processing

Throws:
NullPointerException - if context is null

public void validate(FacesContext context)

UIInput

- getSubmittedValue() null
- #getConvertedValue "local value"
- #validateValue
- valid true getValue() setValue()
- null ValueChangeEvent
validate

```java
public void validate(FacesContext context)
```

Perform the following algorithm to validate the local value of this `UIInput`.

- Retrieve the submitted value with `getSubmittedValue()`. If this returns null, exit without further processing. (This indicates that no value was submitted for this component.)
- Convert the submitted value into a "local value" of the appropriate data type by calling `getConvertedValue(javax.faces.context.FacesContext, java.lang.Object)`.
- Validate the property by calling `validateValue(javax.faces.context.FacesContext, java.lang.Object)`.
- If the `valid` property of this component is still `true`, retrieve the previous value of the component (with `getValue()`), store the new local value using `setValue()`, and reset the submitted value to null. If the local value is different from the previous value of this component, fire a `ValueChangeEvent` to be broadcast to all interested listeners.

Application components implementing `UIInput` that wish to perform validation with logic embedded in the component should perform their own correctness checks, and then call the `super.validate()` method to perform the standard processing described above.

**Parameters:**
- `context` - The `FacesContext` for the current request

**Throws:**
- `NullPointerException` - if `context` is null
protected Object getConvertedValue(FacesContext context, Object newSubmittedValue) throws ConverterException

"local value"

- Renderer getConvertedValue()
- Renderer String Converter
  - getConverter() null Converter
  - value getType()
    - null String
    - Application.createConverter(Class)
      Converter
- ConvertergetAsObject()
  - FacesContext addMessage()
  - valid false

getConvertedValue

protected Object getConvertedValue(FacesContext context, Object newSubmittedValue) throws ConverterException

Convert the submitted value into a "local value" of the appropriate data type, if necessary. Employ the following algorithm to do so:

- If a Renderer is present, call getConvertedValue() to convert the submitted value.
- If no Renderer is present, and the submitted value is a String, locate a Converter as follows:
  - If getConverter() returns a non-null Converter, use that
instance.
  • Otherwise, if a value binding for value exists, call getType() on it.
    • If this call returns null, assume the output type is String and perform no conversion.
    • Otherwise, call Application.createConverter(Class) to locate any registered Converter capable of converting data values of the specified type.
  • If a Converter instance was located, call itsgetAsObject() method to perform the conversion. If conversion fails:
    • Enqueue an appropriate error message by calling the addMessage() method on the FacesContext.
    • Set the valid property on this component to false
  • Otherwise, use the submitted value without any conversion

This method can be overridden by subclasses for more specific behavior.

Throws:
  ConverterException

protected void validateValue(FacesContext context, Object newValue)

"valid"

  • valid true required true"empty"
  null 0 String
    • FacesContext addMessage()
      #getRequiredMessage null FacesContext
      FacesMessage summary detail
      #REQUIRED_MESSAGE_ID
    • valid false

  • valid true UIInput Validator validate()
    validatorBinding ValidatorException
      FacesContext valid false
validateValue

protected void validateValue(FacesContext context, Object newValue)

Set the "valid" property according to the below algorithm.

- If the valid property on this component is still true, and the required property is also true, ensure that the local value is not empty (where "empty" is defined as null or a zero-length String). If the local value is empty:
  - Enqueue an appropriate error message by calling the addMessage() method on the FacesContext instance for the current request. If the getRequiredMessage() returns non-null, use the value as the summary and detail in the FacesMessage that is enqueued on the FacesContext, otherwise use the message for the REQUIRED_MESSAGE_ID.
  - Set the valid property on this component to false.
- If the valid property on this component is still true, and the local value is not empty, call the validate() method of each Validator registered for this UIInput, followed by the method pointed at by the validatorBinding property (if any). If any of these validators or the method throws a ValidatorException, catch the exception, add its message (if any) to the FacesContext, and set the valid property of this component to false.

protected boolean compareValues(Object previous, Object value)

  true

    previous
    value
compareValues

protected boolean compareValues(Object previous, Object value)

Return true if the new value is different from the previous value.

Parameters:
previous - old value of this component (if any)
value - new value of this component (if any)

public void addValidator(Validator validator)

Validator UIInput

validator
Throws
Validator
NullPointerException: validator null

addValidator

public void addValidator(Validator validator)

Add a Validator instance to the set associated with this UIInput.

Specified by:
addValidator in interface EditableValueHolder

Parameters:
validator - The Validator to add

Throws:
NullPointerException - if validator is null

public Validator[] getValidators()
getValidators

public Validator[] getValidators()

Return the set of registered Validators for this UIInput instance. If there are no registered validators, a zero-length array is returned.

Specified by: getValidators in interface EditableValueHolder

public void removeValidator(Validator validator)

Remove a Validator instance from the set associated with this UIInput, if it was previously associated. Otherwise, do nothing.

Specified by: removeValidator in interface EditableValueHolder
Parameters:
 validator - The Validator to remove

public void addValueChangeListener(ValueChangeListener listener)

ValueChangeEvent ValueChangeListener
addValueChangeListener

public void addValueChangeListener(ValueChangeListener listener)

Add a new ValueChangeListener to the set of listeners interested in being notified when ValueChangeEvent occurs.

Specified by:
    addValueChangeListener in interface EditableValueHolder

Parameters:
    listener - The ValueChangeListener to be added

Throws:
    NullPointerException - if listener is null

-------------------------------------------

public ValueChangeListener[] getValueChangeListeners()

UIInput    ValueChangeListener  0

getValueChangeListeners

public ValueChangeListener[] getValueChangeListeners()

Return the set of registered ValueChangeListenerS for this UIInput instance. If there are no registered listeners, a zero-length array is returned.

Specified by:
    getValueChangeListeners in interface EditableValueHolder

-------------------------------------------

public void
removeValueChangeListener

**ValueChangeEvent**  **ValueChangeListener**

<table>
<thead>
<tr>
<th>listener</th>
<th>ValueChangeListener</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td>NullPointerException: listener null</td>
</tr>
</tbody>
</table>

**removeValueChangeListener**

```java
public void removeValueChangeListener(ValueChangeListener listener)
```

Remove an existing `ValueChangeListener` (if any) from the set of listeners interested in being notified when `ValueChangeEvent`s occur.

**Specified by:**

`removeValueChangeListener` in interface `EditableValueHolder`

**Parameters:**

- `listener` - The `ValueChangeListener` to be removed

**Throws:**

- `NullPointerException` - if `listener` is null

---

**public Object saveState(FacesContext context)**

**saveState**

```java
public Object saveState(FacesContext context)
```

**Description copied from interface: StateHolder**

Gets the state of the instance as a `Serializable` Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a `UIComponent` with event handlers, validators, etc.) this method must call the
StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**
saveState in interface StateHolder

**Overrides:**
saveState in class UIOutput

---

**public void restoreState(FacesContext context, Object state)**

restoreState

**public void restoreState(FacesContext context,**
**Object state)**

**Description copied from interface:** StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.
Specified by:  
`restoreState` in interface `StateHolder`

Overrides:  
`restoreState` in class `UIOutput`
javax.faces.component Class UIMessage

java.lang.Object
   └─javax.faces.component.UIComponent
       └─javax.faces.component.UIComponentBase
           └─javax.faces.component.UIMessage

All Implemented Interfaces:
   StateHolder

Direct Known Subclasses:
   HtmlMessage

public class UIMessage
    extends UIComponentBase

Extends: UIComponent > UIComponentBase
Extended by: HtmlMessage

    clientId   UIComponent   FacesContext
    
    rendererType "javax.faces.Message"   setRendererType()

This component is responsible for displaying messages for a specific UIComponent, identified by a clientId. The component obtains the messages from the FacesContext.

By default, the rendererType property must be set to "javax.faces.Message". This value can be changed by calling the setRendererType() method.

Field Summary
### Constructor Summary

**UIMessage()**
Create a new UIMessage instance with default property values.

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><strong>getFamily()</strong></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td>String</td>
<td><strong>getFor()</strong></td>
<td>Return the client identifier of the component for which this component represents associated message(s) (if any).</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isShowDetail()</strong></td>
<td>Return the flag indicating whether the detail property of the associated message(s) should be displayed.</td>
</tr>
<tr>
<td>boolean</td>
<td><strong>isShowSummary()</strong></td>
<td>Return the flag indicating whether the summary property of the associated message(s) should be displayed.</td>
</tr>
<tr>
<td>void</td>
<td><strong>restoreState</strong>(FacesContext context, Object state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td><strong>saveState</strong>(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td><strong>setFor</strong>(String newFor)</td>
<td>Set the client identifier of the component for which this component represents associated message(s) (if any).</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.faces.component.UIComponent

- **COMPONENT_FAMILY**
The standard component family for this component.
- **COMPONENT_TYPE**
The standard component type for this component.

**bindings**
Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`
- `broadcast`
- `decode`
- `encodeBegin`
- `encodeChildren`
- `encodeEnd`
- `findComponent`
- `getAttributes`
- `getChildCount`
- `getChildren`
- `getClientId`
- `getFacesContext`
- `getFacesListeners`
- `getFacet`
- `getFacetCount`
- `getFacets`
- `getFacetsAndChildren`
- `getId`
- `getParent`
- `getRenderer`
- `getRendererType`
- `getRendersChildren`
- `getValueBinding`
- `invokeOnComponent`
- `isRendered`
- `isTransient`
- `processDecodes`
- `processRestoreState`
- `processSaveState`
- `processUpdates`
- `processValidators`
- `queueEvent`
- `removeFacesListener`
- `restoreAttachedState`
- `saveAttachedState`
- `setId`
- `setParent`
- `setRendered`
- `setRendererType`
- `setTransient`
- `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`
- `getContainerClientId`
- `getValueExpression`
- `setValueExpression`

Methods inherited from class `java.lang.Object`

- `clone`
- `equals`
- `finalize`
- `getClass`
- `hashCode`
- `notify`
- `notifyAll`
- `toString`
- `wait`
- `wait`
- `wait`
COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UIMessage()

UIMessage

UIMessage

public UIMessage()

Create a new UIMessage instance with default property values.

Method Detail

public String getFamily()

ggetFamily

public String getFamily()
Description copied from class: [UIComponent](#)

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:  
  [getFamily](#) in class [UIComponent](#)

---

**public String getFor()**

---

### getFor

**public String getFor()**

Return the client identifier of the component for which this component represents associated message(s) (if any).

---

**public void setFor(String newFor)**

---

### setFor

**public void setFor(String newFor)**

Set the client identifier of the component for which this component represents associated message(s) (if any). This property must be set
before the message is displayed.

**Parameters:**
- `newFor` - The new client id

---

```java
public boolean isShowDetail()
{
    return true;
}
```

**isShowDetail**

Public boolean `isShowDetail()`

Return the flag indicating whether the `detail` property of the associated message(s) should be displayed. Defaults to `true`.

---

```java
public void setShowDetail(boolean showDetail)
{
    this.showDetail = showDetail;
}
```

**setShowDetail**

Public void `setShowDetail(boolean showDetail)`

Set the flag indicating whether the `detail` property of the associated message(s) should be displayed.

**Parameters:**
- `showDetail` - The new flag
public boolean isShowSummary()

    summary false

isShowSummary

public boolean isShowSummary()

    Return the flag indicating whether the summary property of the associated message(s) should be displayed. Defaults to false.

public void setShowSummary(boolean showSummary)

    summary

        showSummary

setShowSummary

public void setShowSummary(boolean showSummary)

    Set the flag indicating whether the summary property of the associated message(s) should be displayed.

    Parameters:
        showSummary - The new flag value

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)
Description copied from interface: **StateHolder**

Gets the state of the instance as a **Serializable** Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a **UIComponent** with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext)` method on all those instances as well. **This method must not save the state of children and facets.** That is done via the **StateManager**

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be **Serializable**

**Specified by:**

- `saveState` in interface **StateHolder**

**Overrides:**

- `saveState` in class **UIComponentBase**

---

**public void restoreState(FacesContext context, Object state)**

**restoreState**

**public void restoreState(FacesContext context, Object state)**

**Description copied from interface: **StateHolder**

Perform any processing required to restore the state from the entries in the state Object.
If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
restoreState in interface StateHolder

Overrides:
restoreState in class UIComponentBase
javax.faces.component Class UIMessages

java.lang.Object
   └─javax.faces.component.UIComponent
        └─javax.faces.component.UIComponentBase
            └─javax.faces.component.UIMessages

All Implemented Interfaces:
   StateHolder

Direct Known Subclasses:
   HtmlMessages

public class UIMessages
    extends UIComponentBase

Extends: UIComponent > UIComponentBase
Extended by: HtmlMessages

renderer    FacesContext

   Messages renderer

   rendererType "javax.faces.Messages"    setRendererType()

The renderer for this component is responsible for obtaining the messages from the FacesContext and displaying them to the user.

This component supports the Messages renderer-type.

By default, the rendererType property must be set to "javax.faces.Messages". This value can be changed by calling the setRendererType() method.
### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>static String COMPONENT_FAMILY</code></td>
<td>The standard component family for this component.</td>
</tr>
<tr>
<td><code>static String COMPONENT_TYPE</code></td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class `javax.faces.component.UIComponent` bindings

### Constructor Summary

**UIMessages()**

Create a new `UIMessages` instance with default property values.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>boolean isGlobalOnly()</code></td>
<td>Return the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered.</td>
</tr>
<tr>
<td><code>boolean isShowDetail()</code></td>
<td>Return the flag indicating whether the detail property of the associated message(s) should be displayed.</td>
</tr>
<tr>
<td><code>boolean isShowSummary()</code></td>
<td>Return the flag indicating whether the summary property of the associated message(s) should be displayed.</td>
</tr>
<tr>
<td><code>void restoreState(FacesContext context, Object state)</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>Object saveState(FacesContext context)</code></td>
<td>Gets the state of the instance as a <code>Serializable</code> Object.</td>
</tr>
<tr>
<td><code>setGlobalOnly(boolean globalOnly)</code></td>
<td>Set the global flag for the component.</td>
</tr>
<tr>
<td>void</td>
<td>Set the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered.</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| void | **setShowDetail**(boolean showDetail)  
Set the flag indicating whether the detail property of the associated message(s) should be displayed. |
| void | **setShowSummary**(boolean showSummary)  
Set the flag indicating whether the summary property of the associated message(s) should be displayed. |

**Methods inherited from class** **javax.faces.component.UIComponentBase**

- addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding

**Methods inherited from class** **javax.faces.component.UIComponent**

- encodeAll, getContainerClientId, getValueExpression, setValueExpression

**Methods inherited from class** **java.lang.Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Field Detail**

COMPONENT_TYPE
public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
   Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
   Constant Field Values

Constructor Detail

public UIMessages()

UIMessages

UIMessages

public UIMessages()

Create a new UIMessages instance with default property values.

Method Detail

public String getFamily()
getFamily

public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:
   getFamily in class UIComponent

-----------------------------------------------

public boolean isGlobalOnly()

false

isGlobalOnly

public boolean isGlobalOnly()

Return the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered. Defaults to false.

-----------------------------------------------

public void setGlobalOnly(boolean globalOnly)

   globalOnly

setGlobalOnly
public void setGlobalOnly(boolean globalOnly)

Set the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered.

Parameters:
  globalOnly - The new flag value

---------------------------------------------

public boolean isShowDetail()

detail false

---------------------------------------------

isShowDetail

public boolean isShowDetail()

Return the flag indicating whether the detail property of the associated message(s) should be displayed. Defaults to false.

---------------------------------------------

public void setShowDetail(boolean showDetail)

detail

    showDetail

---------------------------------------------

setShowDetail

public void setShowDetail(boolean showDetail)

Set the flag indicating whether the detail property of the associated message(s) should be displayed.

Parameters:
  showDetail - The new flag
public boolean isShowSummary()

    summary true

isShowSummary

public boolean isShowSummary()

    Return the flag indicating whether the summary property of the associated message(s) should be displayed. Defaults to true.

public void setShowSummary(boolean showSummary)

    summary

    showSummary

setShowSummary

public void setShowSummary(boolean showSummary)

    Set the flag indicating whether the summary property of the associated message(s) should be displayed.

    Parameters:
    showSummary - The new flag value

public Object saveState(FacesContext context)

saveState
public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by: saveState in interface StateHolder
Overrides: saveState in class UIComponentBase

public void restoreState(FacesContext context, Object state)

restoreState

public void restoreState(FacesContext context, Object state)

Description copied from interface: StateHolder
Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:  
restoreState in interface StateHolder

Overrides:  
restoreState in class UIComponentBase
javax.faces.component  **Class UINamingContainer**

java.lang.Object  
  ↘ javax.faces.component.UIComponent  
    ↘ javax.faces.component.UIComponentBase  
      ↘ javax.faces.component.UINamingContainer

**All Implemented Interfaces:**  
   [NamingContainer](#), [StateHolder](#)

---

```java
public class UINamingContainer
extends UIBase
implements NamingContainer

**Extends:** UIBase > UIComponentBase  
**Implements:** NamingContainer
```

**UINamingContainer** is a convenience base class for components that wish to implement [NamingContainer](#) functionality.

---

### Field Summary

| static String COMPONENT_FAMILY | The standard component family for this component. |
| static String COMPONENT_TYPE  | The standard component type for this component. |

**Fields inherited from class javax.faces.component.UIComponent**  
   bindings
### Fields inherited from interface

`javax.faces.component.NamingContainer`  
`SEPARATOR_CHAR`

### Constructor Summary

`UINamingContainer()`  
Create a new `UINamingContainer` instance with default property values.

### Method Summary

#### String `getFamily()`  
Return the identifier of the component family to which this component belongs.

### Methods inherited from class

`javax.faces.component.UIComponentBase`  
`addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `restoreState`, `saveAttachedState`, `saveState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`

### Methods inherited from class

`javax.faces.component.UIComponent`  
`encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`

### Methods inherited from class

`java.lang.Object`  
`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
**Field Detail**

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**
- Constant Field Values

**COMPONENT_FAMILY**

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**
- Constant Field Values

**Constructor Detail**

```java
public UINamingContainer()
```

**UINamingContainer**

```java
public UINamingContainer()
```
Create a new `UINamingContainer` instance with default property values.

**Method Detail**

```java
public String getFamily()
```

### getFamily

```java
public String getFamily()
```

**Description copied from class:** `UIComponent`

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the `rendererType` property, may be used to select the appropriate `Renderer` for this component instance.

**Specified by:**

`getFamily` in class `UIComponent`

---

**Submit a bug or feature**

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component  Class UIOutput

java.lang.Object
  └─ javax.faces.component.UIComponent
      └─ javax.faces.component.UIComponentBase
          └─ javax.faces.component.UIOutput

All Implemented Interfaces:
  StateHolder, ValueHolder

Direct Known Subclasses:
  HtmlOutputFormat, HtmlOutputLabel, HtmlOutputLink, HtmlOutputText, UIInput

public class UIOutput

extends UIComponentBase
implements ValueHolder

Extends: UIComponent > UIComponentBase
Implements: ValueHolder
Extended by: HtmlOutputFormat, HtmlOutputLabel, HtmlOutputLink, HtmlOutputText, UIInput

UIOutput  UIComponent Bean

String String

  • null String Converter
      ◦ getConverter() null Converter Converter
      ◦ Application.createConverter(Class) null Converter Converter
  • null Converter getString()
  • null Converter toString()

rendererType "javax.faces.Text" setRendererType()
UIOutput is a UIComponent that has a value, optionally retrieved from a model tier bean via a value expression, that is displayed to the user. The user cannot directly modify the rendered value; it is for display purposes only.

During the Render Response phase of the request processing lifecycle, the current value of this component must be converted to a String (if it is not already), according to the following rules:

- If the current value is not null, and is not already a String, locate a Converter (if any) to use for the conversion, as follows:
  - If getConverter() returns a non-null Converter, use that one, otherwise
  - If Application.createConverter(Class), passing the current value's class, returns a non-null Converter, use that one.
- If the current value is not null and a Converter was located, call its getAsString() method to perform the conversion.
- If the current value is not null but no Converter was located, call toString() on the current value to perform the conversion.

By default, the rendererType property must be set to "javax.faces.Text". This value can be changed by calling the setRendererType() method.

### Field Summary

<table>
<thead>
<tr>
<th>Static String</th>
<th>COMPONENT_FAMILY</th>
<th>The standard component family for this component.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.faces.component.UIComponent

| bindings |

### Constructor Summary

UIOutput()  
Create a new UIOutput instance with default property values.
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getConverter()</code></td>
<td>Return the <code>Converter</code> (if any) that is registered for this <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>getLocalValue()</code></td>
<td>Return the local value of this <code>UIComponent</code> (if any), without evaluating any associated <code>ValueExpression</code>.</td>
</tr>
<tr>
<td><code>getValue()</code></td>
<td>Gets the value of this <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>restoreState()</code></td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td><code>saveState()</code></td>
<td>Gets the state of the instance as a <code>Serializable</code> Object.</td>
</tr>
<tr>
<td><code>setConverter()</code></td>
<td>Set the <code>Converter</code> (if any) that is registered for this <code>UIComponent</code>.</td>
</tr>
<tr>
<td><code>setValue()</code></td>
<td>Set the value of this <code>UIComponent</code> (if any).</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildCount`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`
### Field Detail

**COMPONENT_TYPE**

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

[Constant Field Values](#)

---

**COMPONENT_FAMILY**

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**

[Constant Field Values](#)
public UIOutput()

UIOutput

public UIOutput()

Create a new UIOutput instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:

getFamily in class UIComponent

public Converter getConverter()

getConverter
public Converter getConverter()

Description copied from interface: ValueHolder

Return the Converter (if any) that is registered for this UIComponent.

Specified by: getConverter in interface ValueHolder

public void setConverter(Converter converter)

setConverter

public void setConverter(Converter converter)

Description copied from interface: ValueHolder

Set the Converter (if any) that is registered for this UIComponent.

Specified by: setConverter in interface ValueHolder

Parameters:
  converter - New Converter (or null)

public Object getLocalValue()

getLocalValue

public Object getLocalValue()

Description copied from interface: ValueHolder

Return the local value of this UIComponent (if any), without evaluating any associated ValueExpression.
public Object getValue()

getValue

public Object getValue()

Description copied from interface: ValueHolder

 Gets the value of this UICOMPONENT. First, consult the local value property of this component. If non-null return it. If null, see if we have a ValueExpression for the value property. If so, return the result of evaluating the property, otherwise return null. Note that because the specification for UICOMPONENT.setValueBinding(java.lang.String, javax.faces.el.ValueBinding) requires a call through to UICOMPONENT.setValueExpression(java.lang.String, javax.el.ValueExpression), legacy tags will continue to work.

Specified by: getValue in interface ValueHolder

public void setValue(Object value)

setValue

public void setValue(Object value)

Description copied from interface: ValueHolder

Set the value of this UICOMPONENT (if any).

Specified by: setValue in interface ValueHolder
Parameters:
  value - The new local value

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via theStateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

    Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable

Specified by: saveState in interface StateHolder

Overrides: saveState in class UIComponentBase

public void restoreState(FacesContext context, Object
public void restoreState(FacesContext context, Object state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.

Specified by:
restoreState in interface StateHolder

Overrides:
restoreState in class UICOMPONENTBASE
javax.faces.component Class UIPanel

java.lang.Object
  └─javax.faces.component.UIComponent
    └─javax.faces.component.UIComponentBase
        └─javax.faces.component.UIPanel

All Implemented Interfaces:
  StateHolder

Direct Known Subclasses:
  HtmlPanelGrid, HtmlPanelGroup

public class UIPanel
  extends UIComponentBase

Extends: UIComponent > UIComponentBase
Extended by: HtmlPanelGrid, HtmlPanelGroup

UIPanel is a UIComponent that manages the layout of its child components.

---

Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>COMPONENT_FAMILY</th>
<th>The standard component family for this component.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

Fields inherited from class javax.faces.component.UIComponent
Constructor Summary

**UIPanel()**
Create a new **UIPanel** instance with default property values.

Method Summary

**getFamily()**
Return the identifier of the component family to which this component belongs.

Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `getValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `restoreState`, `saveAttachedState`, `saveState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`

Methods inherited from class `javax.faces.component.UIComponent`

- `encodeAll`, `getContainerClientId`, `getValueExpression`, `setValueExpression`

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UIPanel()

UIPanel

UIPanel

public UIPanel()

Create a new UIPanel instance with default property values.

Method Detail
public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by: getFamily in class UIComponent
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
</tr>
</tbody>
</table>
javax.faces.component Class UIParameter

java.lang.Object
  ↑  javax.faces.component.UIComponent
  |  ↑  javax.faces.component.UIComponentBase
  |  ↑  javax.faces.component.UIParameter

All Implemented Interfaces:
  StateHolder

public class UIParameter
    extends UIComponentBase

Extends: UIComponent > UIComponentBase

UIParameter  UIComponent,

    getValue()         setValue()

    getValue()   UIParameter
javax.faces.render_RENDERER

UIParameter is a UIComponent that represents an optionally named configuration parameter for a parent component.

Parent components should retrieve the value of a parameter by calling getValue(). In this way, the parameter value can be set directly on the component (via setValue()), or retrieved indirectly via the value binding expression.

In some scenarios, it is necessary to provide a parameter name, in addition to the parameter value that is accessible via the getValue() method. Renderers that support parameter names on their nested UIParameter child components should document their use of this property.
## Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_FAMILY</td>
<td>The standard component family for this component.</td>
</tr>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

## Constructor Summary

**UIParameter()**
Create a new `UIParameter` instance with default property values.

## Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>getFamily()</td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td>String</td>
<td>getName()</td>
<td>Return the optional parameter name for this parameter.</td>
</tr>
<tr>
<td>Object</td>
<td>getValue()</td>
<td>Returns the value property of the UIParameter.</td>
</tr>
<tr>
<td>void</td>
<td>restoreState(FacesContext context, Object state)</td>
<td>Perform any processing required to restore the state from the entries in the state Object.</td>
</tr>
<tr>
<td>Object</td>
<td>saveState(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>void</td>
<td>setName(String name)</td>
<td>Set the optional parameter name for this parameter.</td>
</tr>
<tr>
<td>void</td>
<td>setValue(Object value)</td>
<td>Sets the value property of the UIParameter.</td>
</tr>
</tbody>
</table>

## Methods inherited from class
<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UiComponentBase</th>
</tr>
</thead>
<tbody>
<tr>
<td>addFacesListener, broadcast, decode, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParents, getRenderer, getRendererType, getRendersChildren, getValueBinding, invokeOnComponent, isRendered, isTransient, processDecodes, processRestoreState, processSaveState, processUpdates, processValidators, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient, setValueBinding</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait</td>
</tr>
</tbody>
</table>

**Field Detail**

**COMPONENT_TYPE**

public static final String COMPONENT_TYPE

The standard component type for this component.

**See Also:**

[Constant Field Values](#)
public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UIParameter()

UIParameter

UIParameter

public UIParameter()

Create a new UIParameter instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.
public String getName()

getName

public String getName()

Return the optional parameter name for this parameter.

public void setName(String name)

setName

public void setName(String name)

Set the optional parameter name for this parameter.

Parameters:
  name - The new parameter name, or null for no name

public Object getValue()

getValue

public Object getValue()

UIParameter value
getValue

public Object getValue()

Returns the value property of the UIParameter.

setValue

public void setValue(Object value)

UIParameter value

Sets the value property of the UIParameter.

Parameters:
value - the new value

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to
instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable.

Specified by:
- saveState in interface StateHolder

Overrides:
- saveState in class UIComponentBase

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface:** StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, Object state).
java.lang.Object) method on all those instances as well.

Specified by:
   restoreState in interface StateHolder

Overrides:
   restoreState in class UIComponentBase
javax.faces.component  Class UISelectBoolean

java.lang.Object
  ▼ javax.faces.component.UIComponent
    ▼ javax.faces.component.UIComponentBase
      ▼ javax.faces.component.UIOutput
        ▼ javax.faces.component.UIInput
          ▼ javax.faces.component.UISelectBoolean

All Implemented Interfaces:
  EditableValueHolder, StateHolder, ValueHolder

Direct Known Subclasses:
  HtmlSelectBooleanCheckbox

public class UISelectBoolean
  extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput
Extended by: HtmlSelectBooleanCheckbox

UISelectBoolean  boolean  true  false  UIComponent

  rendererType "javax.faces.Checkbox"  setRendererType()

UISelectBoolean is a UIComponent that represents a single boolean (true or false) value. It is most commonly rendered as a checkbox.

By default, the rendererType property must be set to "javax.faces.Checkbox". This value can be changed by calling the setRendererType() method.

---

Field Summary
| static String COMPONENT_FAMILY | The standard component family for this component. |
| static String COMPONENT_TYPE | The standard component type for this component. |

Fields inherited from class javax.faces.component.UIInput
CONVERSION_MESSAGE_ID, REQUIRED_MESSAGE_ID, UPDATE_MESSAGE_ID

Fields inherited from class javax.faces.component.UIComponent
bindings

## Constructor Summary

**UISelectBoolean()**
Create a new **UISelectBoolean** instance with default property values.

## Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td><code>getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td>ValueBinding</td>
<td><code>getValueBinding(String name)</code></td>
<td><strong>Deprecated. This has been replaced by</strong> getValueExpression(java.lang.String).</td>
</tr>
<tr>
<td>ValueExpression</td>
<td><code>getValueExpression(String name)</code></td>
<td>Return any ValueExpression set for value if a ValueExpression for selected is requested; otherwise, perform the default superclass processing for this method.</td>
</tr>
<tr>
<td>boolean</td>
<td><code>isSelected()</code></td>
<td>Return the local value of the selected state of this component.</td>
</tr>
<tr>
<td>void</td>
<td><code>setSelected(boolean selected)</code></td>
<td>Set the local value of the selected state of this component.</td>
</tr>
</tbody>
</table>
void **setValueBinding**(String name, ValueBinding binding)

**Deprecated. This has been replaced by**


void **setValueExpression**(String name, ValueExpression binding)

Store any ValueExpression specified for selected under value instead; otherwise, perform the default superclass processing for this method.

### Methods inherited from class javax.faces.component.UIInput

addValidator, addValueChangeListener, compareValues, decode, getConvertedValue, getConverterMessage, getRequiredMessage, getSubmittedValue, getValidator, getValidatorMessage, getValidators, getValueChangeListeners, isImmediate, isLocalValueSet, isRequired, isValid, processDecodes, processUpdates, processValidators, removeValidator, removeValueChangeListener, resetValue, restoreState, saveState, setConverterMessage, setValue, setValueChangeListener, updateModel, validate, validateValue

### Methods inherited from class javax.faces.component.UIOutput

getConverter, getLocalValue, getValue, setConverter

### Methods inherited from class javax.faces.component.UIComponentBase

addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getChildren, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getLabel, getRenderer, getRendererType, getRendersChildren, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient

### Methods inherited from class javax.faces.component.UIComponent

encodeAll, getContainerClientId

---

**Deprecated. This has been replaced by** setValueExpression(java.lang.String, javax.el.ValueExpression).
Methods inherited from class java.lang.**Object**

- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Methods inherited from interface javax.faces.component.**ValueHolder**

- getConverter, getLocalValue, getValue, setConverter

---

**Field Detail**

**COMPONENT_TYPE**

public static final String COMPONENT_TYPE

The standard component type for this component.

**See Also:**

- Constant Field Values

---

**COMPONENT_FAMILY**

public static final String COMPONENT_FAMILY

The standard component family for this component.

**See Also:**

- Constant Field Values

---

**Constructor Detail**

public UISelectBoolean()
public UISelectBoolean()

Create a new UISelectBoolean instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UICOMPONENT

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Overrides:

getFamily in class UINPUT

public boolean isSelected()

isSelected

getValue()
public boolean isSelected()

    Return the local value of the selected state of this component. This method is a typesafe alias for getValue().

public void setSelected(boolean selected)

    setValue()

    selected

setSelected

public void setSelected(boolean selected)

    Set the local value of the selected state of this component. This method is a typesafe alias for setValue().

    Parameters:
    selected - The new selected state

public ValueBinding getValueBinding(String name)

    selected   ValueBinding  value   ValueBinding
    ValueBinding  ValueExpression

    name
    NullPointerException:  name  null
    deprecated

getValueBinding

public ValueBinding getValueBinding(String name)
Deprecated. This has been replaced by `getValueExpression(java.lang.String)`.

Return any ValueBinding set for value if a ValueBinding for selected is requested; otherwise, perform the default superclass processing for this method.

Rely on the superclass implementation to wrap the returned ValueExpression in a ValueBinding.

Overrides: `getValueBinding` in class `UIComponentBase`

Parameters:
- `name` - Name of the attribute or property for which to retrieve a ValueBinding

Throws:
- `NullPointerException` - if name is null

```java
public void setValueBinding(String name, ValueBinding binding)
```

value selected ValueBinding

ValueExpression ValueBinding

```java
name ValueBinding
binding ValueBinding null ValueBinding
```

Throws
- `NullPointerException` - name null

deprecated

#setValueExpression

**set ValueBinding**

```java
public void setValueBinding(String name, ValueBinding binding)
```

**Deprecated. This has been replaced by**
**setValueExpression**(*java.lang.String, javax.el.ValueExpression*).

Store any **ValueBinding** specified for **selected** under **value** instead; otherwise, perform the default superclass processing for this method.

Rely on the superclass implementation to wrap the argument **ValueBinding** in a **ValueExpression**.

**Overrides:**

**setValueBinding** in class **UIComponentBase**

**Parameters:**

- name - Name of the attribute or property for which to set a **ValueBinding**
- binding - The **ValueBinding** to set, or null to remove any currently set **ValueBinding**

**Throws:**

- **NullPointerException** - if name is null

---

**public** **ValueExpression** **getValueExpression**(*String name*)

**selected**  **ValueExpression**  **value**  **ValueExpression**

**name**  **ValueExpression**

**Throws**

- **NullPointerException**

**since**

- 1.2

**getValueExpression**

**public** **ValueExpression** **getValueExpression**(*String name*)

Return any **ValueExpression** set for **value** if a **ValueExpression** for **selected** is requested; otherwise, perform the default superclass processing for this method.

**Overrides:**
getValueExpression in class UIComponent

Parameters:
name - Name of the attribute or property for which to retrieve a ValueExpression

Throws:
NullPointerException - if name is null

Since:
1.2

public void setValueExpression(String name, ValueExpression binding)

value selected ValueExpression

name ValueExpression
binding ValueExpression null ValueExpression

Throws
NullPointerException: name null

since
1.2

setValueExpression

public void setValueExpression(String name, ValueExpression binding)

Store any ValueExpression specified for selected under value instead; otherwise, perform the default superclass processing for this method.

Overrides:
setValueExpression in class UIComponent

Parameters:
name - Name of the attribute or property for which to set a ValueExpression
binding - The ValueExpression to set, or null to remove any currently set ValueExpression

Throws:
NullPointerException - if name is null
Since:
1.2

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.component  Class UISelectItem

java.lang.Object  
  └ javax.faces.component.UIComponent  
    └ javax.faces.component.UIComponentBase  
      └ javax.faces.component.UISelectItem

All Implemented Interfaces:  
  StateHolder

public class UISelectItem  
  extends UIComponentBase

Extends: UIComponent  >  UIComponentBase

UISelectItem  UISelectMany  UISelectOne  SelectItem  SelectItem

- value  SelectItem
- javax.el.ValueExpression  SelectItem
- itemDescription  itemDisabled  itemLabel  itemValue  SelectItem

UISelectItem is a component that may be nested inside a UISelectMany  
or UISelectOne component, and causes the addition of a SelectItem  
instance to the list of available options for the parent component. The  
contents of the SelectItem can be specified in one of the following ways:

- The value attribute's value is an instance of SelectItem.
- The associated ValueExpression points at a model data item of type 
  SelectItem.
- A new SelectItem instance is synthesized from the values of the 
  itemDescription, itemDisabled, itemLabel, and itemValue attributes.
### Field Summary

<table>
<thead>
<tr>
<th>Static Type</th>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_FAMILY</td>
<td>The standard component family for this component.</td>
</tr>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.faces.component.UIComponent

- bindings

### Constructor Summary

**UISelectItem()**

Create a new UISelectItem instance with default property values.

### Method Summary

- **String getFamily()**
  
  Return the identifier of the component family to which this component belongs.

- **String getItemDescription()**
  
  Return the description for this selection item.

- **String getItemLabel()**
  
  Return the localized label for this selection item.

- **Object getItemValue()**
  
  Return the server value for this selection item.

- **Object getValue()**
  
  Returns the value property of the UISelectItem.

- **boolean isItemDisabled()**
  
  Return the disabled setting for this selection item.

- **boolean isItemEscaped()**
  
  Return the escape setting for the label of this selection item.

- **void restoreState(FacesContext context, Object state)**
  
  Perform any processing required to restore the state from
the entries in the state Object.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>saveState(FacesContext context)</td>
<td>Gets the state of the instance as a Serializable Object.</td>
</tr>
<tr>
<td>setItemDescription(String itemDescription)</td>
<td>Set the description for this selection item.</td>
</tr>
<tr>
<td>setItemDisabled(boolean itemDisabled)</td>
<td>Set the disabled value for this selection item.</td>
</tr>
<tr>
<td>setItemEscaped(boolean itemEscaped)</td>
<td>Set the escape value for the label of this selection item.</td>
</tr>
<tr>
<td>setItemLabel(String itemLabel)</td>
<td>Set the localized label for this selection item.</td>
</tr>
<tr>
<td>setItemValue(Object itemValue)</td>
<td>Set the server value for this selection item.</td>
</tr>
<tr>
<td>setValue(Object value)</td>
<td>Sets the value property of the UISelectItem.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.component.UIComponentBase

- addFacesListener
- broadcast
- decode
- encodeBegin
- encodeChildren
- encodeEnd
- findComponent
- getAttributes
- getChildCount
- getChildren
- getClientId
- getFacesContext
- getFacesListeners
- getFacet
- getFacetCount
- getFacets
- getFacetsAndChildren
- getId
- getParent
- getRenderer
- getRendererType
- getRenderersChildren
- getValueBinding
- invokeOnComponent
- isRendered
- isTransient
- processDecodes
- processRestoreState
- processSaveState
- processUpdates
- processValidators
- queueEvent
- removeFacesListener
- restoreAttachedState
- saveAttachedState
- setId
- setParent
- setRendered
- setRendererType
- setTransient
- setValueBinding

Methods inherited from class javax.faces.component.UIComponent

- encodeAll
- getContainerClientId
- getValueExpression
- setValueExpression

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- toString
- wait
- wait
Field Detail

COMPONENT>Type

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UISelectItem()

UISelectItem

UISelectItem

public UISelectItem()
Create a new `UISelectItem` instance with default property values.

### Method Detail

**public String getFamily()**

#### getFamily

**public String getFamily()**

**Description copied from class:** `UIComponent`

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the `rendererType` property, may be used to select the appropriate `Renderer` for this component instance.

**Specified by:**

- `getFamily` in class `UIComponent`

---

**public String getItemDescription()**

#### getItemDescription

**public String getItemDescription()**

Return the description for this selection item.

---

**public void setItemDescription(String itemDescription)**
public void setItemDescription(String itemDescription)

Set the description for this selection item.

Parameters:
  itemDescription - The new description

public boolean isItemDisabled()

public void setItemDisabled(boolean itemDisabled)

public boolean isItemDisabled()
Set the disabled value for this selection item.

**Parameters:**
- itemDisabled - The new disabled flag

---

**public boolean isItemEscaped()**

**isItemEscaped**

**public boolean isItemEscaped()**

Return the escape setting for the label of this selection item.

---

**public void setItemEscaped(boolean itemEscaped)**

**setItemEscaped**

**public void setItemEscaped(boolean itemEscaped)**

Set the escape value for the label of this selection item.

**Parameters:**
- itemEscaped - The new disabled flag

---

**public String getItemLabel()**
**getImltemLabel**

```java
public String getItemLabel()
```

Return the localized label for this selection item.

---

**public void setItemLabel(String itemLabel)**

```java

```

---

**setItemLabel**

```java
public void setItemLabel(String itemLabel)
```

Set the localized label for this selection item.

**Parameters:**

- `itemLabel` - The new localized label

---

**public Object getItemValue()**

---

**getImltemValue**

```java
public Object getItemValue()
```

---
Return the server value for this selection item.

```java
public void setItemValue(Object itemValue)
```

_setItemValue_

```
public void setItemValue(Object itemValue)
```

Set the server value for this selection item.

**Parameters:**

- `itemValue` - The new server value

```java
public Object getValue()
```

_getValue_

```
public Object getValue()
```

Returns the `value` property of the `UISelectItem`.

```java
public void setValue(Object value)
```

dSetValue_

```
public void setValue(Object value)
```

`UISelectItem` value

`value`
setValue

public void setValue(Object value)

Sets the value property of the UISelectItem.

Parameters:
value - the new value

saveState

public Object saveState(FacesContext context)

Description copied from interface: StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the state of children and facets. That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

Object state = component.saveState(facesContext);

component should be the same as before executing it.

The return from this method must be Serializable
public void restoreState(FacesContext context, Object state)

Description copied from interface: StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object) method on all those instances as well.
PS:
javax.faces.component  **Class UISelectItems**

java.lang.Object
| ↓ | javax.faces.component.UIComponent
| ↓ | javax.faces.component.UIComponentBase
| ↓ | javax.faces.component.UISelectItems

All Implemented Interfaces:
- **StateHolder**

```java
class UISelectItems
extends UIComponentBase
```

**Extends:** **UIComponent** > **UIComponentBase**

**UISelectItems**  **UISelectMany**  **UISelectOne**  **UISelectItem**

- **value**
  - javax.el.ValueExpression
  - *SelectItem*
  - *SelectItem* Collection
  - *Map* - String String *SelectItem*

**UISelectItems** is a component that may be nested inside a **UISelectMany** or **UISelectOne** component, and causes the addition of one or more **SelectItem** instances to the list of available options in the parent component. The **value** of this component (set either directly, or acquired indirectly a **ValueExpression**), can be of any of the following types:

- **Single instance of SelectItem** - This instance is added to the set of available options for the parent tag.
- **Array of SelectItem** - This set of instances is added to the set of available options for the parent component, in ascending subscript
order.
- *Collection of SelectItem* - This set of instances is added to the set of available options for the parent component, in the order provided by an iterator over them.
- *Map* - The keys of this object (once converted to Strings) are assumed to be labels, and the values of this object (once converted to Strings) are assumed to be values, of *SelectItem* instances that will be constructed dynamically and added to the set of available options for the parent component, in the order provided by an iterator over the keys.

### Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>COMPONENT_FAMILY</th>
<th>The standard component family for this component.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>COMPONENT_TYPE</td>
<td>The standard component type for this component.</td>
</tr>
</tbody>
</table>

### Fields inherited from class javax.faces.component.UIComponent

- bindings

### Constructor Summary

- **UISelectItems()**
  Create a new *UISelectItems* instance with default property values.

### Method Summary

- **String getFamily()**
  Return the identifier of the component family to which this component belongs.

- **Object getValue()**
  Returns the *value* property of the *UISelectItems*.

- **void restoreState(FacesContext context, Object state)**
  Perform any processing required to restore the state from
the entries in the state Object.

### saveState

```java
Object saveState(FacesContext context)
```

Gets the state of the instance as a Serializable Object.

### setValue

```java
void setValue(Object value)
```

Sets the value property of the UISelectItems.

### Methods inherited from class `javax.faces.component.UIComponentBase`

- `addFacesListener`, `broadcast`, `decode`, `encodeBegin`, `encodeChildren`, `encodeEnd`, `findComponent`, `getAttributes`, `getChildren`, `getClientId`, `getFacesContext`, `getFacesListeners`, `getFacet`, `getFacetCount`, `getFacets`, `getFacetsAndChildren`, `getId`, `getParent`, `getRenderer`, `getRendererType`, `getRendersChildren`, `get_ValueBinding`, `invokeOnComponent`, `isRendered`, `isTransient`, `processDecodes`, `processRestoreState`, `processSaveState`, `processUpdates`, `processValidators`, `queueEvent`, `removeFacesListener`, `restoreAttachedState`, `saveAttachedState`, `setId`, `setParent`, `setRendered`, `setRendererType`, `setTransient`, `setValueBinding`
COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

Constructor Detail

public UISelectItems()

UISelectItems

public UISelectItems()

Create a new UISelectItems instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()
Description copied from class: **UIComponent**

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the `rendererType` property, may be used to select the appropriate `Renderer` for this component instance.

**Specified by:**

`getFamily` in class **UIComponent**

---

### public Object getValue()

**UISelectItems**  
**value**

**getValue**

public **Object** getValue()

Returns the `value` property of the `UISelectItems`.

---

### public void setValue(Object value)

**UISelectItems**  
**value**

**value**

**setValue**

public void **setValue**(Object value)

Sets the `value` property of the `UISelectItems`.

**Parameters:**
public Object saveState(FacesContext context)

**Description copied from interface: **StateHolder

Gets the state of the instance as a Serializable Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. **This method must not save the state of children and facets.** That is done via the StateManager

This method must not alter the state of the implementing object. In other words, after executing this code:

```
Object state = component.saveState(facesContext);
```

component should be the same as before executing it.

The return from this method must be Serializable

**Specified by:**

saveState in interface StateHolder

**Overrides:**

saveState in class UIComponentBase

public void restoreState(FacesContext context, Object state)
restoreState

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface:** StateHolder

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**

`restoreState` in interface StateHolder

**Overrides:**

`restoreState` in class UIComponentBase
javax.faces.component Class UISelectMany

java.lang.Object
  ↓ javax.faces.component.UIComponent
     ↓ javax.faces.component.UIComponentBase
        ↓ javax.faces.component.UIOutput
           ↓ javax.faces.component.UIInput
              ↓ javax.faces.component.UISelectMany

All Implemented Interfaces:
   EditableValueHolder, StateHolder, ValueHolder

Direct Known Subclasses:
   HtmlSelectManyCheckbox, HtmlSelectManyListbox, HtmlSelectManyMenu

public class UISelectMany
  extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput
Extended by: HtmlSelectManyCheckbox, HtmlSelectManyListbox, HtmlSelectManyMenu

UISelectMany 0  UIComponent 0  value

rendererType "javax.faces.Listbox"  setRendererType()

javax.faces.render.Renderer  getConvertedValue()

javax.faces.convert.Converter

javax.faces.convert.Converter

value ValueExpression  ValueExpression
UISelectMany is a UIComponent that represents the user's choice of a zero or more items from among a discrete set of available options. The user can modify the selected values. Optionally, the component can be preconfigured with zero or more currently selected items, by storing them as an array in the value property of the component.

This component is generally rendered as a select box or a group of checkboxes.

By default, the rendererType property must be set to "javax.faces.Listbox". This value can be changed by calling the setRendererType() method.

The Renderer for this component must perform the following logic on getConvertedValue():

Obtain the Converter using the following algorithm:

- If the component has an attached Converter, use it.
- If not, look for a ValueExpression for value (if any). The ValueExpression must point to something that is:
  - An array of primitives (such as int[]). Look up the registered by-class Converter for this primitive type.
  - An array of objects (such as Integer[] or String[]). Look up the registered by-class Converter for the underlying element type.
- A `java.util.List`. Assume that the element type is `java.lang.String`, so no conversion is required.

If for any reason a `Converter` cannot be found, assume the type to be a String array.

Use the selected `Converter` (if any) to convert each element in the values array or list from the request to the proper type. If the component has a `ValueBinding` for value, create an array of the expected type to hold the converted values. If the component does not have a `ValueBinding` for value, create an array of type `Object`. Store the created array as the local value of the component, set the component’s `valid` state to `true` and return.

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static String COMPONENT_FAMILY</strong></td>
</tr>
<tr>
<td><strong>static String COMPONENT_TYPE</strong></td>
</tr>
<tr>
<td><strong>static String INVALID_MESSAGE_ID</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from class <code>javax.faces.component.UIInput</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>CONVERSION_MESSAGE_ID</code>, <code>REQUIRED_MESSAGE_ID</code>, <code>UPDATE_MESSAGE_ID</code></td>
</tr>
</tbody>
</table>

| Fields inherited from class `javax.faces.component.UIComponent` bindings |

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UISelectMany()</code> Create a new <code>UISelectMany</code> instance with default property values.</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>compareValues</code></td>
<td><code>protected boolean compareValues(Object previous, Object value)</code></td>
<td>Return <code>true</code> if the new value is different from the previous value.</td>
</tr>
<tr>
<td><code>getFamily</code></td>
<td><code>String getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>getSelectedValues</code></td>
<td><code>Object[] getSelectedValues()</code></td>
<td>Return the currently selected values, or <code>null</code> if there are no currently selected values.</td>
</tr>
<tr>
<td><code>getValueBinding</code></td>
<td><code>ValueBinding getValueBinding(String name)</code></td>
<td>Deprecated. this has been replaced by <code>getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td><code>getValueExpression</code></td>
<td><code>ValueExpression getValueExpression(String name)</code></td>
<td>Return any <code>ValueExpression</code> set for <code>value</code> if a <code>ValueExpression</code> for <code>selectedValues</code> is requested; otherwise, perform the default superclass processing for this method.</td>
</tr>
<tr>
<td><code>setSelectedValues</code></td>
<td><code>void setSelectedValues(Object[] selectedValues)</code></td>
<td>Set the currently selected values, or <code>null</code> to indicate that there are no currently selected values.</td>
</tr>
<tr>
<td><code>setValueBinding</code></td>
<td><code>void setValueBinding(String name, ValueBinding binding)</code></td>
<td>Deprecated. This has been replaced by <code>setValueExpression(java.lang.String, javax.el.ValueExpression)</code></td>
</tr>
<tr>
<td><code>setValueExpression</code></td>
<td><code>void setValueExpression(String name, ValueExpression binding)</code></td>
<td>Store any <code>ValueExpression</code> specified for <code>selectedValues</code> under <code>value</code> instead; otherwise, perform the default superclass processing for this method.</td>
</tr>
<tr>
<td><code>validateValue</code></td>
<td><code>protected void validateValue(FacesContext context, Object value)</code></td>
<td>In addition to the standard validation behavior inherited from <code>UIInput</code>, ensure that any specified values are equal to one of the available options.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.component.UIInput`
<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UiOutput</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>getConverter, getLocalValue, getValue, setConverter</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UiComponentBase</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>addFacesListener, broadcast, encodeBegin, encodeChildren, encodeEnd, findComponent, getAttributes, getChildCount, getSupportFragmentManager, getClientId, getFacesContext, getFacesListeners, getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId, getParent, getRenderer, getRendererType, getRendersChildren, invokeOnComponent, isRendered, isTransient, processRestoreState, processSaveState, queueEvent, removeFacesListener, restoreAttachedState, saveAttachedState, setId, setParent, setRendered, setRendererType, setTransient</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UiComponent</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>encodeAll, getContainerClientId</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface javax.faces.component.ValueHolder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>getConverter, getLocalValue, getValue, setConverter</td>
<td></td>
</tr>
</tbody>
</table>

addValidator, addValueChangeListener, decode, getConvertedValue, getConverterMessage, getRequiredMessage, getSubmittedValue, getValidator, getValidatorMessage, getValidators, getValueChangeListener, getValueChangeListeners, isImmediate, isLocalValueSet, isRequired, isValid, processDecodes, processUpdates, processValidators, removeValidator, removeValueChangeListener, resetValue, restoreState, saveState, setConverterMessage, setImmediate, setLocalValueSet, setRequired, setRequiredMessage, setSubmittedValue, setValid, setValidator, setValidatorMessage, setValue, setValueChangeListener, updateModel, validate
### Field Detail

#### COMPONENT_TYPE

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**

[Constant Field Values](#)

---

#### COMPONENT_FAMILY

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**

[Constant Field Values](#)

---

#### INVALID_MESSAGE_ID

```java
public static final String INVALID_MESSAGE_ID
```

The message identifier of the `FacesMessage` to be created if a value not matching the available options is specified.

**See Also:**

[Constant Field Values](#)
### Constructor Detail

public UISelectMany()

UISelectMany

public UISelectMany()

Create a new UISelectMany instance with default property values.

### Method Detail

public String getFamily()

getFamily

public String getFamily()

**Description copied from class: UIComponent**

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

**Overrides:**

g Familie in class UIInput

public Object[] getSelectedValues()
null getValue()

**getSelectedValues**

```java
public Object[] getSelectedValues()
```

Return the currently selected values, or `null` if there are no currently selected values. This is a typesafe alias for `getValue()`.

---

**public void setSelectedValues(Object[] selectedValues)**

```java
null setValue()
```

```
selectedValues
```

**setSelectedValues**

```java
public void setSelectedValues(Object[] selectedValues)
```

Set the currently selected values, or `null` to indicate that there are no currently selected values. This is a typesafe alias for `setValue()`.

**Parameters:**

- `selectedValues` - The new selected values (if any)

---

**public ValueBinding getValueBinding(String name)**

```java
selectedValues ValueBinding value ValueBinding
```

```
ValueExpression ValueBinding
```

```
name ValueBinding
```
Throws  NullPointerException:  name  null
deprecated  #getValueExpression(java.lang.String)

getValueBinding

public  ValueBinding  getValueBinding(String  name)

   Deprecated.  this has been replaced by
   getValueExpression(java.lang.String).

Return any ValueBinding set for value if a ValueBinding for
selectedValues is requested; otherwise, perform the default
superclass processing for this method.

This method relies on the superclass to provide the ValueExpression
to ValueBinding wrapping.

Overrides:
   getValueBinding  in class  UICOMPONENTBASE

Parameters:
   name - Name of the attribute or property for which to retrieve a
   ValueBinding

Throws:
   NullPointerException - if name is null

---

public  void  setValueBinding(String  name,  ValueBinding
binding)

value  selectedValues  ValueBinding

ValueExpression  ValueBinding

   name  ValueBinding
binding  ValueBinding  null  ValueBinding
Threws  NullPointerException:  name  null

#setValueExpression(java.lang.String)
setValueBinding

public void setValueBinding(String name, ValueBinding binding)

Deprecated. This has been replaced by setValueExpression(java.lang.String, javax.el.ValueExpression).

Store any ValueBinding specified for selectedValues under value instead; otherwise, perform the default superclass processing for this method.

This method relies on the superclass to wrap the argument ValueBinding in a ValueExpression.

Overrides:
setValueBinding in class UIComponentBase

Parameters:
- name - Name of the attribute or property for which to set a ValueBinding
- binding - The ValueBinding to set, or null to remove any currently set ValueBinding

Throws:
- NullPointerException - if name is null

public ValueExpression getValueExpression(String name)

selectedValues ValueExpression value
ValueExpression

name ValueExpression

Throws
- NullPointerException: name null

since 1.2
**getValueExpression**

```java
public ValueExpression getValueExpression(String name)
```

Return any `ValueExpression` set for value if a `ValueExpression` for `selectedValues` is requested; otherwise, perform the default superclass processing for this method.

**Overrides:**
- `getValueExpression` in class `UIComponent`

**Parameters:**
- `name` - Name of the attribute or property for which to retrieve a `ValueExpression`

**Throws:**
- `NullPointerException` - if `name` is null

**Since:**
1.2

---

**setValueExpression**

```java
public void setValueExpression(String name, ValueExpression binding)
```

Store any `ValueExpression` specified for `selectedValues` under `value` instead; otherwise, perform the default superclass processing for this method.

**Throws:**
- `NullPointerException` - if name is null

**Since:**
1.2
Overrides:

setValueExpression in class UIComponent

Parameters:

name - Name of the attribute or property for which to set a ValueExpression
binding - The ValueExpression to set, or null to remove any currently set ValueExpression

Throws:

NullPointerException - if name is null

Since:

1.2

protected boolean compareValues(Object previous, Object value)

true

previous

value

compareValues

protected boolean compareValues(Object previous, Object value)

Return true if the new value is different from the previous value. Value comparison must not be sensitive to element order.

Overrides:

compareValues in class UIInput

Parameters:

previous - old value of this component
value - new value of this component

protected void validateValue(FacesContext context, Object
validateValue

protected void validateValue(FacesContext context, Object value)

In addition to the standard validation behavior inherited from UIInput, ensure that any specified values are equal to one of the available options. Before comparing each option, coerce the option value type to the type of this component's value following the Expression Language coercion rules. If the specified value is not equal to any of the options, enqueue an error message and set the valid property to false.

Overrides:
validateValue in class UIInput

Parameters:
context - The FacesContext for the current request
value - The converted value to test for membership.

Throws:
NullPointerException - if context is null
PS:
javax.faces.component Class UISelectOne

java.lang.Object
  ↓ javax.faces.component.UIComponent
       ↓ javax.faces.component.UIComponentBase
            ↓ javax.faces.component.UIOutput
                 ↓ javax.faces.component.UIInput
                      ↓ javax.faces.component.UISelectOne

All Implemented Interfaces:
  EditableValueHolder, StateHolder, ValueHolder

Direct Known Subclasses:
  HtmlSelectOneListbox, HtmlSelectOneMenu, HtmlSelectOneRadio

public class UISelectOne
  extends UIInput

Extends: UIComponent > UIComponentBase > UIOutput > UIInput
Extended by: HtmlSelectOneListbox, HtmlSelectOneMenu, HtmlSelectOneRadio

UISelectOne 0 1  UIComponent value

  rendererType "javax.faces.Menu"  setRendererType()

UISelectOne is a UIComponent that represents the user's choice of zero or one items from among a discrete set of available options. The user can modify the selected value. Optionally, the component can be preconfigured with a currently selected item, by storing it as the value property of the component.

This component is generally rendered as a select box or a group of radio
buttons.

By default, the `rendererType` property is set to "javax.faces.Menu". This value can be changed by calling the `setRendererType()` method.

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>COMPONENT_FAMILY</code></td>
<td>static String</td>
</tr>
<tr>
<td><code>COMPONENT_TYPE</code></td>
<td>static String</td>
</tr>
<tr>
<td><code>INVALID_MESSAGE_ID</code></td>
<td>static String</td>
</tr>
</tbody>
</table>

### Fields inherited from class `javax.faces.component.UIInput`

- `CONVERSION_MESSAGE_ID`, `REQUIRED_MESSAGE_ID`, `UPDATE_MESSAGE_ID`

### Fields inherited from class `javax.faces.component.UIComponent`

- `bindings`

### Constructor Summary

- `UISelectOne()`  
  Create a new `UISelectOne` instance with default property values.

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getFamily()</code></td>
<td>String</td>
</tr>
<tr>
<td><code>validateValue(FacesContext context, Object value)</code></td>
<td>protected void</td>
</tr>
</tbody>
</table>
the available options.

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UIInput</th>
</tr>
</thead>
<tbody>
<tr>
<td>addValidator, addValueChangeListener, compareValues, decode,</td>
</tr>
<tr>
<td>getConvertedValue, getConverterMessage, getRequiredMessage,</td>
</tr>
<tr>
<td>getSubmittedValue, getValidator, getValidatorMessage,</td>
</tr>
<tr>
<td>getValidators, getValueChangeListener, getValueChangeListeners,</td>
</tr>
<tr>
<td>isImmediate, isLocalValueSet, isRequired, isValid, processDecodes,</td>
</tr>
<tr>
<td>processUpdates, processValidators, removeValidator,</td>
</tr>
<tr>
<td>removeValueChangeListener, resetValue, restoreState, saveState,</td>
</tr>
<tr>
<td>setConverterMessage, setImmediate, setLocalValueSet, setRequired,</td>
</tr>
<tr>
<td>setRequiredMessage, setSubmittedValue, setValue, setValueChangeListener,</td>
</tr>
<tr>
<td>updateModel, validate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UIOutput</th>
</tr>
</thead>
<tbody>
<tr>
<td>getConverter, getLocalValue, getValue, setConverter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UIComponentBase</th>
</tr>
</thead>
<tbody>
<tr>
<td>addFacesListener, broadcast, encodeBegin, encodeChildren,</td>
</tr>
<tr>
<td>encodeEnd, findComponent, getAttributes, getChildCount,</td>
</tr>
<tr>
<td>getChildren, getClientId, getFacesContext, getFacesListeners,</td>
</tr>
<tr>
<td>getFacet, getFacetCount, getFacets, getFacetsAndChildren, getId,</td>
</tr>
<tr>
<td>getParent, getRenderer, getRendererType, getRendersChildren,</td>
</tr>
<tr>
<td>getValueBinding, invokeOnComponent, isRendered, isTransient,</td>
</tr>
<tr>
<td>processRestoreState, processSaveState, queueEvent,</td>
</tr>
<tr>
<td>removeFacesListener, restoreAttachedState, saveAttachedState,</td>
</tr>
<tr>
<td>setId, setParent, setRendered, setRendererType, setTransient,</td>
</tr>
<tr>
<td>setValueBinding</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class javax.faces.component.UIComponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>encodeAll, getContainerClientId, getValueExpression,</td>
</tr>
<tr>
<td>setValueExpression</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from class java.lang.Object</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll,</td>
</tr>
<tr>
<td>toString, wait, wait, wait</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Methods inherited from interface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
javax.faces.component.ValueHolder
getConverter, getLocalValue, getValue, setConverter

Field Detail

COMPONENT_TYPE

public static final String COMPONENT_TYPE

The standard component type for this component.

See Also:
Constant Field Values

COMPONENT_FAMILY

public static final String COMPONENT_FAMILY

The standard component family for this component.

See Also:
Constant Field Values

INVALID_MESSAGE_ID

public static final String INVALID_MESSAGE_ID

The message identifier of the FacesMessage to be created if a value not matching the available options is specified.
See Also:
Constant Field Values

## Constructor Detail

```java
public UISelectOne()
```

**UISelectOne**

```java
public UISelectOne()
```

Create a new **UISelectOne** instance with default property values.

## Method Detail

```java
public String getFamily()
```

**getFamily**

```java
public String getFamily()
```

**Description copied from class: UIComponent**

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

**Overrides:**
```
getFamily in class UIInput
```
protected void validateValue(FacesContext context, Object value)

UIInput Expression valid false

custom context FacesContext
value
Throws NullPointerException: context null

validateValue

protected void validateValue(FacesContext context, Object value)

In addition to the standard validation behavior inherited from UIInput, ensure that any specified value is equal to one of the available options. Before comparing each option, coerce the option value type to the type of this component's value following the Expression Language coercion rules. If the specified value is not equal to any of the options, enqueue an error message and set the valid property to false.

Overrides:
validateValue in class UIInput

Parameters:
context - The FacesContext for the current request
value - The converted value to test for membership.

Throws:
NullPointerException - if context is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject
to license terms.

PS:
javax.faces.component  **Class UIViewRoot**

java.lang.Object
   └─javax.faces.component.UIComponent
       └─javax.faces.component.UIComponentBase
           └─javax.faces.component.UIViewRoot

**All Implemented Interfaces:**
   StateHolder

```java
public class UIViewRoot
    extends UIComponentBase

Extends: UIComponent > UIComponentBase

UIViewRoot UIComponent UIComponent
PhaseListener
```

- #processDecodes
- #processValidators
- #processUpdates
- #processApplication
- RenderResponse  #encodeBegin  #encodeEnd

PhaseListener

false

#getBeforePhaseListener   null   PhaseId

FacesContext#getResponseComplete
UIViewRoot is the UIComponent that represents the root of the UIComponent tree. This component has no rendering, it just serves as the root of the component tree, and as a place to hang per-view PhaseListenerS.

For each of the following lifecycle phase methods:

- `processDecodes(javax.faces.context.FacesContext)`
- `processValidators(javax.faces.context.FacesContext)`
- `processUpdates(javax.faces.context.FacesContext)`
- `processApplication(javax.faces.context.FacesContext)`
- RenderResponse, via
  - `encodeBegin(javax.faces.context.FacesContext)` and
  - `encodeEnd(javax.faces.context.FacesContext)`

Take the following action regarding PhaseListenerS.

Initialize a state flag to `false`.

If `getBeforePhaseListener()` returns non-null, invoke the listener,
passing in the correct corresponding PhaseId for this phase.

Upon return from the listener, call 
FacesContext.getResponseComplete() and 
FacesContext.getRenderResponse(). If either return true set the internal state flag to true.

If or one or more listeners have been added by a call to 
addPhaseListener(javax.faces.event.PhaseListener), invoke the beforePhase method on each one whose PhaseListener.getPhaseId() matches the current phaseId, passing in the same PhaseId as in the previous step.

Upon return from each listener, call 
FacesContext.getResponseComplete() and 
FacesContext.getRenderResponse(). If either return true set the internal state flag to true.

Execute any processing for this phase if the internal state flag was not set.

If getAfterPhaseListener() returns non-null, invoke the listener, passing in the correct corresponding PhaseId for this phase.

If or one or more listeners have been added by a call to 
addPhaseListener(javax.faces.event.PhaseListener), invoke the afterPhase method on each one whose PhaseListener.getPhaseId() matches the current phaseId, passing in the same PhaseId as in the previous step.

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>COMPONENT_FAMILY</td>
</tr>
<tr>
<td>String</td>
<td>The standard component family name for this component.</td>
</tr>
<tr>
<td>static</td>
<td>COMPONENT_TYPE</td>
</tr>
<tr>
<td>String</td>
<td>The standard component type name for this component.</td>
</tr>
<tr>
<td>static</td>
<td>UNIQUE_ID_PREFIX</td>
</tr>
<tr>
<td>String</td>
<td>The prefix that will be used for identifiers generated by</td>
</tr>
</tbody>
</table>
the `createUniqueId()` method.

Fields inherited from class `javax.faces.component.UIComponent` bindings

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UIViewRoot()</code></td>
<td>Create a new <code>UIViewRoot</code> instance with default property values.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void addPhaseListener(PhaseListener newPhaseListener)</code></td>
<td></td>
</tr>
<tr>
<td><code>String createUniqueId()</code></td>
<td>Generate an identifier for a component.</td>
</tr>
<tr>
<td><code>void encodeBegin(FacesContext context)</code></td>
<td>Override the default <code>UIComponentBase.encodeBegin(javax.faces.context.FacesContext)</code> behavior.</td>
</tr>
<tr>
<td><code>void encodeEnd(FacesContext context)</code></td>
<td>Override the default <code>UIComponentBase.encodeEnd(javax.faces.context.FacesContext)</code> behavior.</td>
</tr>
<tr>
<td><code>MethodExpression getAfterPhaseListener()</code></td>
<td></td>
</tr>
<tr>
<td><code>MethodExpression getBeforePhaseListener()</code></td>
<td></td>
</tr>
<tr>
<td><code>String getFamily()</code></td>
<td>Return the identifier of the component family to which this component belongs.</td>
</tr>
<tr>
<td><code>Locale getLocale()</code></td>
<td>Return the Locale to be used in localizing the response being created for this view.</td>
</tr>
<tr>
<td><code>String getRenderKitId()</code></td>
<td>Return the render kit identifier of the <code>RenderKit</code> associated...</td>
</tr>
</tbody>
</table>
**getViewId()**

Return the view identifier for this view.

**processApplication(FacesContext context)**

Broadcast any events that have been queued for the Application phase of the request processing lifecycle and to any events for later phases if the event processing for this FacesContext.renderResponse() or FacesContext.responseComplete() be called.

**processDecodes(FacesContext context)**

Override the default UIComponentBase.processDecodes(javax.faces.context.FacesContext) behavior to broadcast any queued events after the default has been completed and to clear out any events for later phases if the event processing for this phase caused FacesContext.renderResponse() or FacesContext.responseComplete() to be called.

**processUpdates(FacesContext context)**

Override the default UIComponentBase behavior to broadcast queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused FacesContext.renderResponse() or FacesContext.responseComplete() to be called.

**processValidators(FacesContext context)**

Override the default UIComponentBase.processValidators(javax.faces.context.FacesContext) behavior to broadcast any queued events after the default has been completed and to clear out any events for later phases if the event processing for this phase caused FacesContext.renderResponse() or FacesContext.responseComplete() to be called.

**queueEvent(FacesEvent event)**

Override the default UIComponentBase.queueEvent(javax.faces.event.FacesEvent) accumulate the queued events for later broadcasting.

**removePhaseListener(PhaseListener toRemove)**

**restoreState(FacesContext context, Object state)**

Perform any processing required to restore the state.
entries in the state Object.

```java
Object saveState(FacesContext context)
    Gets the state of the instance as a Serializable Ob
```

```java
void setAfterPhaseListener(MethodExpression newAfterPhase)
    Allow an arbitrary method to be called for the "afterP
as the UIViewRoot runs through its lifecycle.
```

```java
void setBeforePhaseListener(MethodExpression newBeforePhase
    Allow an arbitrary method to be called for the "before
as the UIViewRoot runs through its lifecycle.
```

```java
void setLocale(Locale locale)
    Set the Locale to be used in localizing the response
for this view.
```

```java
void setRenderKitId(String renderKitId)
    Set the render kit identifier of the RenderKit associated
view.
```

```java
void setViewId(String viewId)
    Set the view identifier for this view.
```

Methods inherited from class
javax.faces.component.UIComponentBase
addFacesListener, broadcast, decode, encodeChildren,
findComponent, getAttributes, getChildCount, getChildren,
getClientId, getFacesContext, getFacesListeners, getFacet,
getFacetCount, getFacets, getFacetsAndChildren, getId, getParent,
gerenderer, getRendererType, getRendersChildren, getValueBinding,
invokeOnComponent, isRendered, isTransient, processRestoreState,
processSaveState, removeFacesListener, restoreAttachedState,
saveAttachedState, setId, setParent, setRendered, setRendererType,
setTransient, setValueBinding
```

Methods inherited from class
javax.faces.component.UIComponent
encodeAll, getContainerClientId, getValueExpression,
setValueExpression
```

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
### COMPONENT_TYPE

```java
public static final String COMPONENT_TYPE
```

The standard component type for this component.

**See Also:**
- [Constant Field Values](#)

### COMPONENT_FAMILY

```java
public static final String COMPONENT_FAMILY
```

The standard component family for this component.

**See Also:**
- [Constant Field Values](#)

### UNIQUE_ID_PREFIX

```java
public static final String UNIQUE_ID_PREFIX
```

The prefix that will be used for identifiers generated by the `createUniqueId()` method.

**See Also:**
- [Constant Field Values](#)
Constructor Detail

public UIViewRoot()

UIViewRoot

public UIViewRoot()

Create a new UIViewRoot instance with default property values.

Method Detail

public String getFamily()

getFamily

public String getFamily()

Description copied from class: UIComponent

Return the identifier of the component family to which this component belongs. This identifier, in conjunction with the value of the rendererType property, may be used to select the appropriate Renderer for this component instance.

Specified by:

getchFamily in class UIComponent

public String getRenderKitId()
public String getViewId()
getViewById

public String getViewId()

Return the view identifier for this view.

public void setViewId(String viewId)


setViewById

public void setViewId(String viewId)

Set the view identifier for this view.

Parameters:

viewId - The new view identifier

getBeforePhaseListener

public MethodExpression getBeforePhaseListener()

return

getBeforePhaseListener

public MethodExpression getBeforePhaseListener()

Returns:

the MethodExpression that will be invoked before this view is
public void setBeforePhaseListener(MethodExpression newBeforePhase)

UIViewRoot "beforePhase"

PhaseId#RESTORE_VIEW PhaseListener PhaseEvent

PhaseListener#beforePhase

newBeforePhase

MethodExpression

setBeforePhaseListener

public void setBeforePhaseListener(MethodExpression newBeforePhase)

Allow an arbitrary method to be called for the "beforePhase" event as the UIViewRoot runs through its lifecycle. This method will be called for all phases except PhaseId.RESTORE_VIEW. Unlike a true PhaseListener, this approach doesn't allow for only receiving PhaseEventS for a given phase.

The method must conform to the signature of PhaseListener.beforePhase(javax.faces.event.PhaseEvent).

Parameters:

newBeforePhase - the MethodExpression that will be invoked before this view is rendered.

public MethodExpression getAfterPhaseListener()

return

MethodExpression

getAfterPhaseListener
public MethodExpression getAfterPhaseListener()

    Returns:
    the MethodExpression that will be invoked after this view is rendered.

public void setAfterPhaseListener(MethodExpression newAfterPhase)

    UIViewRoot "afterPhase"
    PhaseId#RESTORE_VIEW PhaseListener PhaseEvent

        PhaseListener#afterPhase

            newAfterPhase MethodExpression

setAfterPhaseListener

public void setAfterPhaseListener(MethodExpression newAfterPhase)

    Allow an arbitrary method to be called for the "afterPhase" event as
    the UIViewRoot runs through its lifecycle. This method will be called
    for all phases except PhaseId.RESTORE_VIEW. Unlike a true
    PhaseListener, this approach doesn't allow for only receiving
    PhaseEventS for a given phase.

    The method must conform to the signature of
    PhaseListener.afterPhase(javax.faces.event.PhaseEvent).

    Parameters:
    newAfterPhase - the MethodExpression that will be invoked after
    this view is rendered.

public void removePhaseListener(PhaseListener toRemove)
removePhaseListener

public void removePhaseListener(PhaseListener toRemove)

addPhaseListener

public void addPhaseListener(PhaseListener newPhaseListener)

queueEvent

public void queueEvent(FacesEvent event)

UIComponentBase#queueEvent

event FacesEvent
Throws IllegalStateException: UIViewRoot
Throws NullPointerException: event null

queueEvent

public void queueEvent(FacesEvent event)

Override the default
UIComponentBase.queueEvent(javax.faces.event.FacesEvent) behavior to accumulate the queued events for later broadcasting.

Overrides:
queueEvent in class UIComponentBase

Parameters:
event - FacesEvent to be queued

Throws:
IllegalStateException - if this component is not a descendant of a UIViewRoot
NullPointerException - if event is null

public void processDecodes(FacesContext context)

FacesContext#renderResponse
FacesContext#responseComplete
UIComponentBase#processDecodes

context
Throws

FacesContext

Throws

NullPointerException: context null

processDecodes

public void processDecodes(FacesContext context)

Override the default UIComponentBase.processDecodes(javax.faces.context.FacesContext) behavior to broadcast any queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused FacesContext.renderResponse() or FacesContext.responseComplete() to be called.

Overrides:
processDecodes in class UIComponentBase

Parameters:
context - FacesContext for the request we are processing

Throws:

NullPointerException - if context is null

public void encodeBegin(FacesContext context) throws java.io.IOException
encodeBegin

public void encodeBegin(FacesContext context) throws IOException

    Override the default UICOMPONENTBASE.encodeBegin (javax.faces.context.FacesContext) behavior. If getBeforePhaseListener () returns non-null, invoke it, passing a PhaseEvent for the PhaseId.RENDER_RESPONSE phase. If the internal list populated by calls to addPhaseListener (javax.faces.event.PhaseListener) is non-empty, any listeners in that list must have their PhaseListener.beforePhase (javax.faces.event.PhaseEvent) method called, passing the PhaseEvent. Any errors that occur during invocation of any of the the beforePhase listeners must be logged and swallowed. After listeners are invoked call superclass processing.

    Overrides: encodeBegin in class UICOMPONENTBASE

    Parameters: 
        context - FacesContext for the response we are creating

    Throws: 
        IOException - if an input/output error occurs while rendering

public void encodeEnd(FacesContext context) throws java.io.IOException

    OVERRIDE UICOMPONENTBASE.encodeEnd (javax.faces.context.FacesContext)
afterPhase

encodeEnd

public void encodeEnd(FacesContext context)
    throws IOException

  Override the default UIComponentBase.encodeEnd(javax.faces.context.FacesContext) behavior. If getAfterPhaseListener() returns non-null, invoke it, passing a PhaseEvent for the PhaseId.RENDER_RESPONSE phase. Any errors that occur during invocation of the afterPhase listener must be logged and swallowed.

  Overrides:
  encodeEnd in class UIComponentBase

  Parameters:
  context - FacesContext for the response we are creating

  Throws:
  IOException - if an input/output error occurs while rendering

public void processValidators(FacesContext context)

  FacesContext#renderResponse
  FacesContext#responseComplete
  UIComponentBase#processValidators

  context - FacesContext
  Throws
  NullPointerException: context null

processValidators

public void processValidators(FacesContext context)
Override the default `UIComponentBase.processValidators(javax.faces.context.FacesContext)` behavior to broadcast any queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused `FacesContext.renderResponse()` or `FacesContext.responseComplete()` to be called.

Overrides:
- `processValidators` in class `UIComponentBase`

Parameters:
- `context` - `FacesContext` for the request we are processing

Throws:
- `NullPointerException` - if `context` is `null`

---

```java
public void processUpdates(FacesContext context)
```

```java
FacesContext#renderResponse
FacesContext#responseComplete
UIComponentBase
```

- `context` - `FacesContext` for the request
- Throws `NullPointerException`: `context` is `null`

---

processUpdates

```java
public void processUpdates(FacesContext context)
```

Override the default `UIComponentBase` behavior to broadcast any queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused `FacesContext.renderResponse()` or `FacesContext.responseComplete()` to be called.

Overrides:
- `processUpdates` in class `UIComponentBase`

Parameters:
public void processApplication(FacesContext context)

Broadcast any events that have been queued for the *Invoke Application* phase of the request processing lifecycle and to clear out any events for later phases if the event processing for this phase caused `FacesContext.renderResponse()` or `FacesContext.responseComplete()` to be called.

Parameters:
context - *FacesContext* for the request we are processing

Throws:
`NullPointException` - if context is null

public String createUniqueId()

*UNIQUE_ID_PREFIX* UIViewRoot

createUniqueId
public String createUniqueId()

Generate an identifier for a component. The identifier will be prefixed with UNIQUE_ID_PREFIX, and will be unique within this UIViewRoot.

public java.util.Locale getLocale()

Locale

locale ivar "locale" null
javax.faces.application.ViewHandler#calculateLocale
java.util.Locale String java.util.Locale "locale"
javax.faces.application.ViewHandler#calculateLocale

return Locale

getLocale

public Locale getLocale()

Return the Locale to be used in localizing the response being created for this view.

Algorithm:

If we have a locale ivar, return it. If we have a value expression for "locale", get its value. If the value is null, return the result of calling ViewHandler.calculateLocale(javax.faces.context.FacesContext). If the value is an instance of java.util.Locale return it. If the value is a String, convert it to a java.util.Locale and return it. If there is no value expression for "locale", return the result of calling ViewHandler.calculateLocale(javax.faces.context.FacesContext).
Returns:
The current `Locale` obtained by executing the above algorithm.

```java
public void setLocale(java.util.Locale locale)

Locale
locale
locale
Locale

setLocale

public void setLocale(Locale locale)

Set the `Locale` to be used in localizing the response being created for this view.

Parameters:
locale - The new localization `Locale`

public Object saveState(FacesContext context)

saveState

public Object saveState(FacesContext context)

Description copied from interface: `StateHolder`

Gets the state of the instance as a `Serializable` Object.

If the class that implements this interface has references to instances that implement StateHolder (such as a `UIComponent` with event handlers, validators, etc.) this method must call the `StateHolder.saveState(javax.faces.context.FacesContext) method on all those instances as well. This method must not save the
**state of children and facets.** That is done via the `StateManager`.

This method must not alter the state of the implementing object. In other words, after executing this code:

```java
Object state = component.saveState(facesContext);
```

`component` should be the same as before executing it.

The return from this method must be `Serializable`.

**Specified by:**
`saveState` in interface `StateHolder`  
**Overrides:**
`saveState` in class `UIComponentBase`

---

**public void restoreState(FacesContext context, Object state)**

**restoreState**

```java
public void restoreState(FacesContext context, Object state)
```

**Description copied from interface: StateHolder**

Perform any processing required to restore the state from the entries in the state Object.

If the class that implements this interface has references to instances that also implement StateHolder (such as a UIComponent with event handlers, validators, etc.) this method must call the `StateHolder.restoreState(javax.faces.context.FacesContext, java.lang.Object)` method on all those instances as well.

**Specified by:**
`restoreState` in interface `StateHolder`
Overrides:

restoreState in class UICOMPONENTBASE

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi  Class UnavailableException

java.lang.Object  
   java.lang.Throwable
      java.lang.Exception
         javax.resource.ResourceException
            javax.resource.spi.UnavailableException

All Implemented Interfaces:
   Serializable

public class UnavailableException

extends  ResourceException

Extends: Throwable > Exception > ResourceException

version 1.0

This is thrown to indicate that a service is unavailable.

Version: 1.0

Author:  Ram Jeyaraman

See Also:  Serialized Form

---

**Constructor Summary**

-UnavailableException()  
   Constructs a new instance with null as its detail message.

-UnavailableException(String message)  
   Constructs a new instance with the specified detail message.
### UnavailableException

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UnavailableException(String message, String errorCode)</code></td>
<td>Constructs a new throwable with the specified detail message and an error code.</td>
</tr>
<tr>
<td><code>UnavailableException(String message, Throwable cause)</code></td>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
<tr>
<td><code>UnavailableException(Throwable cause)</code></td>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>

### Method Summary

#### Methods inherited from class javax.resource.ResourceException

- `getErrorCode`, `getLinkedException`, `getMessage`, `setErrorCode`, `setLinkedException`

#### Methods inherited from class java.lang.Throwable

- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

#### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

### Constructor Detail

- `public UnavailableException()`
  `null`

---

**UnavailableException**

- `public UnavailableException()`
Constructs a new instance with null as its detail message.

```java
public UnavailableException(String message)
```

*message*

**UnavailableException**

```java
public UnavailableException(String message)
```

Constructs a new instance with the specified detail message.

**Parameters:**

*message* - the detail message.

```java
public UnavailableException((Throwable cause)
```

*cause* throwable

```java
public UnavailableException(Throwable cause)
```

Constructs a new throwable with the specified cause.

**Parameters:**

*cause* - a chained exception of type Throwable.

```java
public UnavailableException(String message, Throwable cause)
```

*cause* throwable

```java
public UnavailableException(String message, Throwable cause)
```

*message*
UnavailableException

public UnavailableException(String message, Throwable cause)

Constructs a new Throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public UnavailableException(String message, String errorCode)

Constructs a new Throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.servlet Class UnavailableException

java.lang.Object
   ^ java.lang.Throwable
      ^ java.lang.Exception
         ^ javax.servlet.ServletException
            ^ javax.servlet.UnavailableException

All Implemented Interfaces:
   Serializable

public class UnavailableException

extends ServletException

Extends: Throwable > Exception > ServletException

servlet servlet

servlet servlet servlet servlet

servlet servlet

servlet servlet servlet servlet servlet

Defines an exception that a servlet or filter throws to indicate that it is permanently or temporarily unavailable.

When a servlet or filter is permanently unavailable, something is wrong with it, and it cannot handle requests until some action is taken. For example, a servlet might be configured incorrectly, or a filter's state may be corrupted. The component should log both the error and the corrective action that is needed.

A servlet or filter is temporarily unavailable if it cannot handle requests momentarily due to some system-wide problem. For example, a third-tier
server might not be accessible, or there may be insufficient memory or disk storage to handle requests. A system administrator may need to take corrective action.

Servlet containers can safely treat both types of unavailable exceptions in the same way. However, treating temporary unavailability effectively makes the servlet container more robust. Specifically, the servlet container might block requests to the servlet or filter for a period of time suggested by the exception, rather than rejecting them until the servlet container restarts.

Author:
Various
See Also:
Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>UnavailableException(int seconds, Servlet servlet, String msg)</code></td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.2, use <code>UnavailableException(String, int)</code> instead.</td>
</tr>
<tr>
<td><code>UnavailableException(Servlet servlet, String msg)</code></td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.2, use <code>UnavailableException(String)</code> instead.</td>
</tr>
<tr>
<td><code>UnavailableException(String msg)</code></td>
<td>Constructs a new exception with a descriptive message indicating that the servlet is permanently unavailable.</td>
</tr>
<tr>
<td><code>UnavailableException(String msg, int seconds)</code></td>
<td>Constructs a new exception with a descriptive message indicating that the servlet is temporarily unavailable and giving an estimate of how long it will be unavailable.</td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getServlet()</code></td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.2, with no replacement. Returns the servlet that is reporting its</td>
</tr>
</tbody>
</table>
**UnavailableException**

public UnavailableException(Servlet servlet, String msg)

**Deprecation:**
Java Servlet API 2.2

**UnavailableException(String)**

Servlet
String

Deprecated. As of Java Servlet API 2.2, use UnavailableException(String) instead.
public UnavailableException(int seconds, Servlet servlet, String msg)

Deprecated. As of Java Servlet API 2.2, use UnavailableException(String, int) instead.

Parameters:
seconds - an integer specifying the number of seconds the servlet expects to be unavailable; if zero or negative, indicates that the servlet can't make an estimate
servlet - the Servlet that is unavailable
msg - a String specifying the descriptive message, which can be written to a log file or displayed for the user.
public UnavailableException(String msg)

Constructs a new exception with a descriptive message indicating that the servlet is permanently unavailable.

Parameters:
  msg - a String specifying the descriptive message

public UnavailableException(String msg, int seconds)

Constructs a new exception with a descriptive message indicating that the servlet is temporarily unavailable and giving an estimate of how long it will be unavailable.

In some cases, the servlet cannot make an estimate. For example, the servlet might know that a server it needs is not running, but not be able to report how long it will take to be restored to functionality. This can be indicated with a negative or zero value for the seconds argument.

Parameters:
  msg - a String specifying the descriptive message, which can be written to a log file or displayed for the user.
  seconds - an integer specifying the number of seconds the servlet expects to be unavailable; if zero or negative, indicates that the servlet can't make an estimate
public boolean isPermanent()

Returns a boolean indicating whether the servlet is permanently unavailable. If so, something is wrong with the servlet, and the system administrator must take some corrective action.

Returns:
true if the servlet is permanently unavailable; false if the servlet is available or temporarily unavailable

getServlet

public Servlet getServlet()

Deprecated. As of Java Servlet API 2.2, with no replacement. Returns the servlet that is reporting its unavailability.

Returns:
the Servlet object that is throwing the UnavailableException
getUnavailableSeconds

```
public int getUnavailableSeconds()
```

Returns the number of seconds the servlet expects to be temporarily unavailable.

If this method returns a negative number, the servlet is permanently unavailable or cannot provide an estimate of how long it will be unavailable. No effort is made to correct for the time elapsed since the exception was first reported.

**Returns:**

an integer specifying the number of seconds the servlet will be temporarily unavailable, or a negative number if the servlet is permanently unavailable or cannot make an estimate.
javax.xml.registry Class UnexpectedObjectException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.xml.registry.JAXRException
              └ javax.xml.registry.UnexpectedObjectException

All Implemented Interfaces:
    Serializable, JAXRResponse

public class UnexpectedObjectException
  extends JAXRException

Extends: Throwable > Exception > JAXRException

JAXR Object BusinessLifeCycleManager
saveOrganizations Object Collection Exception

This exception is thrown when the JAXR provider finds a Object that is out-of-place or of the wrong type within the context of a user request. For example a saveOrganizations request in BusinessLifeCycleManager would throw this Exception if it found an Object other than Organization in the Collection of Objects provided by client to be saved by the request.

Author:
    Farrukh S. Najmi

See Also:
    Serialized Form

Field Summary

Fields inherited from class javax.xml.registry.JAXRException
cause
Fields inherited from interface javax.xml.registry.JAXRResponse
STATUS_FAILURE, STATUS_SUCCESS, STATUS_UNAVAILABLE, STATUS_WARNING

Constructor Summary

UnexpectedObjectException()
Constructs a JAXRException object with no reason or embedded Throwable.

UnexpectedObjectException(String reason)
Constructs a JAXRException object with the given String as the reason for the exception being thrown.

UnexpectedObjectException(String reason, Throwable cause)
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

UnexpectedObjectException(Throwable cause)
Constructs a JAXRException object initialized with the given Throwable object.

Method Summary

Methods inherited from class javax.xml.registry.JAXRException
getCause, getMessage, getRequestID, getStatus, initCause, isAvailable

Methods inherited from class java.lang.Throwable
fillInStackTrace, getLocalizedMessage, getStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait
public UnexpectedObjectException()
	Throwable JAXRException

**UnexpectedObjectException**

public UnexpectedObjectException()

Constructs a JAXRException object with no reason or embedded Throwable.

---

public UnexpectedObjectException(String reason)

**JAXRException** **String** reason

**UnexpectedObjectException**

public UnexpectedObjectException(String reason)

Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**Parameters:**

- **reason** - a description of what caused the exception

---

public UnexpectedObjectException(String reason, Throwable cause)

**JAXRException** **String** Throwable

**Throwable**

- **reason**
- **cause** 

---

**JAXRException** **Throwable**
public UnexpectedObjectException(String reason, Throwable cause)

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
  reason - a description of what caused the exception
  cause - a Throwable object that is to be embedded in this JAXRException object

public UnexpectedObjectException(Throwable cause)

Constructs a JAXRException object initialized with the given Throwable object.

Parameters:
  cause - the Throwable that caused this Exception
to license terms.

PS:
javax.persistence  **Annotation Type UniqueConstraint**  

@Target(value={})  
@Retention(value=RUNTIME)  
public @interface UniqueConstraint  

**Implements:** Annotation  
@Target(value={})  
@Retention(value=RUNTIME)  

**DDL**  

```java  
@Entity  
@Table(  
    name="EMPLOYEE",  
    uniqueConstraints=  
        @UniqueConstraint(columnNames={"EMP_ID", "EMP_NAME"})  
)  
public class Employee { ... }  
```

**since**  
Java Persistence 1.0  

This annotation is used to specify that a unique constraint is to be included in the generated DDL for a primary or secondary table.  

**Example:**  
@Target  
@Retention  
public class Employee { ... }  

**Since:**  
Java Persistence 1.0
Required Element Summary

<table>
<thead>
<tr>
<th>String[] columnNames</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Required) An array of the column names that make up the constraint.</td>
</tr>
</tbody>
</table>

Element Detail

abstract public String[] columnNames()

columnNames

public abstract String[] columnNames

(Required) An array of the column names that make up the constraint.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind Class UnmarshalException

java.lang.Object
├ java.lang.Throwable
│  └ java.lang.Exception
│     └ javax.xml.bind.JAXBException
│        └ javax.xml.bind.UnmarshalException

All Implemented Interfaces:
    Serializable

public class UnmarshalException
    extends JAXBException

Extends: Throwable > Exception > JAXBException

JAXB

ValidationEventHandler
    ValidationEventHandler.handleEvent(ValidationEvent)

version $Revision: 1.1 $
since JAXB1.0
See also
    javax.xml.bind.JAXBException,
    javax.xml.bind.Unmarshaller,
    javax.xml.bind.ValidationEventHandler

This exception indicates that an error has occurred while performing an unmarshal operation that prevents the JAXB Provider from completing the operation.

The ValidationEventHandler can cause this exception to be thrown during the unmarshal operations. See
    ValidationEventHandler.handleEvent(ValidationEvent).

Since:
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnmarshalException(String message)</td>
<td>Construct an UnmarshalException with the specified detail message.</td>
</tr>
<tr>
<td>UnmarshalException(String message, String errorCode)</td>
<td>Construct an UnmarshalException with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td>UnmarshalException(String message, String errorCode, Throwable exception)</td>
<td>Construct an UnmarshalException with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td>UnmarshalException(String message, Throwable exception)</td>
<td>Construct an UnmarshalException with the specified detail message and linkedException.</td>
</tr>
<tr>
<td>UnmarshalException(Throwable exception)</td>
<td>Construct an UnmarshalException with a linkedException.</td>
</tr>
</tbody>
</table>

### Method Summary

Methods inherited from class javax.xml.bind.JAXBException

getCause, getErrorCode, getLinkedException, printStackTrace, printStackTrace, printStackTrace, setLinkedException, toString

Methods inherited from class java.lang.Throwable

fillInStackTrace, getLocalizedMessage, getMessage, getStackTrace,
Methods inherited from class java.lang.\texttt{Object}
\texttt{clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait}

\textbf{Constructor Detail}

\begin{verbatim}
public UnmarshalException(String message)
 UnmarshalException errorCode  linkedException
 null

   message

UnmarshalException

public UnmarshalException(\texttt{String} message)

    Construct an UnmarshalException with the specified detail message. The errorCode and linkedException will default to null.

    Parameters:
    message - a description of the exception

public UnmarshalException(String message, String errorCode)
    errorCode  UnmarshalException linkedException
 null

   message
   errorCode

UnmarshalException
\end{verbatim}
public UnmarshalException(String message,
                         String errorCode)

        Construct an UnmarshalException with the specified detail message
        and vendor specific errorCode. The linkedException will default to
        null.

        Parameters:
            message - a description of the exception
            errorCode - a string specifying the vendor specific error code

public UnmarshalException(Throwable exception)
        linkedException UnmarshalException errorCode
        null

        exception

UnmarshalException

public UnmarshalException(Throwable exception)

        Construct an UnmarshalException with a linkedException. The detail
        message and vendor specific errorCode will default to null.

        Parameters:
            exception - the linked exception

public UnmarshalException(String message, Throwable exception)
        linkedException UnmarshalException errorCode
        null

        message
        exception
UnmarshalException

public UnmarshalException(String message, Throwable exception)

Construct an UnmarshalException with the specified detail message and linkedException. The errorCode will default to null.

Parameters:
message - a description of the exception
exception - the linked exception

public UnmarshalException(String message, String errorCode, Throwable exception)
errorCode  linkedException
UnmarshalException
message
errorCode
exception

UnmarshalException

public UnmarshalException(String message, String errorCode, Throwable exception)

Construct an UnmarshalException with the specified detail message, vendor specific errorCode, and linkedException.

Parameters:
message - a description of the exception
errorCode - a string specifying the vendor specific error code
exception - the linked exception
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS : 
javax.xml.bind  Interface Unmarshaller

All Known Implementing Classes:
   AbstractUnmarshallerImpl

public interface Unmarshaller

Inner classes: Unmarshaller.Listener
Implemented by: AbstractUnmarshallerImpl

Unmarshaller  XML  Java  XML  unmarshal

File

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
Object o = u.unmarshal( new File("nosferatu.xml") );

InputStream

InputStream is = new FileInputStream("nosferatu.xml");
JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();
Object o = u.unmarshal(is);

URL

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
URL url = new URL("http://beaker.east/nosferatu.xml");
Object o = u.unmarshal(url);

javax.xml.transform.stream.StreamSource  StringBuffer

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
StringBuffer xmlStr = new StringBuffer("<?xml version="1.0"?>..."
Object o = u.unmarshal(new StreamSource(new StringReader(xmlStr.toString())));
org.w3c.dom.Node

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

Unmarshaller u = jc.createUnmarshaller();
Unmarshaller u = jc.createUnmarshaller();

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.parse(new File("nosferatu.xml"));

Object o = u.unmarshal(doc);

SAX2.0

javax.xml.transform.sax.SAXSource

// configure a validating SAX2.0 parser (Xerces2)
static final String JAXP_SCHEMA_LANGUAGE = "http://java.sun.com/xml/jaxp/properties/schemaLanguage";
static final String JAXP_SCHEMA_LOCATION = "http://java.sun.com/xml/jaxp/properties/schemaSource";
static final String W3C_XML_SCHEMA = "http://www.w3.org/2001/XMLSchema";

System.setProperty("javax.xml.parsers.SAXParserFactory",
"org.apache.xerces.jaxp.SAXParserFactoryImpl");

SAXParserFactory spf = SAXParserFactory.newInstance();
spf.setNamespaceAware(true);
spf.setValidating(true);
SAXParser saxParser = spf.newSAXParser();
try {
saxParser.setProperty(JAXP_SCHEMA_LANGUAGE, W3C_XML_SCHEMA);
saxParser.setProperty(JAXP_SCHEMA_LOCATION, "http://....");
} catch (SAXNotRecognizedException x) {
    // exception handling omitted
}

XMLReader xmlReader = saxParser.getXMLReader();
SAXSource source = new SAXSource( xmlReader, new InputSource("http://..."));

// Setup JAXB to unmarshal
JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();
ValidationEventCollector vec = new ValidationEventCollector();
u.setEventHandler(vec);

// turn off the JAXB provider's default validation mechanism to
// avoid duplicate validation
u.setValidating( false )

// unmarshal
Object o = u.unmarshal( source );

// check for events
if( vec.hasEvents() ) {
    // iterate over events
}

StAX XMLStreamReader

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

javax.xml.stream.XMLStreamReader xmlStreamReader =
javax.xml.stream.XMLInputFactory().newInstance().createXMLStreamReader( ... );

Object o = u.unmarshal( xmlStreamReader );

StAX XMLEventReader

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

javax.xml.stream.XMLEventReader xmlEventReader =
javax.xml.stream.XMLInputFactory().newInstance().createXMLEventReader( ... );

Object o = u.unmarshal( xmlEventReader );

XML

unmarshal XML

JAXBContext unmarshal XML

JAXBContext XML

unmarshal null XML JAXB JAXBException

declaredType unmarshal JAXBContext XML
JAXBContext Unmarshaller JAXBContext XML
JAXB unmarshal JAXB JAXB XML
JAXBElement XML JAXBContext @xsi:type
UnmarshalException

declaredType unmarshal XML JAXBContext XML
unmarshaller declaredType JAXBContext

JAXBContext JAXB xsi:type unmarshaller

JAXBElement JAXBElemnt<declaredType>

<table>
<thead>
<tr>
<th>JAXBElemnt</th>
<th>JAXBElemnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
<td>xml</td>
</tr>
<tr>
<td>value</td>
<td>instanceof declaredType</td>
</tr>
<tr>
<td>declaredType</td>
<td>unmarshal declaredType</td>
</tr>
<tr>
<td></td>
<td>null</td>
</tr>
</tbody>
</table>

declaredType org.w3c.dom.Node

Schema fragment for example
<xs:schema>
<xs:complexType name="FooType">...
<xs:element name="foo" type="FooType"/>
...
</xs:complexType>
</xs:schema>
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.parse(new File( "nosferatu.xml" ));
Element fooSubtree = ...; // traverse DOM till reach xml element
// local element declaration in schema.

// FooType is the JAXB mapping of the type of local element declar
JAXBElement<FooType> foo = u.unmarshal( fooSubtree, FooType.class)

SAX2.0

SAX2.0 SAX JAXB JAXB SAX2.0
SAX2.0 SAX2.0

setSchema(javax.xml.validation.Schema) API JAXF
SAX 2.0 unmarshal(Source) API JAXP 1.3

XML JAXB 2.0 Unmarshaller JAXB 1.0
JAXBContext JAXB 1.0
javax.xml.bind.helpers.DefaultValidationEventHandler
JAXB 2.0

Unmarshaller JAXB

Unmarshaller "" JAXB ""

"" JAXB

// This method is called immediately after the object is createc
// object begins. The callback provides an opportunity to initial
void beforeUnmarshal(Unmarshaller, Object parent);
//This method is called after all the properties (except IDREF) //but before this object is set to the parent object.
void afterUnmarshal(Unmarshaller, Object parent);

Listener setListener(Listener)
JAXB

''

afterUnmarshal(Object, Object)

version $Revision: 1.32 $ $Date: 2005/08/18 15:18:26 $
since JAXB1.0
See javax.xml.bind.JAXBContext, javax.xml.bind.Marshaller,
also javax.xml.bind.Validator

The Unmarshaller class governs the process of deserializing XML data into newly created Java content trees, optionally validating the XML data as it is unmarshalled. It provides an overloading of unmarshal methods for many different input kinds.

Unmarshalling from a File:

    JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
    Unmarshaller u = jc.createUnmarshaller();
    Object o = u.unmarshal( new File( "nosferatu.xml" ) );

Unmarshalling from an InputStream:

    InputStream is = new FileInputStream( "nosferatu.xml" );
    JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
    Unmarshaller u = jc.createUnmarshaller();
    Object o = u.unmarshal( is );

Unmarshalling from a URL:
JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();
URL url = new URL("http://beaker.east/nosferatu.xml");
Object o = u.unmarshal(url);

Unmarshalling from a StringBuffer using a
javax.xml.transform.stream.StreamSource:

JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();
StringBuffer xmlStr = new StringBuffer("<?xml version="1.0"
Object o = u.unmarshal(new StreamSource(new StringReader(xmlStr.toString()));

Unmarshalling from a org.w3c.dom.Node:

JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.parse(new File("nosferatu.xml"));
Object o = u.unmarshal(doc);

Unmarshalling from a javax.xml.transform.sax.SAXSource using a client
specified validating SAX2.0 parser:

    // configure a validating SAX2.0 parser (Xerces2)
    static final String JAXB_SCHEMA_LANGUAGE =
        "http://java.sun.com/xml/jaxp/properties/schemaLanguage
    static final String JAXB_SCHEMA_LOCATION =
        "http://java.sun.com/xml/jaxp/properties/schemaSource"
    static final String W3C_XML_SCHEMA =
        "http://www.w3.org/2001/XMLSchema"
    System.setProperty("javax.xml.parsers.SAXParserFactory",
        "org.apache.xerces.jaxp.SAXParserFactory"
    SAXParserFactory spf = SAXParserFactory.newInstance();
    spf.setNamespaceAware(true);
    spf.setValidating(true);
    SAXParser saxParser = spf.newSAXParser();
try {
    saxParser.setProperty(JAXP_SCHEMA_LANGUAGE, W3C_XML_SCHEMA);
    saxParser.setProperty(JAXP_SCHEMA_LOCATION, "http://...");
} catch (SAXNotRecognizedException x) {
    // exception handling omitted
}

XMLReader xmlReader = saxParser.getXMLReader();
SAXSource source =
    new SAXSource( xmlReader, new InputSource( "http://..." ) );

// Setup JAXB to unmarshal
JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();
ValidationEventCollector vec = new ValidationEventCollector();
u.setEventHandler( vec );

// turn off the JAXB provider's default validation mechanism
// avoid duplicate validation
u.setValidating( false )

// unmarshal
Object o = u.unmarshal( source );

// check for events
if( vec.hasEvents() ) {
    // iterate over events
}

Unmarshalling from a StAX XMLStreamReader:

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

javax.xml.stream.XMLStreamReader xmlStreamReader =
    javax.xml.stream.XMLInputFactory().newInstance().createXMLStreamReader();

Object o = u.unmarshal( xmlStreamReader );

Unmarshalling from a StAX XMLEventReader:

JAXBContext jc = JAXBContext.newInstance( "com.acme.foo" );
Unmarshaller u = jc.createUnmarshaller();

javax.xml.stream.XMLEventReader xmlEventReader =
    javax.xml.stream.XMLInputFactory().newInstance().createXMLEventReader();
Object o = u.unmarshal( xmlEventReader );

Unmarshalling XML Data

Unmarshalling can deserialize XML data that represents either an entire XML document or a subtree of an XML document. Typically, it is sufficient to use the unmarshalling methods described by Unmarshal root element that is declared globally. These unmarshal methods utilize JAXBContext’s mapping of global XML element declarations and type definitions to JAXB mapped classes to initiate the unmarshalling of the root element of XML data. When the JAXBContext’s mappings are not sufficient to unmarshal the root element of XML data, the application can assist the unmarshalling process by using the unmarshal by declaredType methods. These methods are useful for unmarshalling XML data where the root element corresponds to a local element declaration in the schema.

An unmarshal method never returns null. If the unmarshal process is unable to unmarshal the root of XML content to a JAXB mapped object, a fatal error is reported that terminates processing by throwing JAXBException.

Unmarshal a root element that is globally declared

The unmarshal methods that do not have an declaredType parameter use JAXBContext to unmarshal the root element of an XML data. The JAXBContext instance is the one that was used to create this Unmarshaller. The JAXBContext instance maintains a mapping of globally declared XML element and type definition names to JAXB mapped classes. The unmarshal method checks if JAXBContext has a mapping from the root element’s XML name and/or @xsi:type to a JAXB mapped class. If it does, it umarshalls the XML data using the appropriate JAXB mapped class. Note that when the root element name is unknown and the root element has an @xsi:type, the XML data is unmarshalled using that JAXB mapped class as the value of a JAXBElement. When the JAXBContext object does not have a mapping for the root element’s name nor its @xsi:type, if it exists, then the unmarshal operation will abort immediately by throwing a UnmarshalException. This exception scenario can be worked around
by using the unmarshal by declaredType methods described in the next subsection.

Unmarshal by Declared Type

The unmarshal methods with a declaredType parameter enable an application to deserialize a root element of XML data, even when there is no mapping in JAXBContext of the root element's XML name. The unmarshaller unmarshals the root element using the application provided mapping specified as the declaredType parameter. Note that even when the root element's element name is mapped by JAXBContext, the declaredType parameter overrides that mapping for deserializing the root element when using these unmarshal methods. Additionally, when the root element of XML data has an xsi:type attribute and that attribute's value references a type definition that is mapped to a JAXB mapped class by JAXBContext, that the root element's xsi:type attribute takes precedence over the unmarshal methods declaredType parameter. These methods always return a JAXBElement<declaredType> instance. The table below shows how the properties of the returned JAXBElement instance are set.

<table>
<thead>
<tr>
<th>Unmarshal By Declared Type returned JAXBElement</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXBElemnt Property</td>
<td>Value</td>
</tr>
<tr>
<td>name</td>
<td>xml element name</td>
</tr>
<tr>
<td>value</td>
<td>instanceof declaredType</td>
</tr>
<tr>
<td>declaredType</td>
<td>unmarshal method declaredType parameter</td>
</tr>
<tr>
<td>scope</td>
<td>null (actual scope is unknown)</td>
</tr>
</tbody>
</table>

The following is an example of unmarshal by declaredType method.

Unmarshal by declaredType from a org.w3c.dom.Node:

```xml
<xs:schema>
  <xs:complexType name="FooType">...
  </xs:complexType>
<!-- global element declaration "PurchaseOrder" -->
<xs:element name="PurchaseOrder">
  <xs:complexType>
    <xs:sequence>
```
<xs:element name="foo" type="FooType"/>
...
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

JAXBContext jc = JAXBContext.newInstance("com.acme.foo");
Unmarshaller u = jc.createUnmarshaller();

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
dbf.setNamespaceAware(true);
DocumentBuilder db = dbf.newDocumentBuilder();
Document doc = db.parse(new File("nosferatu.xml"));
Element fooSubtree = ...; // traverse DOM till reach xml e
// local element declaration in

// FooType is the JAXB mapping of the type of local element
JAXBElement<FooType> foo = u.unmarshal(fooSubtree, FooType

**Support for SAX2.0 Compliant Parsers**

A client application has the ability to select the SAX2.0 compliant parser of their choice. If a SAX parser is not selected, then the JAXB Provider's default parser will be used. Even though the JAXB Provider's default parser is not required to be SAX2.0 compliant, all providers are required to allow a client application to specify their own SAX2.0 parser. Some providers may require the client application to specify the SAX2.0 parser at schema compile time. See [unmarshal(Source)](#) for more detail.

**Validation and Well-Formedness**

A client application can enable or disable JAXP 1.3 validation mechanism via the `setSchema(javax.xml.validation.Schema)` API. Sophisticated clients can specify their own validating SAX 2.0 compliant parser and bypass the JAXP 1.3 validation mechanism using the [unmarshal(Source)](#) API.

Since unmarshalling invalid XML content is defined in JAXB 2.0, the Unmarshaller default validation event handler was made more lenient than in JAXB 1.0. When schema-derived code generated by
JAXB 1.0 binding compiler is registered with JAXBContext, the default unmarshal validation handler is DefaultValidationEventHandler and it terminates the marshal operation after encountering either a fatal error or an error. For a JAXB 2.0 client application, there is no explicitly defined default validation handler and the default event handling only terminates the marshal operation after encountering a fatal error.

**Supported Properties**

There currently are not any properties required to be supported by all JAXB Providers on Unmarshaller. However, some providers may support their own set of provider specific properties.

**Unmarshal Event Callbacks**

The Unmarshaller provides two styles of callback mechanisms that allow application specific processing during key points in the unmarshalling process. In 'class defined' event callbacks, application specific code placed in JAXB mapped classes is triggered during unmarshalling. 'External listeners' allow for centralized processing of unmarshal events in one callback method rather than by type event callbacks.

'Class defined' event callback methods allow any JAXB mapped class to specify its own specific callback methods by defining methods with the following method signature:

```java
// This method is called immediately after the object is created.
// object begins. The callback provides an opportunity to initialize JavaBean properties prior to unmarshalling.
void beforeUnmarshal(Unmarshaller, Object parent);

// This method is called after all the properties (except IDREF) are unmarshalled for this object,
// but before this object is set to the parent object.
void afterUnmarshal(Unmarshaller, Object parent);
```

The class defined callback methods should be used when the callback method requires access to non-public methods and/or fields of the class.
The external listener callback mechanism enables the registration of a `Unmarshaller.Listener` instance with an `Unmarshaller.Listener.setListener(Listener)`. The external listener receives all callback events, allowing for more centralized processing than per class defined callback methods. The external listener receives events when unmarshalling process is marshalling to a JAXB element or to JAXB mapped class.

The 'class defined' and external listener event callback methods are independent of each other, both can be called for one event. The invocation ordering when both listener callback methods exist is defined in `Unmarshaller.Listener.beforeUnmarshal(Object, Object)` and `Unmarshaller.Listener.afterUnmarshal(Object, Object)`. An event callback method throwing an exception terminates the current unmarshal process.

Since: JAXB1.0
Version: $Revision: 1.32 $ $Date: 2005/08/18 15:18:26 $
Author: Ryan Shoemaker, Sun Microsystems, Inc.
Kohsuke Kawaguchi, Sun Microsystems, Inc.
Joe Fialli, Sun Microsystems, Inc.
See Also: JAXBContext, Marshaller, Validator

### Nested Class Summary

<table>
<thead>
<tr>
<th>static class</th>
<th>Unmarshaller.Listener</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Register an instance of an implementation of this class with <code>Unmarshaller</code> to externally listen for unmarshal events.</td>
</tr>
</tbody>
</table>

### Method Summary
**getAdapter(Class type)**

Gets the adapter associated with the specified type.

**AttachmentUnmarshaller getAttachmentUnmarshaller()**

Return the current event handler or the default event handler if one hasn't been set.

**Unmarshaller.Listener getListener()**

Return `Unmarshaller.Listener` registered with this `Unmarshaller`.

**setAdapter(Class type, A adapter)**

Associates a configured instance of `XmlAdapter` with this unmarshaller.

**setAdapter(XmlAdapter adapter)**

Associates a configured instance of `XmlAdapter` with this unmarshaller.

**setAttachmentUnmarshaller(AttachmentUnmarshaller au)**

Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments.

**setEventHandler(ValidationEventHandler handler)**

Allow an application to register a `ValidationEventHandler`.  

**setListener(Unmarshaller.Listener listener)**

Register unmarshal event callback `Unmarshaller.Listener` with this `Unmarshaller`.  

**setProperty(String name, Object value)**

Set the particular property in the underlying implementation of Unmarshaller.

**setSchema(Schema schema)**

Specify the JAXP 1.3 `Schema` object that should be used to validate subsequent unmarshal operations against.

**setValidating(boolean validating)**

Deprecated. since JAXB2.0, please see `setSchema()`
Unmarshal XML data from the specified file and return the resulting content tree.  
`object unmarshal(InputSource source)`
Unmarshal XML data from the specified SAX InputSource and return the resulting content tree.  
`object unmarshal(InputStream is)`
Unmarshal XML data from the specified InputStream and return the resulting content tree.  
`object unmarshal(Node node)`
Unmarshal global XML data from the specified DOM tree and return the resulting content tree.

<T>  
`JAXBElement<T> unmarshal(Node node, Class<T> declaredType)`
Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.  
`Object unmarshal(Reader reader)`
Unmarshal XML data from the specified Reader and return the resulting content tree.  
`Object unmarshal(Source source)`
Unmarshal XML data from the specified XML Source and return the resulting content tree.

<T>  
`JAXBElement<T> unmarshal(Source source, Class<T> declaredType)`
Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.  
`Object unmarshal(URL url)`
Unmarshal XML data from the specified URL and return the resulting content tree.  
`Object unmarshal(XMLEventReader reader)`
Unmarshal XML data from the specified pull parser and return the resulting content tree.

<T>  
`JAXBElement<T> unmarshal(XMLEventReader reader, Class<T> declaredType)`
Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.  
`Object unmarshal(XMLStreamReader reader)`
Unmarshal XML data from the specified pull parser and return the resulting content tree.

<T>  
`JAXBElement<T> unmarshal(XMLStreamReader reader, Class<T> declaredType)`
Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

---

**Method Detail**
public Object unmarshal(java.io.File f) throws JAXBException
XML

f XML
return java
Throws JAXBException:
Throws UnmarshalException: ValidationEventHandler handleEvent
false Unmarshaller XML Java XML
Throws IllegalArgumentException: File null

unmarshal

Object unmarshal(File f)
throws JAXBException

Unmarshal XML data from the specified file and return the resulting content tree.

Implements Unmarshal Global Root Element.

Parameters:
  f - the file to unmarshal XML data from

Returns:
  the newly created root object of the java content tree

Throws:
  JAXBException - If any unexpected errors occur while unmarshalling
  UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
  IllegalArgumentException - If the file parameter is null

public Object unmarshal(java.io.InputStream is) throws
**JAXBException**

**InputStream XML unmarshal API**

```
is XML InputStream
return java

Throws JAXBException:

Throws UnmarshalException: ValidationEventHandler handleEvent
false Unmarshaller XML Java XML

Throws IllegalArgumentException: InputStream null
```

**unmarshal**

```
Object unmarshal(InputStream is)
throws JAXBException
```

Unmarshal XML data from the specified InputStream and return the resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API.

Implements **Unmarshal Global Root Element**.

**Parameters:**

- is - the InputStream to unmarshal XML data from

**Returns:**

- the newly created root object of the java content tree

**Throws:**

- JAXBException - If any unexpected errors occur while unmarshalling
- UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
- IllegalArgumentException - If the InputStream parameter is null

```
public Object unmarshal(java.io.Reader reader) throws
```
JAXBException
Reader XML unmarshal API Reader ID

```
  reader       XML Reader
  return      java
Throws    JAXBException:
  UnmarshalException: ValidationEventHandler handleEvent
  false Unmarshaller XML Java XML
  IllegalArgumentException: InputStream null
  since     JAXB2.0
```

unmarshal

```
  Object unmarshal(Reader reader)
  throws JAXBException
```

Unmarshal XML data from the specified Reader and return the resulting content tree. Validation event location information may be incomplete when using this form of the unmarshal API, because a Reader does not provide the system ID.

Implements Unmarshal Global Root Element.

Parameters:
  reader - the Reader to unmarshal XML data from

Returns:
  the newly created root object of the java content tree

Throws:
  JAXBException - If any unexpected errors occur while unmarshalling
  UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
  IllegalArgumentException - If the InputStream parameter is null
public Object unmarshal(java.net.URL url) throws JAXBException

Parameters:
url - the url to unmarshal XML data from

Returns:
the newly created root object of the java content tree

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshalingException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
public Object unmarshal(org.xml.sax.InputSource source) throws JAXBException

SAX InputSource  XML

unmarshal

Object unmarshal(InputSource source)

throws JAXBException

Unmarshal XML data from the specified SAX InputSource and return the resulting content tree.

Implements Unmarshal Global Root Element.

Parameters:

- source - the input source to unmarshal XML data from

Returns:

the newly created root object of the java content tree

Throws:

- JAXBException - If any unexpected errors occur while unmarshalling
- UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
- IllegalArgumentException - If the InputSource parameter is null
public Object unmarshal(org.w3c.dom.Node node) throws JAXBException
DOM XML

return java
Throws JAXBException:
UnmarshalException: Unmarshaller XML Java XML

Throws ValidationEventHandler handleEvent
false

Throws IllegalArgumentException: Node null
See also unmarshal(org.w3c.dom.Node, Class)

unmarshal

Object unmarshal(Node node)
throws JAXBException

Unmarshal global XML data from the specified DOM tree and return the resulting content tree.

Implements Unmarshal Global Root Element.

Parameters:
node - the document/element to unmarshal XML data from. The caller must support at least Document and Element.

Returns:
the newly created root object of the java content tree

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
IllegalArgumentException - If the Node parameter is null

See Also:
unmarshal(org.w3c.dom.Node, Class)

unmarshal

<T> JAXBEElement<T> unmarshal(Node node, Class<T> declaredType) throws JAXBException

Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.

Implements Unmarshal by Declared Type

Parameters:
node - the document/element to unmarshal XML data from. The caller must support at least Document and Element.
declaredType - appropriate JAXB mapped class to hold node's XML data.

Returns:
JAXB Element representation of node

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
IllegalArgumentException - If any parameter is null

Since:
JAXB2.0

public Object unmarshal(javax.xml.transform.Source source) throws JAXBException
XML Source  XML
SAX 2.0

JAXB SAX 2.0 JAXB
SAXSource XMLReader SAX 2.0 XMLReader
org.xml.sax.ErrorHandler JAXB JAXB
ValidationEventHandler SAXSource XMLReader JAXB

JAXB SAX 2.0
JAXB JAXB ValidationEventHandler SAX 2.0 JAXB

Node JAXB

source XML XML Source SAXSourceDOMSource StreamSource
return java
Throws JAXBException:
Throws UnmarshalException: ValidationEventHandler handleEvent
false Unmarshaller XML Java XML

See also unmarshal(javax.xml.transform.Source, Class)

unmarshal

Object unmarshal(Source source)
throws JAXBException

Unmarshal XML data from the specified XML Source and return the resulting content tree.
Implements **Unmarshal Global Root Element**.

**SAX 2.0 Parser Pluggability**

A client application can choose not to use the default parser mechanism supplied with their JAXB provider. Any SAX 2.0 compliant parser can be substituted for the JAXB provider's default mechanism. To do so, the client application must properly configure a SAXSource containing an XMLReader implemented by the SAX 2.0 parser provider. If the XMLReader has an org.xml.sax.ErrorHandler registered on it, it will be replaced by the JAXB Provider so that validation errors can be reported via the ValidationEventHandler mechanism of JAXB. If the SAXSource does not contain an XMLReader, then the JAXB provider's default parser mechanism will be used.

This parser replacement mechanism can also be used to replace the JAXB provider's unmarshal-time validation engine. The client application must properly configure their SAX 2.0 compliant parser to perform validation (as shown in the example above). Any SAXParserExceptions encountered by the parser during the unmarshal operation will be processed by the JAXB provider and converted into JAXB ValidationEvent objects which will be reported back to the client via the ValidationEventHandler registered with the Unmarshaller. *Note:* specifying a substitute validating SAX 2.0 parser for unmarshalling does not necessarily replace the validation engine used by the JAXB provider for performing on-demand validation.

The only way for a client application to specify an alternate parser mechanism to be used during unmarshal is via the unmarshal(SAXSource) API. All other forms of the unmarshal method (File, URL, Node, etc) will use the JAXB provider's default parser and validator mechanisms.

**Parameters:**

- **source** - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and StreamSource)

**Returns:**

- the newly created root object of the java content tree
Throws:

- **JAXBException** - If any unexpected errors occur while unmarshalling.
- **UnmarshalException** - If the `ValidationEventHandler` returns false from its `handleEvent` method or the Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#).

See Also:

- `unmarshal(javax.xml.transform.Source, Class)`

---

```java
<T> JAXBElement<T> unmarshal(Source source, Class<T> declaredType) throws JAXBException
```

Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.

Implements [Unmarshal by Declared Type](#)

See [SAX 2.0 Parser Pluggability](#)

**Parameters:**

- `source` - the XML Source to unmarshal XML data from (providers are only required to support SAXSource, DOMSource, and StreamSource).
- `declaredType` - appropriate JAXB mapped class to hold `source`'s xml root element.

**Returns:**

Java content rooted by [JAXB Element](#)

**Throws:**

- **JAXBException** - If any unexpected errors occur while unmarshalling.
- **UnmarshalException** - If the `ValidationEventHandler` returns false from its `handleEvent` method or the Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](#).
IllegalArgumentException - If any parameter is null

Since:
JAXB2.0

public Object unmarshal(XMLStreamReader reader) throws JAXBException
pull XML

START_DOCUMENT START_ELEMENT
reader
reader
return java
Throws JAXBException:
throws UnmarshalException: ValidationEventHandler handleEvent
false Unmarshaller XML Java XML
Throws IllegalStateException: reader null
Throws IllegalArgumentException: reader null
since JAXB2.0
See also unmarshal(javax.xml.stream.XMLStreamReader, Class)

unmarshal

Object unmarshal(XMLStreamReader reader)
throws JAXBException

Unmarshal XML data from the specified pull parser and return the resulting content tree.

Implements Unmarshal Global Root Element.

This method assumes that the parser is on a START_DOCUMENT
or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

**Parameters:**
reader - The parser to be read.

**Returns:**
the newly created root object of the java content tree.

**Throws:**
- **JAXBException** - If any unexpected errors occur while unmarshalling
- **UnmarshallerException** - If the ValidationEventHandle returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See [Unmarshalling XML Data](Unmarshalling_XML_Data)
- **IllegalArgumentException** - If the reader parameter is null
- **IllegalStateException** - If reader is not pointing to a START_DOCUMENT or START_ELEMENT event.

**Since:**
JAXB2.0

**See Also:**
unmarshal(javax.xml.stream.XMLStreamReader, Class)

---

unmarshal

<T> JAXBElement<T> unmarshal(XMLStreamReader reader, Class<T> declaredType)

throws JAXBException

Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

This method implements **unmarshal by declaredType**.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns
successfully, the reader will be pointing at the token right after the end event.

**Parameters:**
- reader - The parser to be read.
- declaredType - appropriate JAXB mapped class to hold reader's START_ELEMENT XML data.

**Returns:**
- content tree rooted by JAXB Element representation

**Throws:**
- JAXBException - If any unexpected errors occur while unmarshalling
- UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
- IllegalArgumentException - If any parameter is null

**Since:**
- JAXB2.0

---

```java
public Object unmarshal(XMLEventReader reader) throws JAXBException
{
    pull XML

    START_DOCUMENT  START_ELEMENT
    reader
    reader
    return  java
    Throws  JAXBException:
    Throws  UnmarshalException: ValidationEventHandler handleEvent
    false Unmarshaller XML Java XML
    Throws  IllegalArgumentException: reader null
    Throws  IllegalStateException: reader null
    since  JAXB2.0
}
unmarshal

Object unmarshal(XMLEventReader reader) throws JAXBException

Unmarshal XML data from the specified pull parser and return the resulting content tree.

This method is an Unmarshal Global Root method.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

Parameters:
    reader - The parser to be read.

Returns:
the newly created root object of the java content tree.

Throws:
    JAXBException - If any unexpected errors occur while unmarshalling
    UnmarshalException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
    IllegalArgumentException - If the reader parameter is null
    IllegalStateException - If reader is not pointing to a START_DOCUMENT or START_ELEMENT event.

Since:
    JAXB2.0

See Also:
    unmarshal(javax.xml.stream.XMLEventReader, Class)
unmarshal

<T> JAXBElement<T> unmarshal(XMLEventReader reader, Class<T> declaredType) throws JAXBException

Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

This method implements unmarshal by declaredType.

This method assumes that the parser is on a START_DOCUMENT or START_ELEMENT event. Unmarshalling will be done from this start event to the corresponding end event. If this method returns successfully, the reader will be pointing at the token right after the end event.

Parameters:
reader - The parser to be read.
declaredType - appropriate JAXB mapped class to hold reader's START_ELEMENT XML data.

Returns:
content tree rooted by JAXB Element representation

Throws:
JAXBException - If any unexpected errors occur while unmarshalling
UnmarshallingException - If the ValidationEventHandler returns false from its handleEvent method or the Unmarshaller is unable to perform the XML to Java binding. See Unmarshalling XML Data
IllegalArgumentException - If any parameter is null

Since:
JAXB2.0

public UnmarshallerHandler getUnmarshallerHandler() XML unmarshaller
getUnmarshallerHandler

Get an unmarshaller handler object that can be used as a component in an XML pipeline.

The JAXB Provider can return the same handler object for multiple invocations of this method. In other words, this method does not necessarily create a new instance of UnmarshallerHandler. If the application needs to use more than one UnmarshallerHandler, it should create more than one Unmarshaller.

Returns:
the unmarshaller handler object

See Also:
UnmarshallerHandler

public void setValidating(boolean validating) throws JAXBException

unmarshal

JAXB SAX 2.0 API JAXB ""

JAXB 2.0 API

setSchema(javax.xml.val
setValidating

void setValidating(boolean validating)
    throws JAXBException

Deprecated. since JAXB2.0, please see
setSchema(javax.xml.validation.Schema)

Specifies whether or not the default validation mechanism of the
Unmarshaller should validate during unmarshal operations. By
default, the Unmarshaller does not validate.

This method may only be invoked before or after calling one of the
unmarshal methods.

This method only controls the JAXB Provider's default unmarshal-
time validation mechanism - it has no impact on clients that specify
their own validating SAX 2.0 compliant parser. Clients that specify
their own unmarshal-time validation mechanism may wish to turn off
the JAXB Provider's default validation mechanism via this API to avoid "double validation".

This method is deprecated as of JAXB 2.0 - please use the new
setSchema(javax.xml.validation.Schema) API.

Parameters:
    validating - true if the Unmarshaller should validate during
unmarshal, false otherwise

Throws:
    JAXBException - if an error occurred while enabling or disabling
validation at unmarshal time
    UnsupportedOperationException - could be thrown if this method
is invoked on an Unmarshaller created from a JAXBContext referencing JAXB 2.0 mapped classes

public boolean isValidating() throws JAXBException
Unmarshaller

API JAXB

JAXB 2.0

getSchema() API

return Unmarshaller true false
Throws JAXBException:
Throws UnsupportedOperationException: JAXB 2.0
JAXBContext Unmarshaller
deprecated JAXB 2.0 getSchema()

isValidating

boolean isValidating() throws JAXBException

Deprecated. since JAXB2.0, please see getSchema()

Indicates whether or not the Unmarshaller is configured to validate during unmarshal operations.

This API returns the state of the JAXB Provider's default unmarshal-time validation mechanism.

This method is deprecated as of JAXB 2.0 - please use the new getSchema() API.

Returns:
true if the Unmarshaller is configured to validate during unmarshal operations, false otherwise

Throws:
public void setEventHandler(ValidationEventHandler handler) throws JAXBException

ValidationEventHandler

unmarshal ValidationEventHandler
unmarshal ValidationEventHandler

null Unmarshaller
handler

Unmarshaller

void setEventHandler(ValidationEventHandler handler)

throws JAXBException

Allow an application to register a ValidationEventHandler.

The ValidationEventHandler will be called by the JAXB Provider if any validation errors are encountered during calls to any of the unmarshal methods. If the client application does not register a ValidationEventHandler before invoking the unmarshal methods, then ValidationEvents will be handled by the default event handler which will terminate the unmarshal operation after the first error or fatal error is encountered.

Calling this method with a null parameter will cause the Unmarshaller to revert back to the default event handler.
Parameters:
  handler - the validation event handler

Throws:
  JAXBException - if an error was encountered while setting the event handler

public ValidationEventHandler getEventHandler() throws JAXBException

  return ValidationEventHandler
  Throws JAXBException:

getEventHandler

ValidationEventHandler getEventHandler() throws JAXBException

  Return the current event handler or the default event handler if one hasn't been set.

Returns:
  the current ValidationEventHandler or the default event handler if it hasn't been set

Throws:
  JAXBException - if an error was encountered while getting the current event handler

public void setProperty(String name, Object value) throws PropertyException

UnmarshalException JAXB PropertyException

  name
  value
  Throws PropertyException:
  Throws IllegalArgumentException: name null
**setProperty**

```java
void setProperty(String name, 
                Object value) 
    throws PropertyException
```

Set the particular property in the underlying implementation of Unmarshaller. This method can only be used to set one of the standard JAXB defined properties above or a provider specific property. Attempting to set an undefined property will result in a PropertyException being thrown. See [Supported Properties](#).

**Parameters:**
- `name` - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
- `value` - the value of the property to be set

**Throws:**
- `PropertyException` - when there is an error processing the given property or value
- `IllegalArgumentException` - If the name parameter is null

---

**public Object getProperty(String name) throws PropertyException**

```java
Unmarshaller JAXB PropertyException
    name
    return
    Throws PropertyException:
    Throws IllegalArgumentException: name null
```

---

**getProperty**

```java
Object getProperty(String name) 
    throws PropertyException
```
Get the particular property in the underlying implementation of Unmarshaller. This method can only be used to get one of the standard JAXB defined properties above or a provider specific property. Attempting to get an undefined property will result in a PropertyException being thrown. See Supported Properties.

**Parameters:**

- `name` - the name of the property to retrieve

**Returns:**

the value of the requested property

**Throws:**

- `PropertyException` - when there is an error retrieving the given property or value property name
- `IllegalArgumentException` - If the name parameter is null

```java
public void setSchema(javax.xml.validation.Schema schema)
```

**JAXP 1.3**

Set the JAXP 1.3 Schema object that should be used to validate subsequent unmarshal operations against. Passing null into this method will disable validation.

```java
public void setValidating(boolean)
```

**API**

Set whether the Unmarshaller should validate during unmarshaling. Null indicates the default, which is to not validate.

```java
setSchema(Schema schema)
```

Throws

- `UnsupportedOperationException`: JAXB 1.0 JAXBContext Unmarshaller

**since**

JAXB2.0
This method replaces the deprecated `setValidating(boolean)` API.

Initially this property is set to `null`.

**Parameters:**

- `schema` - Schema object to validate unmarshal operations against or `null` to disable validation

**Throws:**

- `UnsupportedOperationException` - could be thrown if this method is invoked on an Unmarshaller created from a JAXBContext referencing JAXB 1.0 mapped classes

**Since:**

JAXB2.0

---

```java
public javax.xml.validation.Schema getSchema()
JAXP 1.3
null

#isValidating() API
null

boolean isValidating = u.getSchema()!=null;
return Schema null

Throws
UnsupportedOperationException: JAXB 1.0 JAXBContext Unmarshaller
since JAXB2.0
```

**getSchema**

```java
Schema getSchema()
```

Get the JAXP 1.3 `Schema` object being used to perform unmarshal-time validation. If there is no Schema set on the unmarshaller, then this method will return null indicating that unmarshal-time validation will not be performed.
This method provides replacement functionality for the deprecated `isValidating()` API as well as access to the Schema object. To determine if the Unmarshaller has validation enabled, simply test the return type for null:

```java
boolean isValidating = u.getSchema()!=null;
```

**Returns:**
the Schema object being used to perform unmarshal-time validation or null if not present

**Throws:**
`UnsupportedOperationException` - could be thrown if this method is invoked on an Unmarshaller created from a JAXBContext referencing JAXB 1.0 mapped classes

**Since:**
JAXB2.0

---

**public void setAdapter("XmlAdapter adapter)\**

```java
XmlAdapter unmarshaller

setAdapter(adapter.getClass(),adapter);
```

**Throws**
`IllegalArgumentException`: adapter null

**Throws**
`UnsupportedOperationException`: JAXB 1.0 since JAXB2.0

**See also**
`setAdapter(Class,XmlAdapter)`

---

**setAdapter**

```java
void setAdapter(XmlAdapter adapter)
```

Associates a configured instance of `XmlAdapter` with this unmarshaller.

This is a convenience method that invokes `setAdapter(adapter.getClass(),adapter);`.
Throws:
   IllegalArgumentException - if the adapter parameter is null.
   UnsupportedOperationException - if invoked against a JAXB 1.0 implementation.

Since:
   JAXB2.0

See Also:
   setAdapter(Class,XmlAdapter)

setAdapter

<A extends XmlAdapter> void setAdapter(Class<A> type, A adapter)

Associates a configured instance of XmlAdapter with this unmarshaller.

Every unmarshaller internally maintains a Map<Class,XmlAdapter>, which it uses for unmarshalling classes whose fields/methods are annotated with XmlJavaTypeAdapter.

This method allows applications to use a configured instance of XmlAdapter. When an instance of an adapter is not given, an unmarshaller will create one by invoking its default constructor.

Parameters:
   type - The type of the adapter. The specified instance will be used when XmlJavaTypeAdapter.value() refers to this type.
   adapter - The instance of the adapter to be used. If null, it will un-register the current adapter set for this type.

Throws:
   IllegalArgumentException - if the type parameter is null.
   UnsupportedOperationException - if invoked against a JAXB 1.0 implementation.

Since:
   JAXB2.0
getAdapter

<A extends XmAdapter> A getAdapter(Class<A> type)

 Gets the adapter associated with the specified type. This is the reverse operation of the setAdapter(javax.xml.bind.annotation.adapters.XmlAdapter) method.

 **Throws:**  
  * IllegalArgumentException - if the type parameter is null.  
  * UnsupportedOperationException - if invoked agains a JAXB 1.0 implementation.

 **Since:**  
  JAXB2.0

---

public void setAttachmentUnmarshaller(AttachmentUnmarshaller au)

cid id URI

 #setSchema(Schema) unmarshaller XOP

 **Throws**  
  * IllegalStateException:

---

setAttachmentUnmarshaller

void setAttachmentUnmarshaller(AttachmentUnmarshaller au)

 Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments.

 Unmarshal time validation, enabled via setSchema(Schema), must be supported even when unmarshaller is performing XOP processing.
Throws:

IllegalArgumentException - if attempt to concurrently call this method during a unmarshal operation.

```java
public AttachmentUnmarshaller getAttachmentUnmarshaller()
```

```java
AttachmentUnmarshaller getAttachmentUnmarshaller()
```

```java
public void setListener(Unmarshaller.Listener listener)
```

```java
Unmarshaller Listener Listener Listener Listener listener null Listener
```

**setListener**

```java
void setListener(Unmarshaller.Listener listener)
```

Register unmarshal event callback Unmarshaller.Listener with this Unmarshaller.

There is only one Listener per Unmarshaller. Setting a Listener replaces the previous set Listener. One can unregister current Listener by setting listener to null.

**Parameters:**

- listener - provides unmarshal event callbacks for this
public Unmarshaller.Listener getListener()

Returns:
registered Unmarshaller.Listener or null if no Listener is registered with this Unmarshaller.

Since:
JAXB2.0
javax.xml.bind  **Class Unmarshaller.Listener**

**java.lang.Object**

| javax.xml.bind.Unmarshaller.Listener |

**Enclosing interface:**

**Unmarshaller**

---

```java
public abstract static class Unmarshaller.Listener
edinstanted by Object

**Contained within:** Unmarshaller

Unmarshaller

XML JAXB JAXB XML JAXBElement JAXB Java

Unmarshal Event Callback

(@link #setListener(Listener)) (@link #getListener() since JAXB2.0

Register an instance of an implementation of this class with Unmarshaller
to externally listen for unmarshal events.

This class enables pre and post processing of an instance of a JAXB
mapped class as XML data is unmarshalled into it. The event callbacks
are called when unmarshalling XML content into a JAXBElement
instance or a JAXB mapped class that represents a complex type
definition. The event callbacks are not called when unmarshalling to an
instance of a Java datatype that represents a simple type definition.

External listener is one of two different mechanisms for defining
unmarshal event callbacks. See Unmarshal Event Callbacks for an overview.

(@link #setListener(Listener)} (@link #getListener{})

**Since:**
JAXB2.0

---

### Constructor Summary

**Unmarshaller.Listener()**

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>afterUnmarshal(Object target, Object parent)</strong></td>
<td>Callback method invoked after unmarshalling XML data into target.</td>
</tr>
<tr>
<td><strong>beforeUnmarshal(Object target, Object parent)</strong></td>
<td>Callback method invoked before unmarshalling into target.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.**Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

---

### Constructor Detail

**public Unmarshaller.Listener()**
### Unmarshaller.Listener

**public Unmarshaller.Listener()**

#### Method Detail

**public void beforeUnmarshal(Object target, Object parent)**

```java
public void beforeUnmarshal(Object target, Object parent)
```

**target**

<table>
<thead>
<tr>
<th>target</th>
<th>target</th>
<th>beforeUnmarshal</th>
</tr>
</thead>
<tbody>
<tr>
<td>target</td>
<td>JAXB</td>
<td>null</td>
</tr>
<tr>
<td>parent</td>
<td>target</td>
<td>JAXB</td>
</tr>
<tr>
<td></td>
<td>target</td>
<td>null</td>
</tr>
</tbody>
</table>

**beforeUnmarshal**

Public void **beforeUnmarshal(Object target, Object parent)**

Callback method invoked before unmarshalling into target.

This method is invoked immediately after target was created and before the unmarshalling of this object begins. Note that if the class of target defines its own beforeUnmarshal method, the class specific callback method is invoked before this method is invoked.

**Parameters:**

- **target** - non-null instance of JAXB mapped class prior to unmarshalling into it.
- **parent** - instance of JAXB mapped class that will eventually reference target. null when target is root element.

**public void afterUnmarshal(Object target, Object parent)**

```java
public void afterUnmarshal(Object target, Object parent)
```
afterUnmarshal

```java
public void afterUnmarshal(Object target, Object parent)
```

Callback method invoked after unmarshalling XML data into `target`.

This method is invoked after all the properties (except IDREF) are unmarshalled into `target`, but before `target` is set into its parent object. Note that if the class of `target` defines its own `afterUnmarshal` method, the class specific callback method is invoked before this method is invoked.

**Parameters:**
- `target` - non-null instance of JAXB mapped class prior to unmarshalling into it.
- `parent` - instance of JAXB mapped class that will reference `target`. `null` when `target` is root element.
PS:
javax.xml.bind Interface UnmarshallerHandler

All Superinterfaces:

ContentHandler

public interface UnmarshallerHandler
extends ContentHandler

Implements: org.xml.sax.ContentHandler

SAX ContentHandler Unmarshaller

JAXB XML

JAXBContext context = JAXBContext.newInstance( "org.acme.foo" );
Unmarshaller unmarshaller = context.createUnmarshaller();
UnmarshallerHandler unmarshallerHandler = unmarshaller.getUnmarshaller();

SAXParserFactory spf = SAXParserFactory.newInstance();
spf.setNamespaceAware( true );

XMLReader xmlReader = spf.newSAXParser().getXMLReader();
xmlReader.setContentHandler( unmarshallerHandler );
xmlReader.parse( new InputSource( new FileInputStream( XML_FILE ) ) );

MyObject myObject= (MyObject)unmarshallerHandler.getResult();

/

version '$Revision: 1.2 $ $Date: 2006/03/08 16:55:17 $'

since JAXB1.0

See also getUnmarshallerHandler()

Unmarshaller implemented as SAX ContentHandler.
Applications can use this interface to use their JAXB provider as a component in an XML pipeline. For example:

```java
JAXBContext context = JAXBContext.newInstance( "org.acme.foo"
Unmarshaller unmarshaller = context.createUnmarshaller();
UnmarshallerHandler unmarshallerHandler = unmarshaller.getUnmarshallerHandler();
SAXParserFactory spf = SAXParserFactory.newInstance();
spf.setNamespaceAware( true );
XMLReader xmlReader = spf.newSAXParser().getXMLReader();
xmlReader.setContentHandler( unmarshallerHandler );
xmlReader.parse(new InputSource( new FileInputStream( XML_FILE )
MyObject myObject= (MyObject)unmarshallerHandler.getResult();
```

This interface is reusable: even if the user fails to unmarshal an object, s/he can still start a new round of unmarshalling.

Since:
JAXB1.0

Version:
$Revision: 1.2 $ $Date: 2006/03/08 16:55:17 $

Author:
- Kohsuke KAWAGUCHI, Sun Microsystems, Inc.

See Also:
Unmarshaller.getUnmarshallerHandler()

---

**Method Summary**

<table>
<thead>
<tr>
<th>Object</th>
<th><code>getResult()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Obtains the unmarshalled result.</td>
</tr>
</tbody>
</table>

**Methods inherited from interface org.xml.sax.ContentHandler**

- `characters`, `endDocument`, `endElement`, `endPrefixMapping`, ` ignorableWhitespace`, `processingInstruction`, `setDocumentLocator`, `skippedEntity`, `startDocument`, `startElement`, `startPrefixMapping`
public Object getResult() throws JAXBException, IllegalStateException

endDocument SAX

Throws  IllegalStateException: endDocument
Throws  JAXBException: SAXException

return  null

gResultado

Object  getResult()
throws  JAXBException, IllegalStateException

Obtains the unmarshalled result. This method can be called only after this handler receives the endDocument SAX event.

Returns:
always return a non-null valid object which was unmarshalled.

Throws:
IllegalStateException - if this method is called before this handler receives the endDocument event.
JAXBException - if there is any unmarshalling error. Note that the implementation is allowed to throw SAXException during the parsing when it finds an error.
PS:
javax.xml.registry Class
UnsupportedCapabilityException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.xml.registry.JAXRException
            ↓ javax.xml.registry.UnsupportedCapabilityException

All Implemented Interfaces:
   Serializable, JAXRResponse

public class UnsupportedCapabilityException
extends JAXRException

Extends: Throwable > Exception > JAXRException

JAXR JAXR API

See also  javax.xml.registry.RegistryService

This exception must be thrown when a JAXR client attempts to invoke an API method that is not supported by the capability profile that is supported by the JAXR provider.

Author:
   Farrukh S. Najmi

See Also:
   RegistryService, Serialized Form

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from class javax.xml.registry.JAXRException</th>
</tr>
</thead>
<tbody>
<tr>
<td>cause</td>
</tr>
</tbody>
</table>

### Fields inherited from interface javax.xml.registry.JAXRResponse

| STATUS FAILURE, STATUS SUCCESS, STATUS UNAVAILABLE, STATUS WARNING |

### Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UnsupportedCapabilityException()</strong></td>
<td>Constructs a JAXRException object with no reason or embedded Throwable.</td>
</tr>
<tr>
<td><strong>UnsupportedCapabilityException(String reason)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown.</td>
</tr>
<tr>
<td><strong>UnsupportedCapabilityException(String reason, Throwable cause)</strong></td>
<td>Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.</td>
</tr>
<tr>
<td><strong>UnsupportedCapabilityException(Throwable cause)</strong></td>
<td>Constructs a JAXRException object initialized with the given Throwable object.</td>
</tr>
</tbody>
</table>

### Method Summary

### Methods inherited from class javax.xml.registry.JAXRException

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getCause, getMessage, getRequestId, getStatus, initCause, isAvailable</td>
<td></td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Throwable

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getLocalizedMessage, getStackTrace, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString</td>
<td></td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
<td></td>
</tr>
</tbody>
</table>
public UnsupportedCapabilityException()
    Throwable JAXRException

UnsupportedCapabilityException

public UnsupportedCapabilityException()

    Constructs a JAXRException object with no reason or embedded Throwable.

public UnsupportedCapabilityException(String reason)
    JAXRException String reason

UnsupportedCapabilityException

public UnsupportedCapabilityException(String reason)

    Constructs a JAXRException object with the given String as the reason for the exception being thrown.

    Parameters:
        reason - a description of what caused the exception

public UnsupportedCapabilityException(String reason, Throwable cause)
    JAXRException String Throwable
    Throwable
        reason
        cause
        JAXRException Throwable
UnsupportedCapabilityException

public UnsupportedCapabilityException(String reason, Throwable cause)

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

Parameters:
reason - a description of what caused the exception
cause - a Throwable object that is to be embedded in this JAXRException object

public UnsupportedCapabilityException(Throwable cause)

Throws:
JAXRException

cause - the Throwable that caused this Exception
to license terms.

PS:
javax.activation Class UnsupportedDataTypeException

java.lang.Object
   ▼ java.lang.Throwable
       ▼ java.lang.Exception
           ▼ java.io.IOException
                ▼ javax.activation.UnsupportedDataTypeException

All Implemented Interfaces:
   Serializable

public class UnsupportedDataTypeException
  extends IOException

Extends: Throwable > Exception > java.io.IOException

See also javax.activation.DataHandler

Signals that the requested operation does not support the requested data type.

See Also: DataHandler, Serialized Form

---

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>UnsupportedDataTypeException()</td>
<td>Constructs an UnsupportedDataTypeException with no detail message.</td>
</tr>
<tr>
<td>UnsupportedDataTypeException(String s)</td>
<td>Constructs an UnsupportedDataTypeException with the specified message.</td>
</tr>
</tbody>
</table>
## Method Summary

Methods inherited from class java.lang.Throwables:
- fillInStackTrace
- getCause
- getLocalizedMessage
- getMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.Object:
- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

## Constructor Detail

**public UnsupportedDataTypeException()**

UnsupportedDataTypeException

**UnsupportedDataTypeException**

**public UnsupportedDataTypeException()**

UnsupportedDataTypeException

`Constructs an UnsupportedDataTypeException with no detail message.`

**public UnsupportedDataTypeException(String s)**

UnsupportedDataTypeException

**UnsupportedDataTypeException**

**public UnsupportedDataTypeException(String s)**

`Constructs an UnsupportedDataTypeException with the specified`
message.

**Parameters:**

s - The detail message.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/corporate/privacy.html).

PS:
javax.xml.registry.infomodel Interface URIValidator

All Known Subinterfaces:
   ExternalLink, ServiceBinding

public interface URIValidator

Implemented by: ExternalLink, ServiceBinding

URI

Defines common behavior expected of any class that validates URIs.

Author:
   Farrukh S. Najmi

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean validateURI()</td>
</tr>
<tr>
<td>void setValidateURI(boolean validate)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void setValidateURI(boolean validate) throws JAXRException</td>
</tr>
</tbody>
</table>

validate true JAXR URI URI false
Throws: **JAXRException**: JAXR

**setValidateURI**

```java
void setValidateURI(boolean validate)
throws JAXRException
```

Sets whether to do URI validation for this object. Default is true.

**Capability Level: 0**

**Parameters:**
- `validate` - true implies JAXR provider must perform validation of URIs when they are set; false implies validation is turned off

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

**public boolean getValidateURI() throws JAXRException**

```java
URI
```

0

```java
return true JAXR URI URI false
```

**Throws:** **JAXRException**: JAXR

**getValidateURI**

```java
boolean getValidateURI()
throws JAXRException
```

Gets whether to do URI validation for this object.

**Capability Level: 0**

**Returns:**
true implies JAXR provider must perform validation of URIs when they are set; false implies validation is turned off

Throws:

JAXRException - If the JAXR provider encounters an internal error

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.activation Class URLDataSource

java.lang.Object
   javax.activation.URLDataSource

All Implemented Interfaces:
   DataSource

Public class URLDataSource

extends Object
implements DataSource

Implements: DataSource

URLDataSource URL DataSource URLDataSource
JavaBeans Activation Framework URL DataHandler
URL DataHandler DataHandler URLDataSource

See also javax.activation.DataSource,
also javax.activation.DataHandler

The URLDataSource class provides an object that wraps a URL object in a
DataSource interface. URLDataSource simplifies the handling of data
described by URLs within the JavaBeans Activation Framework because
this class can be used to create new DataHandlers. NOTE: The
DataHandler object creates a URLDataSource internally, when it is
constructed with a URL.

See Also:
   DataSource, DataHandler

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>URLDataSource(URL url)</td>
</tr>
<tr>
<td>URLDataSource constructor.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getContentType()</code></td>
<td>Returns the value of the URL content-type header field.</td>
</tr>
<tr>
<td><code>InputStream getInputStream()</code></td>
<td>The <code>getInputStream</code> method from the URL.</td>
</tr>
<tr>
<td><code>String getName()</code></td>
<td>Calls the <code>getFile</code> method on the URL used to instantiate the object.</td>
</tr>
<tr>
<td><code>OutputStream getOutputStream()</code></td>
<td>The <code>getOutputStream</code> method from the URL.</td>
</tr>
<tr>
<td><code>URL getURL()</code></td>
<td>Return the URL used to create this DataSource.</td>
</tr>
</tbody>
</table>

### Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`<br>

### Constructor Detail

**public URLDataSource(java.net.URL url)**

**URLDataSource**

- `URL url`<br>

**URLDataSource**

**public URLDataSource(URL url)**

URLDataSource constructor. The URLDataSource class will not open a connection to the URL until a method requiring it to do so is called.
Parameters:

url - The URL to be encapsulated in this object.

### Method Detail

**public String getContentType()**

```java
URL - (content-type) URLConnection
URLConnection.openConnection
URLConnection.getContentType
"application/octet-stream"
return
```

**getContentType**

**public String getContentType()**

Returns the value of the URL content-type header field. It calls the URL's `URLConnection.getContentType` method after retrieving a `URLConnection` object. **Note: this method attempts to call the openConnection method on the URL. If this method fails, or if a content type is not returned from the URLConnection, getContentType returns "application/octet-stream" as the content type.**

**Specified by:**  
`getContentType` in interface `DataSource`

**Returns:**  
the content type.

**public String getName()**

```java
URL    return
getFile  URL    getFile
```
**getName**

```java
public String getName()
```

Calls the `getFile` method on the URL used to instantiate the object.

**Specified by:**
`getName` in interface `DataSource`

**Returns:**
the result of calling the URL's `getFile` method.

---

**public java.io.InputStream getInputStream() throws java.io.IOException**

```java
URL getInputStream URL
return InputStream
```

The `getInputStream` method from the URL. Calls the `openStream` method on the URL.

**Specified by:**
`getInputStream` in interface `DataSource`

**Returns:**
the InputStream.

**Throws:**
`IOException`

---

**public java.io.OutputStream getOutputStream() throws java.io.IOException**

```java
URL getOutputStream URL URLConnection
URLConnection getOutputStream URL
```

---
getOutputStream

public OutputStream getOutputStream()
throws IOException

The getOutputStream method from the URL. First an attempt is made to get the URLConnection object for the URL. If that succeeds, the getOutputStream method on the URLConnection is returned.

Specified by:
getOutputStream in interface DataSource

Returns:
the OutputStream.

Throws:
IOException

public java.net.URL getURL()

DataSource URL

return URL

getURL

public URL getURL()

Return the URL used to create this DataSource.

Returns:
The URL.
javax.mail  Class URLName

java.lang.Object
   _javax.mail.URLName

public class URLName
extends Object

URL URL Internet URL

   java.net.URL URL URL
   version 1.19, 07/05/04

The name of a URL. This class represents a URL name and also provides the basic parsing functionality to parse most internet standard URL schemes.

Note that this class differs from java.net.URL in that this class just represents the name of a URL, it does not model the connection to a URL.

Version:
   1.19, 07/05/04

Author:
   Christopher Cotton, Bill Shannon

Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected String</td>
<td>fullURL</td>
<td>The full version of the URL</td>
</tr>
</tbody>
</table>

Constructor Summary
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean equals(Object obj)</code></td>
<td>Compares two URLNames.</td>
</tr>
<tr>
<td><code>String getFile()</code></td>
<td>Returns the file name of this URLName.</td>
</tr>
<tr>
<td><code>String getHost()</code></td>
<td>Returns the host of this URLName.</td>
</tr>
<tr>
<td><code>String getPassword()</code></td>
<td>Returns the password of this URLName.</td>
</tr>
<tr>
<td><code>int getPort()</code></td>
<td>Returns the port number of this URLName.</td>
</tr>
<tr>
<td><code>String getProtocol()</code></td>
<td>Returns the protocol of this URLName.</td>
</tr>
<tr>
<td><code>String getRef()</code></td>
<td>Returns the reference of this URLName.</td>
</tr>
<tr>
<td><code>URL getURL()</code></td>
<td>Constructs a URL from the URLName.</td>
</tr>
<tr>
<td><code>String getUsername()</code></td>
<td>Returns the user name of this URLName.</td>
</tr>
<tr>
<td><code>int hashCode()</code></td>
<td>Compute the hash code for this URLName.</td>
</tr>
<tr>
<td><code>protected void parseString(String url)</code></td>
<td>Method which does all of the work of parsing the string.</td>
</tr>
<tr>
<td><code>String toString()</code></td>
<td>Constructs a string representation of this URLName.</td>
</tr>
</tbody>
</table>
Methods inherited from class java.lang.**Object**

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

---

**Field Detail**

**fullURL**

protected **String** fullURL

The full version of the URL

---

**Constructor Detail**

public **URLName**(String protocol, String host, int port, String file, String username, String password)

**URLName** -1 URL

**URLName**

public **URLName**(String protocol,

    String host,

    int port,

    String file,

    String username,

    String password)

Creates a URLName object from the specified protocol, host, port number, file, username, and password. Specifying a port number of -1 indicates that the URL should use the default port for the protocol.

public **URLName**(java.net.URL url)
java.net.URL  URLName

URLName

public URLName(URL url)

Construct a URLName from a java.net.URL object.

public URLName(String url)

Construct a URLName from the string. Parses out all the possible information (protocol, host, port, file, username, password).

Method Detail

public String toString()

toString

public String toString()

Constructs a string representation of this URLName.

Overrides:

toString in class Object
protected void parseString(String url)

parseString

protected void parseString(String url)

Method which does all of the work of parsing the string.

public int getPort()
URLName -1

getPort

public int getPort()

Returns the port number of this URLName. Returns -1 if the port is not set.

public String getProtocol()
URLName URLName null

getProtocol

public String getProtocol()

Returns the protocol of this URLName. Returns null if this URLName has no protocol.

public String getFile()
URLName URLName null
**getFileName**

```java
public String getFile()
```

Returns the file name of this URLName. Returns null if this URLName has no file name.

**getRef**

```java
public String getRef()
```

Returns the reference of this URLName. Returns null if this URLName has no reference.

**getHost**

```java
public String getHost()
```

Returns the host of this URLName. Returns null if this URLName has no host.
**getUsername**

```java
public String getUsername()

    Returns the user name of this URLName. Returns null if this
    URLName has no user name.
```

**public String getPassword()**

```java
URLName  URLName  null
```

**getPassword**

```java
public String getPassword()

    Returns the password of this URLName. Returns null if this
    URLName has no password.
```

**public java.net.URL getURL() throws java.net.MalformedURLException**

```java
URLName URLException  URL
```

**getURL**

```java
public URL getURL()

    Constructs a URL from the URLName.

    **Throws:**
    java.net.MalformedURLException
```

**public boolean equals(Object obj)**

```java
URLName null URLName  URLName  true
```
equals

public boolean equals(Object obj)

Compares two URLNames. The result is true if and only if the argument is not null and is a URLName object that represents the same URLName as this object. Two URLName objects are equal if they have the same protocol and the same host, the same port number on the host, the same username, and the same file on the host. The fields (host, username, file) are also considered the same if they are both null.

Hosts are considered equal if the names are equal (case independent) or if host name lookups for them both succeed and they both reference the same IP address.

Note that URLName has no knowledge of default port numbers for particular protocols, so "imap://host" and "imap://host:143" would not compare as equal.

Note also that the password field is not included in the comparison, nor is any reference field appended to the filename.

Overrides:

equals in class Object
public int hashCode()
URLName

hashCode

public int hashCode()

Compute the hash code for this URLName.

Overrides:
hashCode in class Object
public interface URLStats

extends Stats

Implements: Stats

URL

Specifies the statistics provided by a URL resource.
<table>
<thead>
<tr>
<th>SUMMARY: NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.registry.infomodel Interface User

All Superinterfaces:
ExtensibleObject, RegistryObject

public interface User
extends RegistryObject

Implements: RegistryObject

User RegistryObjectUser OrganizationUser
RegistryObject

See javax.xml.registry.infomodel.Organization,
also javax.xml.registry.infomodel.AuditableEvent

User instances are RegistryObjects that are used to provide information about registered users within the registry. Users are affiliated with Organizations. User objects are used in the audit trail for a RegistryObject.

Author:
Farrukh S. Najmi
See Also:
Organization, AuditableEvent

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Collection</strong></td>
</tr>
<tr>
<td><strong>Organization</strong></td>
</tr>
<tr>
<td><strong>PersonName</strong></td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
</tbody>
</table>
| **Collection**
  - **getPostalAddresses()**
    - Gets the postal address for this User. |
| **Collection**
  - **getTelephoneNumbers(String phoneType)**
    - Gets the telephone numbers for this User that match the specified telephone number type. |
| **String**
  - **getType()**
    - Gets the type for this User. |
| **URL**
  - **getUrl()**
    - Gets the URL to the web page for this User. |
| **void**
  - **setEmailAddresses(Collection emailAddresses)**
    - Sets the Collection of EmailAddress instances for this User. |
| **void**
  - **setPersonName(PersonName personName)**
    - Sets the name of this User. |
| **void**
  - **setPostalAddresses(Collection addresses)**
    - Sets the addresses for this User. |
| **void**
  - **setTelephoneNumbers(Collection phoneNumbers)**
    - Sets the various telephone numbers for this user. |
| **void**
  - **setType(String type)**
    - Sets the type for this User. |
| **void**
  - **setUrl(URL url)**
    - Sets the URL to the web page for this User. |

**Methods inherited from interface java.xml.registry.infomodel.RegistryObject**

- addAssociation, addAssociations, addClassification, addClassifications, addExternalIdentifier, addExternalIdentifiers, addExternalLink, addExternalLinks, getAssociatedObjects, getAssociations, getAuditTrail, getClassifications, getDescription, getExternalIdentifiers, getExternalLinks, getKey, getLifeCycleManager, getName, getObjectType, getRegistryPackages, getSubmittingOrganization, removeAssociation, removeAssociations, removeClassification, removeClassifications, removeExternalIdentifier, removeExternalIdentifiers, removeExternalLink, removeExternalLinks, setAssociations, setClassifications, setDescription, setExternalIdentifiers, setExternalLinks, setKey, setName, toXML
Methods inherited from interface javax.xml.registry.infomodel.ExtensibleObject
addSlot, addSlots, getSlot, getSlots, removeSlot, removeSlots

Method Detail

public Organization getOrganization() throws JAXRException
User Organization

0

getOrganization

Organization getOrganization()
throws JAXRException

Gets the Organization that this User is affiliated with.

Capability Level: 0

Returns:
the Organization that this User is affiliated with

Throws:
JAXRException - If the JAXR provider encounters an internal error
public PersonName getPersonName() throws JAXRException
User

Returns the name of this User.

Capability Level: 0

Returns: the name of this User

Throws: JAXRException - If the JAXR provider encounters an internal error

public void setPersonName(PersonName personName) throws JAXRException
User

Throws JAXRException: JAXR
setPersonName

void setPersonName(PersonName personName)
    throws JAXRException

    Sets the name of this User.

    Capability Level: 0

    Parameters:
        personName - the name of this User

    Throws:
        JAXRException - If the JAXR provider encounters an internal error

public java.util.Collection<E> getPostalAddresses() throws JAXRException
User

    return PostalAddress CollectionCollection null

    Throws
        JAXRException: JAXR

    supplierCardinality

    associates

    directed

    supplierRole

    clientCardinality

    See also
    javax.xml.registry.infomodel.PostalAddress

getPostalAddresses

Collection getPostalAddresses() throws JAXRException
Gets the postal address for this User.

**Capability Level: 0**

**Returns:**
Collection of PostalAddress instances. The Collection may be empty but not null.

**Throws:**
- [JAXRException](#) - If the JAXR provider encounters an internal error

**See Also:**
PostalAddress

```java
public void setPostalAddresses(java.util.Collection<? extends PostalAddress> addresses) throws JAXRException

User
0

addresses PostAddress Collection
Throws JAXRException: JAXR
```

**setPostalAddresses**

```java
void setPostalAddresses(Collection<PostalAddress> addresses) throws JAXRException

Sets the addresses for this User.

**Capability Level: 0**

**Parameters:**
addresses - Is a Collection of PostAddress instances.

**Throws:**
- [JAXRException](#) - If the JAXR provider encounters an internal error
public java.net.URL getUrl() throws JAXRException
User Web URL

1

    return User URL
    Throws JAXRException: JAXR

getUrl

URL getUrl() throws JAXRException

Gets the URL to the web page for this User.

Capability Level: 1

Returns:
the URL for this User's home page

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setUrl(java.net.URL url) throws JAXRException
User Web URL

1

    url User URL
    Throws JAXRException: JAXR

setUrl

void setUrl(URL url) throws JAXRException
Sets the URL to the web page for this User.

**Capability Level: 1**

**Parameters:**
- `url` - the URL for this User's home page

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

```java
public java.util.Collection<E> getTelephoneNumbers(String phoneType) throws JAXRException
User
```

```java
return null
```

**Throws**
- `JAXRException`: JAXR
- `supplierCardinality`: 1
- `associates`: `(){TelephoneNumber}
- `supplierRole`: `phones`
- `clientCardinality`: `1..*`

**See also**
- `javax.xml.registry.infomodel.TelephoneNumber`

**getTelephoneNumbers**

```java
Collection getTelephoneNumbers(String phoneType)
```

`throws JAXRException`

Gets the telephone numbers for this User that match the specified telephone number type.
Capability Level: 0

Parameters:
    phoneType - specifies the type of phone numbers to be returned. If phoneType is null, return all telephoneNumbers

Returns:
    Collection of TelephoneNumber instances. The Collection may be empty but not null.

Throws:
    JAXRException - If the JAXR provider encounters an internal error

See Also:
    TelephoneNumber

```
public void setTelephoneNumbers(java.util.Collection<E> phoneNumbers) throws JAXRException
```

0

```
phoneNumbers  TelephoneNumber  Collection
Throws         JAXRException: JAXR
```

setTelephoneNumbers

```
void setTelephoneNumbers(Collection phoneNumbers) throws JAXRException
```

Sets the various telephone numbers for this user.

Capability Level: 0

Parameters:
    phoneNumbers - the Collection of TelephoneNumbers to be set

Throws:
    JAXRException - If the JAXR provider encounters an internal error
public java.util.Collection<E> getEmailAddresses() throws JAXRException

User

0

return EmailAddress CollectionCollection null

Throws JAXRException: JAXR

supplierCardinality 1

associates <EmailAddress>

directed

supplierRole emailAddresses

clientCardinality 0..*

See also javax.xml.registry.infomodel.EmailAddress

getEmailAddresses

Collection getEmailAddresses() throws JAXRException

Gets the email addresses for this User.

Capability Level: 0

Returns:
Collection of EmailAddress instances. The Collection may be empty but not null.

Throws:
JAXRException - If the JAXR provider encounters an internal error

See Also:
EmailAddress
public void setEmailAddresses(java.util.Collection<E> emailAddresses) throws JAXRException
User EmailAddress Collection

void setEmailAddresses(Collection emailAddresses)
throws JAXRException

Sets the Collection of EmailAddress instances for this User.

Capability Level: 0

Parameters:
   emailAddresses - the Collection of EmailAddresses to be set

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public String getType() throws JAXRException
User NULL

String getType

return User String

Throws JAXRException: JAXR
String getType() throws JAXRException

Gets the type for this User. Default is a NULL String.

Capability Level: 0

Returns:
the type for this User, which is an arbitrary String

Throws:
JAXRException - If the JAXR provider encounters an internal error

public void setType(String type) throws JAXRException

User

0

type User String

Throws JAXRException: JAXR

setType

void setType(String type)
throws JAXRException

Sets the type for this User.

Capability Level: 0

Parameters:
  type - the type for this User, which is an arbitrary String

Throws:
  JAXRException - If the JAXR provider encounters an internal error
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.transaction  Interface UserTransaction

public interface UserTransaction

UserTransaction

The UserTransaction interface defines the methods that allow an application to explicitly manage transaction boundaries.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void begin()</td>
<td>Create a new transaction and associate it with the current thread.</td>
</tr>
<tr>
<td>void commit()</td>
<td>Complete the transaction associated with the current thread.</td>
</tr>
<tr>
<td>int getStatus()</td>
<td>Obtain the status of the transaction associated with the current thread.</td>
</tr>
<tr>
<td>void rollback()</td>
<td>Roll back the transaction associated with the current thread.</td>
</tr>
<tr>
<td>void setRollbackOnly()</td>
<td>Modify the transaction associated with the current thread such that the only possible outcome of the transaction is to rollback the transaction.</td>
</tr>
<tr>
<td>void setTransactionTimeout(int seconds)</td>
<td>Modify the timeout value that is associated with transactions started by the current thread with the begin method.</td>
</tr>
</tbody>
</table>
public void begin() throws NotSupportedException, SystemException

Throws NotSupportedException: TransactionManager
Throws SystemException:

begin

void begin() throws NotSupportedException, SystemException

Create a new transaction and associate it with the current thread.

Throws:
- NotSupportedException - Thrown if the thread is already associated with a transaction and the Transaction Manager implementation does not support nested transactions.
- SystemException - Thrown if the transaction manager encounters an unexpected error condition.

public void commit() throws RollbackException, HeuristicMixedException, HeuristicRollbackException, SecurityException, IllegalStateException, SystemException

Throws RollbackException:
Throws HeuristicMixedException:
Throws HeuristicRollbackException:
Throws SecurityException:
Throws IllegalStateException:
Throws SystemException:

commit
void commit()
    throws RollbackException,
            HeuristicMixedException,
            HeuristicRollbackException,
            SecurityException,
            IllegalStateException,
            SystemException

Complete the transaction associated with the current thread. When
this method completes, the thread is no longer associated with a
transaction.

Throws:
    RollbackException - Thrown to indicate that the transaction has
been rolled back rather than committed.
    HeuristicMixedException - Thrown to indicate that a heuristic
decision was made and that some relevant updates have been
committed while others have been rolled back.
    HeuristicRollbackException - Thrown to indicate that a heuristic
decision was made and that all relevant updates have been
rolled back.
    SecurityException - Thrown to indicate that the thread is not
allowed to commit the transaction.
    IllegalStateException - Thrown if the current thread is not
associated with a transaction.
    SystemException - Thrown if the transaction manager encounters
an unexpected error condition.

public void rollback() throws IllegalStateException,
            SecurityException, SystemException

    Throws SecurityException: 
    Throws IllegalStateException: 
    Throws SystemException: 

rollback
void rollback() throws IllegalStateException, SecurityException, SystemException

Roll back the transaction associated with the current thread. When this method completes, the thread is no longer associated with a transaction.

Throws:
SecurityException - Thrown to indicate that the thread is not allowed to roll back the transaction.
IllegalStateException - Thrown if the current thread is not associated with a transaction.
SystemException - Thrown if the transaction manager encounters an unexpected error condition.

public void setRollbackOnly() throws IllegalStateException, SystemException

Throws
IllegalStateException:
SystemException:

setRollbackOnly

void setRollbackOnly() throws IllegalStateException, SystemException

Modify the transaction associated with the current thread such that the only possible outcome of the transaction is to roll back the transaction.

Throws:
IllegalStateException - Thrown if the current thread is not associated with a transaction.
SystemException - Thrown if the transaction manager encounters an unexpected error condition.
public int getStatus() throws SystemException

    return Status.NoTransaction

Throws SystemException:

getStatus

int getStatus() throws SystemException

    Obtain the status of the transaction associated with the current thread.

    Returns:
    The transaction status. If no transaction is associated with the current thread, this method returns the Status.NoTransaction value.

    Throws:
    SystemException - Thrown if the transaction manager encounters an unexpected error condition.

public void setTransactionTimeout(int seconds) throws SystemException

begin

    seconds 0 SystemException

Throws SystemException:

setTransactionTimeout

void setTransactionTimeout(int seconds)
throws SystemException

Modify the timeout value that is associated with transactions started by the current thread with the begin method.

If an application has not called this method, the transaction service uses some default value for the transaction timeout.

Parameters:
seconds - The value of the timeout in seconds. If the value is zero, the transaction service restores the default value. If the value is negative a SystemException is thrown.

Throws:
SystemException - Thrown if the transaction manager encounters an unexpected error condition.
public interface ValidatingManagedConnectionFactory

ManagedConnection ManagedConnectionFactory

ManagedConnection

version 1.0

This interface is implemented by a ManagedConnectionFactory instance that supports the ability to validate ManagedConnection objects.

This may be used by the application server to prune invalid ManagedConnection objects from its connection pool.

Version: 1.0
Author: Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set getInvalidConnections(Set connectionSet)</td>
</tr>
</tbody>
</table>

This method returns a set of invalid ManagedConnection objects chosen from a specified set of ManagedConnection objects.

Method Detail

public java.util.Set<E>
getInvalidConnections(java.util.Set<E> connectionSet)
throws ResourceException

ManagedConnection connectionSet
return

Throws

getInvalidConnections

Set getInvalidConnections(Set connectionSet)
throws ResourceException

This method returns a set of invalid ManagedConnection objects chosen from a specified set of ManagedConnection objects.

Parameters:
connectionSet - a set of ManagedConnection objects that need to be validated.

Returns:
a set of invalid ManagedConnection objects.

Throws:
ResourceException - generic exception.
javax.xml.bind Interface ValidationEvent

All Known Subinterfaces:
   NotIdentifiableEvent, ParseConversionEvent, PrintConversionEvent

All Known Implementing Classes:
   NotIdentifiableEventImpl, ParseConversionEventImpl, PrintConversionEventImpl, ValidationEventImpl

public interface ValidationEvent

Implemented by: NotIdentifiableEvent, ParseConversionEvent, PrintConversionEvent, ValidationEventImpl

This event indicates that a problem was encountered while validating the incoming XML data during an unmarshal operation, while performing on-demand validation of the Java content tree, or while marshalling the Java content tree back to XML data.

Since:
   JAXB1.0

Version:
   $Revision: 1.1 $

Author:
   - Ryan Shoemaker, Sun Microsystems, Inc.
   - Kohsuke Kawaguchi, Sun Microsystems, Inc.
   - Joe Fialli, Sun Microsystems, Inc.

See Also:
   Validator, ValidationEventHandler
Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR</td>
<td>Conditions that correspond to the definition of &quot;error&quot; in section 1.2 of the W3C XML 1.0 Recommendation</td>
</tr>
<tr>
<td>FATAL_ERROR</td>
<td>Conditions that correspond to the definition of &quot;fatal error&quot; in section 1.2 of the W3C XML 1.0 Recommendation</td>
</tr>
<tr>
<td>WARNING</td>
<td>Conditions that are not errors or fatal errors as defined by the XML 1.0 recommendation</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getLinkedException()</td>
<td>Retrieve the linked exception for this warning/error.</td>
</tr>
<tr>
<td>getLocator()</td>
<td>Retrieve the locator for this warning/error.</td>
</tr>
<tr>
<td>getMessage()</td>
<td>Retrieve the text message for this warning/error.</td>
</tr>
<tr>
<td>getSeverity()</td>
<td>Retrieve the severity code for this warning/error.</td>
</tr>
</tbody>
</table>

Field Detail

WARNING

static final int WARNING
Conditions that are not errors or fatal errors as defined by the XML 1.0 recommendation

See Also:
Constant Field Values

ERROR
static final int ERROR

Conditions that correspond to the definition of "error" in section 1.2 of the W3C XML 1.0 Recommendation

See Also:
Constant Field Values

FATAL_ERROR
static final int FATAL_ERROR

Conditions that correspond to the definition of "fatal error" in section 1.2 of the W3C XML 1.0 Recommendation

See Also:
Constant Field Values

Method Detail

public int getSeverity()
/

ValidationError.WARNINGValidationError.ERROR
getSeverity

```java
int getSeverity()

Retrieve the severity code for this warning/error.

Must be one of ValidationError.WARNING, ValidationError.ERROR, or ValidationError.FATAL_ERROR.

Returns:
the severity code for this warning/error
```

public String getMessage()
/

```java
return / null
```

getMessage

```java
String getMessage()

Retrieve the text message for this warning/error.

Returns:
the text message for this warning/error or null if one wasn't set
```

public Throwable getLinkedException()
/

```java
return / null
```
getLinkedException

Throwable getLinkedException()

Retrieve the linked exception for this warning/error.

Returns:
the linked exception for this warning/error or null if one wasn't set

public ValidationEventLocator getLocator()
/

    return /

getLocator

ValidationEventLocator getLocator()

Retrieve the locator for this warning/error.

Returns:
the locator that indicates where the warning/error occurred
javax.xml.bind.util  **Class ValidationEventCollector**

java.lang.Object
   └─javax.xml.bind.util.ValidationEventCollector

**All Implemented Interfaces:**
   ValidationEventHandler

---

```java
public class ValidationEventCollector
    extends Object
    implements ValidationEventHandler

Implements: ValidationEventHandler

ValidatorUnmarshaller  Marshaller  setEventHandler
getEvents
```

version  $Revision: 1.2 $  en
since  JAXB1.0

See  javax.xml.bind.Validator,
    javax.xml.bind.ValidationEventHandler,
    javax.xml.bind.ValidationEvent,
    javax.xml.bind.ValidationEventLocator

also

**ValidationEventCollector** implementation that collects all events.

To use this class, create a new instance and pass it to the
setEventHandler method of the Validator, Unmarshaller, Marshaller class.
After the call to validate or unmarshal completes, call the getEvents
method to retrieve all the reported errors and warnings.

Since:
   JAXB1.0
Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ValidationEventCollector()</code></td>
<td>Constructor</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getEvents()</code></td>
<td>Return an array of ValidationEvent objects containing a copy of each of the collected errors and warnings.</td>
</tr>
<tr>
<td><code>handleEvent(ValidationEvent event)</code></td>
<td>Receive notification of a validation warning or error.</td>
</tr>
<tr>
<td><code>hasEvents()</code></td>
<td>Returns true if this event collector contains at least one ValidationEvent.</td>
</tr>
<tr>
<td><code>reset()</code></td>
<td>Clear all collected errors and warnings.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`
**Constructor Detail**

public ValidationEventCollector()

**Method Detail**

public ValidationEvent[] getEvents()

return getEvents

getEvents

public ValidationEvent[] getEvents()

Return an array of ValidationEvent objects containing a copy of each of the collected errors and warnings.

**Returns:**

a copy of all the collected errors and warnings or an empty array if there weren't any

public void reset()

reset

public void reset()
Clear all collected errors and warnings.

```java
public boolean hasEvents()
    ValidationEvent true
    return ValidationEvent true false
```

### hasEvents

```java
public boolean hasEvents()
```

Returns `true` if this event collector contains at least one `ValidationEvent`.

**Returns:**

- `true` if this event collector contains at least one `ValidationEvent`,
- `false` otherwise

```java
public boolean handleEvent(ValidationEvent event)
```

### handleEvent

```java
public boolean handleEvent(ValidationEvent event)
```

**Description copied from interface:** `ValidationEventHandler`  
Receive notification of a validation warning or error. The `ValidationEvent` will have a `ValidationEventLocator` embedded in it that indicates where the error or warning occurred.

If an unchecked runtime exception is thrown from this method, the JAXB provider will treat it as if the method returned false and interrupt the current unmarshal, validate, or marshal operation.

**Specified by:**

- `handleEvent` in interface `ValidationEventHandler`
Parameters:

- **event** - the encapsulated validation event information. It is a provider error if this parameter is null.

Returns:

- true if the JAXB Provider should attempt to continue the current unmarshal, validate, or marshal operation after handling this warning/error, false if the provider should terminate the current operation with the appropriate UnmarshalException, ValidationException, or MarshalException.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>SUMMARY</th>
<th>NESTED</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
<th>DETAIL</th>
<th>FIELD</th>
<th>CONSTR</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>FRAMES</td>
</tr>
</tbody>
</table>

Overview  Package  Tree  Deprecated  Index  Help
javax.xml.bind Interface ValidationEventHandler

All Known Implementing Classes:
  _DefaultValidationEventHandler, _ValidationEventCollector

public interface ValidationEventHandler

Implemented by: _DefaultValidationEventHandler,
_ValidationEventCollector

handleEvent JAXB false

Java

handleEvent false

Validator javadocs

version $Revision: 1.1 $ en
since JAXB1.0

javax.xml.bind.Unmarshaller, javax.xml.bind.Validator,
javax.xml.bind.Marshaller,
javax.xml.bind.ValidationEvent,
javax.xml.bind.util.ValidationEventCollector

A basic event handler interface for validation errors.

If an application needs to implement customized event handling, it must
implement this interface and then register it with either the Unmarshaller,
the Validator, or the Marshaller. The JAXB Provider will then report
validation errors and warnings encountered during the unmarshal, marshal, and validate operations to these event handlers.

If the `handleEvent` method throws an unchecked runtime exception, the JAXB Provider must treat that as if the method returned false, effectively terminating whatever operation was in progress at the time (unmarshal, validate, or marshal).

Modifying the Java content tree within your event handler is undefined by the specification and may result in unexpected behaviour.

Failing to return false from the `handleEvent` method after encountering a fatal error is undefined by the specification and may result in unexpected behavior.

**Default Event Handler**

See: [Validator javadocs](#)

**Since:**

JAXB1.0

**Version:**

$Revision: 1.1 $

**Author:**

- Ryan Shoemaker, Sun Microsystems, Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems, Inc.

**See Also:**

[Unmarshaller](#), [Validator](#), [Marshaller](#), [ValidationEvent](#), [ValidationEventCollector](#)

---

**Method Summary**

```java
boolean handleEvent(ValidationEvent event)
```

Receive notification of a validation warning or error.

---

**Method Detail**
public boolean handleEvent(ValidationEvent event)

ValidationEvent

JAXB false

<table>
<thead>
<tr>
<th>event</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXB</td>
<td>true</td>
</tr>
</tbody>
</table>

return UnmarshalExceptionValidationException MarshalException false

Throws IllegalArgumentException: null

handleEvent

boolean handleEvent(ValidationEvent event)

Receive notification of a validation warning or error. The ValidationEvent will have a ValidationEventLocator embedded in it that indicates where the error or warning occurred.

If an unchecked runtime exception is thrown from this method, the JAXB provider will treat it as if the method returned false and interrupt the current unmarshal, validate, or marshal operation.

Parameters:

  event - the encapsulated validation event information. It is a provider error if this parameter is null.

Returns:

  true if the JAXB Provider should attempt to continue the current unmarshal, validate, or marshal operation after handling this warning/error, false if the provider should terminate the current operation with the appropriate UnmarshalException, ValidationException, or MarshalException.

Throws:

  IllegalArgumentException - if the event object is null.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.helpers Class ValidationEventImpl

java.lang.Object  
  javax.xml.bind.helpers.ValidationEventImpl

All Implemented Interfaces:
  ValidationEvent

Direct Known Subclasses:
  NotIdentifiableEventImpl, ParseConversionEventImpl, PrintConversionEventImpl

public class ValidationEventImpl

extends Object
implements ValidationEvent

Implements: ValidationEvent
Extended by: NotIdentifiableEventImpl, ParseConversionEventImpl, PrintConversionEventImpl

ValidationEvent

JAXB ValidationEvent

version $Revision: 1.1 $ en
since JAXB1.0

See also
javax.xml.bind_VALIDATOR,
javax.xml.bind.ValidationEventHandler,
javax.xml.bind.ValidationEvent,
javax.xml.bind.ValidationEventLocator

Default implementation of the ValidationEvent interface.

JAXB providers are allowed to use whatever class that implements the ValidationEvent interface. This class is just provided for a convenience.
Since: JAXB1.0
Version: $Revision: 1.1 $
Author:
  - Kohsuke Kawaguchi, Sun Microsystems, Inc.
See Also:
  Validator, ValidationEventHandler, ValidationEvent, ValidationEventLocator

Field Summary

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.xml.bind.ValidationEvent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERROR, FATAL_ERROR, WARNING</td>
</tr>
</tbody>
</table>

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValidationEventImpl(int _severity, String _message, ValidationEventLocator _locator)</td>
<td>Create a new ValidationEventImpl.</td>
</tr>
<tr>
<td>ValidationEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)</td>
<td>Create a new ValidationEventImpl.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throwable getLinkedException()</td>
<td>Retrieve the linked exception for this warning/error.</td>
</tr>
<tr>
<td>ValidationEventLocator getLocator()</td>
<td>Retrieve the locator for this warning/error.</td>
</tr>
<tr>
<td>String getMessage()</td>
<td>Retrieve the text message for this warning/error.</td>
</tr>
<tr>
<td>getSeverity()</td>
<td></td>
</tr>
</tbody>
</table>
int setSeverity(int _severity)
Set the severity field of this event.

void setLinkedException(Throwable _linkedException)
Set the linked exception field of this event.

void setLocator(ValidationEventLocator _locator)
Set the locator object for this event.

void setMessage(String _message)
Set the message field of this event.

void setLocator(ValidationEventLocator _locator)
Set the locator object for this event.

String toString()
Returns a string representation of this object in a format helpful to debugging.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

public ValidationEventImpl(int _severity, String _message, ValidationEventLocator _locator)
ValidationEventImpl

_severity
severity ValidationEvent.WARNING
ValidationEvent.ERROR ValidationEvent.FATAL_ERROR

_message
Locator null

_locator
null

Throws IllegalArgumentException: severity

ValidationEventImpl

public ValidationEventImpl(int _severity,
String _message,
Create a new ValidationEventImpl.

**Parameters:**

- `_severity` - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR
- `_message` - The text message for this event - may be null.
- `_locator` - The locator object for this event - may be null.

**Throws:**

- `IllegalArgumentException` - if an illegal severity field is supplied

```java
public ValidationEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)
```

Create a new ValidationEventImpl.

**Parameters:**

- `_severity` - The severity value for this event. Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR

```java
public ValidationEventImpl(int _severity, String _message, ValidationEventLocator _locator, Throwable _linkedException)
```
ValidationEvent.FATAL_ERROR

_message - The text message for this event - may be null.
_locator - The locator object for this event - may be null.
_linkedException - An optional linked exception that may provide additional information about the event - may be null.

Throws:
IllegalArgumentException - if an illegal severity field is supplied

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
</table>

**public int getSeverity()**

**getSeverity**

**public int getSeverity()**

*Description copied from interface: ValidationEvent*
Retrieve the severity code for this warning/error.

Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR.

*Specified by:*
setSeverity in interface ValidationEvent

**Returns:**
the severity code for this warning/error

**public void setSeverity(int _severity)**

**severity**

_severity  ValidationEvent.WARNING
ValidationEvent.ERROR
ValidationEvent.FATAL_ERROR

**Throws**
IllegalArgumentException: severity

**setSeverity**
public void setSeverity(int _severity)

Set the severity field of this event.

**Parameters:**

_severity - Must be one of ValidationEvent.WARNING, ValidationEvent.ERROR, or ValidationEvent.FATAL_ERROR.

**Throws:**

IllegalArgumentException - if an illegal severity field is supplied

---

public String getMessage()

**getMessage**

public String getMessage()

Description copied from interface: ValidationEvent
Retrieve the text message for this warning/error.

**Specified by:**

getMessage in interface ValidationEvent

**Returns:**

the text message for this warning/error or null if one wasn't set

---

public void setMessage(String _message)

setMessage

public void setMessage(String _message)

Set the message field of this event.

**Parameters:**
_message - String message - may be null.

public Throwable getLinkedException()

getLinkedException

public Throwable getLinkedException()

Description copied from interface: ValidationEvent
Retrieve the linked exception for this warning/error.

Specified by:
getLinkedException in interface ValidationEvent

Returns:
the linked exception for this warning/error or null if one wasn't set

public void setLinkedException(Throwable _linkedException)

_setLinkedException

_setLinkedException

null

setLinkedException

public void setLinkedException(Throwable _linkedException)

Set the linked exception field of this event.

Parameters:
_linkedException - Optional linked exception - may be null.

public ValidationEventLocator getLocator()
getLocator

public ValidationEventLocator getLocator()

Description copied from interface: ValidationEvent
Retrieve the locator for this warning/error.

Specified by:
   getLocator in interface ValidationEvent

Returns:
   the locator that indicates where the warning/error occurred

public void setLocator(ValidationEventLocator _locator)

   _locator null

setLocator

public void setLocator(ValidationEventLocator _locator)

   Set the locator object for this event.

   Parameters:
   _locator - The locator - may be null.

public String toString()

   See also            equals(Object)

toString

public String toString()

   Returns a string representation of this object in a format helpful to
debugging.

Overrides:

`toString` in class `Object`

See Also:

`Object.equals(Object)`
**javax.xml.bind** Interface **ValidationEventLocator**

All Known Implementing Classes:
  
  [ValidationEventLocatorImpl](#)

---

```java
public interface ValidationEventLocator

Implemented by: [ValidationEventLocatorImpl](#)

ValidationEvent

  ValidationEventLocator ValidationEvent on-demand validation Java unmarshal-time validation XML url Node

  **version** $Revision: 1.1 $**

  **since** JAXB1.0

See also [javax.xml.bind.Validator](#), [javax.xml.bind.ValidationEvent](#)
```

Encapsulate the location of a ValidationEvent.

The ValidationEventLocator indicates where the ValidationEvent occurred. Different fields will be set depending on the type of validation that was being performed when the error or warning was detected. For example, on-demand validation would produce locators that contained references to objects in the Java content tree while unmarshal-time validation would produce locators containing information appropriate to the source of the XML data (file, url, Node, etc).

Since:
  
  JAXB1.0

Version:
  
  $Revision: 1.1 $

Author:
  
  - Ryan Shoemaker, Sun Microsystems, Inc.
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int <code>getColumnNumber()</code></td>
<td>Return the column number if available</td>
</tr>
<tr>
<td>int <code>getLineNumber()</code></td>
<td>Return the line number if available</td>
</tr>
<tr>
<td>Node <code>getNode()</code></td>
<td>Return a reference to the DOM Node if available</td>
</tr>
<tr>
<td><code>Object</code> <code>getObject()</code></td>
<td>Return a reference to the object in the Java content tree if available</td>
</tr>
<tr>
<td>int <code>getOffset()</code></td>
<td>Return the byte offset if available</td>
</tr>
<tr>
<td>URL <code>getURL()</code></td>
<td>Return the name of the XML source as a URL if available</td>
</tr>
</tbody>
</table>

Method Detail

```
public java.net.URL `getURL()`

URL XML
return URL XML null
```

getURL

```
URL `getURL()`

Return the name of the XML source as a URL if available
```
Returns:
the name of the XML source as a URL or null if unavailable

public int getOffset()

    return -1

getOffset

int getOffset()

    Return the byte offset if available

    Returns:
    the byte offset into the input source or -1 if unavailable

public int getLineNumber()

    return -1

getLineNumber

int getLineNumber()

    Return the line number if available

    Returns:
    the line number or -1 if unavailable

public int getColumnNumber()

    return -1
getColumnNumber

int getColumnNumber()

Return the column number if available

Returns:
   the column number or -1 if unavailable

public Object getObject()
Java

    return Java null

getObject

Object getObject()

Return a reference to the object in the Java content tree if available

Returns:
   a reference to the object in the Java content tree or null if unavailable

public org.w3c.dom.Node getNode()
DOM Node

    return DOM Node null

getNode

Node getNode()

Return a reference to the DOM Node if available
Returns:
a reference to the DOM Node or null if unavailable

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.helpers  **Class ValidationEventLocatorImpl**

`java.lang.Object`  
  `javax.xml.bind.helpers.ValidationEventLocatorImpl`

**All Implemented Interfaces:**

  [ValidationEventLocator](javax.xml.bind.ValidationEventLocator)

---

```java
public class ValidationEventLocatorImpl extends Object implements ValidationEventLocator

Implements: ValidationEventLocator

ValidationEventLocator

JAXB ValidationEventLocator

version $Revision: 1.2 $

since JAXB1.0

See also [javax.xml.bind.Validator](javax.xml.bind.Validator), [javax.xml.bind.ValidationEventHandler](javax.xml.bind.ValidationEventHandler), [javax.xml.bind.ValidationEvent](javax.xml.bind.ValidationEvent), [javax.xml.bind.ValidationEventLocator](javax.xml.bind.ValidationEventLocator)
```

Default implementation of the ValidationEventLocator interface.

JAXB providers are allowed to use whatever class that implements the ValidationEventLocator interface. This class is just provided for a convenience.

**Since:**

JAXB1.0

**Version:**

$Revision: 1.2 $
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ValidationEventLocatorImpl()</code></td>
<td>Creates an object with all fields unavailable.</td>
</tr>
<tr>
<td><code>ValidationEventLocatorImpl(Locator loc)</code></td>
<td>Constructs an object from an org.xml.sax.Locator.</td>
</tr>
<tr>
<td><code>ValidationEventLocatorImpl(Node _node)</code></td>
<td>Constructs an object that points to a DOM Node.</td>
</tr>
<tr>
<td><code>ValidationEventLocatorImpl(Object _object)</code></td>
<td>Constructs an object that points to a JAXB content object.</td>
</tr>
<tr>
<td><code>ValidationEventLocatorImpl(SAXParseException e)</code></td>
<td>Constructs an object from the location information of a SAXParseException.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getColumnNumber()</code></td>
<td>Return the column number if available</td>
</tr>
<tr>
<td><code>getLineNumber()</code></td>
<td>Return the line number if available</td>
</tr>
<tr>
<td><code>getNode()</code></td>
<td>Return a reference to the DOM Node if available</td>
</tr>
<tr>
<td><code>getObject()</code></td>
<td>Return a reference to the object in the Java content tree if available</td>
</tr>
<tr>
<td><code>getOffset()</code></td>
<td>Return the byte offset if available</td>
</tr>
<tr>
<td><code>getURL()</code></td>
<td>Return the name of the XML source as a URL if available</td>
</tr>
</tbody>
</table>
void setColumnNumber(int _columnNumber)
    Set the columnNumber field on this event locator.

void setLineNumber(int _lineNumber)
    Set the lineNumber field on this event locator.

void setNode(Node _node)
    Set the Node field on this event locator.

void setObject(Object _object)
    Set the Object field on this event locator.

void setOffset(int _offset)
    Set the offset field on this event locator.

void setURL(URL _url)
    Set the URL field on this event locator.

String toString()
    Returns a string representation of this object in a format helpful to debugging.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait

Constructor Detail

public ValidationEventLocatorImpl()
**ValidationEventLocatorImpl**

```java
public ValidationEventLocatorImpl(Locator loc)

Constructs an object from an org.xml.sax.Locator. The object's ColumnNumber, LineNumber, and URL become available from the values returned by the locator's getColumnNumber(), getLineNumber(), and getSystemId() methods respectively. Node, Object, and Offset are not available.

**Parameters:**
- **loc** - the SAX Locator object that will be used to populate this event locator.

**Throws:**
- **IllegalArgumentException** - if the Locator is null

public ValidationEventLocatorImpl(org.xml.sax.SAXParseException e)

SAXParseException ColumnNumber
LineNumber URL Locator getColumnNumber() getLineNumber() getSystemId() NodeObject Offset
e SAXParseException

**Throws**
- **IllegalArgumentException** - SAXParseException null
ValidationEventLocatorImpl

public ValidationEventLocatorImpl(SAXParseException e)

Constructs an object from the location information of a SAXParseException. The object's ColumnNumber, LineNumber, and URL become available from the values returned by the locator's getColumnNumber(), getLineNumber(), and getSystemId() methods respectively. Node, Object, and Offset are not available.

Parameters:
e - the SAXParseException object that will be used to populate this event locator.

Throws:
IllegalArgumentException - if the SAXParseException is null

public ValidationEventLocatorImpl(org.w3c.dom.Node _node)

DOM Node ColumnNumber
LineNumber ObjectOffset URL
_node DOM Node

Throws
IllegalArgumentException: Node null

ValidationEventLocatorImpl

public ValidationEventLocatorImpl(Node _node)

Constructs an object that points to a DOM Node. The object's Node becomes available. ColumnNumber, LineNumber, Object, Offset, and URL are not available.

Parameters:
_node - the DOM Node object that will be used to populate this event locator.

Throws:
IllegalArgumentException - if the Node is null
ValidationEventLocatorImpl

public ValidationEventLocatorImpl(Object _object)

Constructs an object that points to a JAXB content object. The object's Object becomes available. ColumnNumber, LineNumber, Node, Offset, and URL are not available.

Parameters:
   _object - the Object that will be used to populate this event locator.

Throws:
   IllegalArgumentException - if the Object is null

Method Detail

getURL

public java.net.URL getURL()

See also getURL()
Returns:
the name of the XML source as a URL or null if unavailable

See Also:
ValidationEventLocator.getURL()

---

class ValidationEventLocator {
    public void setURL(java.net.URL _url) {
        URL _url;
    }

    public void setURL(URL _url) {
        Set the URL field on this event locator. Null values are allowed.
        Parameters:
        _url - the url
    }

    public int getOffset() {
        See also
        getOffset()
    }

    public int getOffset() {
        Description copied from interface: ValidationEventLocator
        Return the byte offset if available
        Specified by:
        getOffset in interface ValidationEventLocator
        Returns:
        the byte offset into the input source or -1 if unavailable
        See Also:
        ValidationEventLocator.getOffset()
    }
}
public void setOffset(int _offset)
  
  offset
  
  _offset

setOffset

public void setOffset(int _offset)

  Set the offset field on this event locator.

  Parameters:
  
  _offset - the offset

public int getLineNumber()

  See also

  getLineNumber()

getLineNumber

public int getLineNumber()

  Description copied from interface: ValidationEventLocator
  Return the line number if available

  Specified by:
  
  getLineNumber in interface ValidationEventLocator

  Returns:
  
  the line number or -1 if unavailable

  See Also:
  
  ValidationEventLocator.getLineNumber()

public void setLineNumber(int _lineNumber)

  lineNumber
  
  _lineNumber
**setLineNumber**

```java
public void setLineNumber(int _lineNumber)
```

Set the lineNumber field on this event locator.

**Parameters:**

_`lineNumber` - the line number

---

**public int getColumnNumber()**

See also

getColumnNumber()

**getColumnNumber**

```java
public int getColumnNumber()
```

**Description copied from interface:** `ValidationEventLocator`

Return the column number if available

**Specified by:**

`getColumnNumber` in interface `ValidationEventLocator`

**Returns:**

the column number or -1 if unavailable

See Also:

`ValidationEventLocator.getColumnNumber()`

---

**public void setColumnNumber(int _columnNumber)**

```java
_columnNumber
```

**setColumnNumber**
public void setColumnNumber(int _columnNumber)

Set the columnNumber field on this event locator.

Parameters:
_columnNumber - the column number

---

public Object getObject()

See also

getObject()

gGetObject

public Object getObject()

Description copied from interface: ValidationEventLocator
Return a reference to the object in the Java content tree if available

Specified by:

gGetObject in interface ValidationEventLocator

Returns:
a reference to the object in the Java content tree or null if unavailable

See Also:

ValidationEventLocator.getObject()

---

public void setObject(Object _object)

Object null

_setObject

java

setObject

public void setObject(Object _object)

Set the Object field on this event locator. Null values are allowed.
Parameters:
_object - the java content object

public org.w3c.dom.Node getNode()

See also
getNode()

getNode

public Node getNode()

Description copied from interface: ValidationEventLocator
Return a reference to the DOM Node if available

Specified by:
getNode in interface ValidationEventLocator

Returns:
a reference to the DOM Node or null if unavailable

See Also:
ValidationEventLocator.getNode()

public void setNode(org.w3c.dom.Node _node)
Node null

_setNode

setNode

public void setNode(Node node)

Set the Node field on this event locator. Null values are allowed.

Parameters:
_node - the Node
public String toString()

See also

equals(Object)

toString

public String toString()

Returns a string representation of this object in a format helpful to debugging.

Overrides:

toString in class Object

See Also:

Object.equals(Object)
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
</tr>
</tbody>
</table>
javax.xml.bind  **Class ValidationException**

```java
java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.xml.bind.JAXBException
              └ javax.xml.bind.ValidationException

All Implemented Interfaces:
  Serializable
```

```java
public class ValidationException
    extends JAXBException

Extends: Throwable > Exception > JAXBException
```

**ValidationEventHandler**

`ValidationEventHandler.handleEvent(ValidationEvent)`

This exception indicates that an error has occurred while performing a validate operation.

The `ValidationEventHandler` can cause this exception to be thrown during the validate operations. See `ValidationEventHandler.handleEvent(ValidationEvent)`.

**Since:**

JAXB1.0

**Version:**

$Revision: 1.1 $
### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ValidationException(String message)</code></td>
<td>Construct an <code>ValidationException</code> with the specified detail message.</td>
</tr>
<tr>
<td><code>ValidationException(String message, String errorCode)</code></td>
<td>Construct an <code>ValidationException</code> with the specified detail message and vendor specific errorCode.</td>
</tr>
<tr>
<td><code>ValidationException(String message, String errorCode, Throwable exception)</code></td>
<td>Construct an <code>ValidationException</code> with the specified detail message, vendor specific errorCode, and linkedException.</td>
</tr>
<tr>
<td><code>ValidationException(String message, Throwable exception)</code></td>
<td>Construct an <code>ValidationException</code> with the specified detail message and linkedException.</td>
</tr>
<tr>
<td><code>ValidationException(Throwable exception)</code></td>
<td>Construct an <code>ValidationException</code> with a linkedException.</td>
</tr>
</tbody>
</table>

### Method Summary

#### Methods inherited from class `javax.xml.bind.JAXBException`
- `getCause`, `getErrorCode`, `getLinkedException`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setLinkedException`, `toString`

#### Methods inherited from class `java.lang.Throwable`
- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `setStackTrace`

#### Methods inherited from class `java.lang.Object`
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
Constructor Detail

```java
public ValidationException(String message)
```

ValidatesException errorCode linkedException null

```java
message
```

### ValidationException

```java
public ValidationException(String message)
```

Construct an ValidationException with the specified detail message. The errorCode and linkedException will default to null.

**Parameters:**

- `message` - a description of the exception

```java
public ValidationException(String message, String errorCode)
```

```java
errorCode ValidationException linkedException null
```

```java
message
```

```java
errorCode
```

### ValidationException

```java
public ValidationException(String message, String errorCode)
```

Construct an ValidationException with the specified detail message
and vendor specific errorCode. The linkedException will default to null.

**Parameters:**
- `message` - a description of the exception
- `errorCode` - a string specifying the vendor specific error code

---

```java
default constructor

public ValidationException(Throwable exception)
linkedException = ValidationException errorCode = null

exception
```

---

```java
ValidationException constructor

public ValidationException(Throwable exception)

Construct an ValidationException with a linkedException. The detail message and vendor specific errorCode will default to null.

**Parameters:**
- `exception` - the linked exception

---

```java
public ValidationException(String message, Throwable exception)
linkedException = ValidationException errorCode = null

message

exception
```

---

```java
ValidationException constructor

public ValidationException(String message, Throwable exception)

```
Construct an ValidationException with the specified detail message and linkedException. The errorCode will default to null.

**Parameters:**
- `message` - a description of the exception
- `exception` - the linked exception

```java
public ValidationException(String message, String errorCode, Throwable exception)
```

**ValidationException**

```java
public ValidationException(String message, String errorCode, Throwable exception)
```

Construct an ValidationException with the specified detail message, vendor specific errorCode, and linkedException.

**Parameters:**
- `message` - a description of the exception
- `errorCode` - a string specifying the vendor specific error code
- `exception` - the linked exception
PS:
javax.servlet.jsp.tagext  

Class **ValidationMessage**

java.lang.Object  
  ↓  javax.servlet.jsp.tagext.ValidationMessage

public class **ValidationMessage**
extends **Object**

TagLibraryValidator  TagExtraInfo

JSP 2.0 JSP  jsp:id  JSP  "id"  id  jsp:id
XML  XML  TagLibraryValidator  ValidationMessage

<table>
<thead>
<tr>
<th>id</th>
<th>jsp</th>
<th><a href="http://java.sun.com/JSP/Page">http://java.sun.com/JSP/Page</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>uri</td>
<td>TagLibraryValidator</td>
<td>id</td>
</tr>
</tbody>
</table>

A validation message from either TagLibraryValidator or TagExtraInfo.

As of JSP 2.0, a JSP container must support a jsp:id attribute to provide higher quality validation errors. The container will track the JSP pages as passed to the container, and will assign to each element a unique "id", which is passed as the value of the jsp:id attribute. Each XML element in the XML view available will be extended with this attribute. The TagLibraryValidator can then use the attribute in one or more ValidationMessage objects. The container then, in turn, can use these values to provide more precise information on the location of an error.

The actual prefix of the id attribute may or may not be jsp but it will always map to the namespace http://java.sun.com/JSP/Page. A TagLibraryValidator implementation must rely on the uri, not the prefix, of the id attribute.
**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValidationMessage(String id, String message)</td>
<td>Create a ValidationMessage.</td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String getId()</td>
<td>Get the jsp:id.</td>
</tr>
<tr>
<td>String getMessage()</td>
<td>Get the localized validation message.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

**Constructor Detail**

public ValidationMessage(String id, String message)  
ValidationMessage String null XML  
jsp:id id null id null id validate()  
PageData jsp:id  
    id null jsp:id  
message

**ValidationMessage**

public ValidationMessage(String id,  
String message)  
Create a ValidationMessage. The message String should be non-null. The value of id may be null, if the message is not specific to any XML element, or if no jsp:id attributes were passed on. If non-null, the value of id must be the value of a jsp:id attribute for the
PageData passed into the validate() method.

**Parameters:**
- id - Either null, or the value of a jsp:id attribute.
- message - A localized validation message.

### Method Detail

**public String getId()**
```java
jsp:idnull
    return jsp:id
```

**getId**

**public String getId()**
```
    Get the jsp:id. Null means that there is no information available.
```

**Returns:**
- The jsp:id information.

**public String getMessage()**
```java
return
```

**getMessage**

**public String getMessage()**
```
    Get the localized validation message.
```

**Returns:**
- A validation message
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.faces.validator Interface Validator

All Superinterfaces:
   EventListener

All Known Implementing Classes:
   DoubleRangeValidator, LengthValidator, LongRangeValidator, MethodExpressionValidator

public interface Validator
extends EventListener

Implements: java.util.EventListener
Implemented by: DoubleRangeValidator, LengthValidator, LongRangeValidator, MethodExpressionValidator

A Validator implementation is a class that can perform validation (correctness checks) on a EditableValueHolder. Zero or more Validators can be associated with each EditableValueHolder in the view, and are called during the Process Validations phase of the request processing lifecycle.

Individual Validators should examine the value and component that they are passed, and throw a ValidatorException containing a FacesMessage,
documenting any failures to conform to the required rules.

For maximum generality, Validator instances may be configurable based on properties of the Validator implementation class. For example, a range check Validator might support configuration of the minimum and maximum values to be used.

Validator implementations must have a zero-arguments public constructor. In addition, if the Validator class wishes to have configuration property values saved and restored with the view, the implementation must also implement StateHolder.

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String NOT_IN_RANGE_MESSAGE_ID</th>
<th>Deprecated. Use DoubleRangeValidator.NOT_IN_RANGE_MESSAGE_ID or LongRangeValidator.NOT_IN_RANGE_MESSAGE_ID instead.</th>
</tr>
</thead>
</table>

**Method Summary**

```java
void validate(FacesContext context, UIComponent component, Object value)
```

Perform the correctness checks implemented by this Validator against the specified UIComponent.

**Field Detail**

**NOT_IN_RANGE_MESSAGE_ID**

static final String NOT_IN_RANGE_MESSAGE_ID

**Deprecated. Use DoubleRangeValidator.NOT_IN_RANGE_MESSAGE_ID or LongRangeValidator.NOT_IN_RANGE_MESSAGE_ID instead.**
The message identifier of the FacesMessage to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set. The message format string for this message may optionally include a \{0\} placeholder, which will be replaced by the configured minimum value, and a \{1\} placeholder, which will be replaced by the configured maximum value.

See Also: Constant Field Values

Method Detail

```java
public void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException
```

- **UICOMPONENT**  
  **Validator**
  
  javax.faces.application.FacesMessage  
  ValidatorException

- **context**  
  FacesContext

- **component**  
  UIComponent

- **value**

**Throws**

- ValidatorException:

**Throws**

- NullPointerException: context component null

validate

```java
void validate(FacesContext context, UIComponent component, Object value) throws ValidatorException
```

Perform the correctness checks implemented by this Validator against the specified UIComponent. If any violations are found, a ValidatorException will be thrown containing the FacesMessage describing the failure.
Parameters:

- context - FacesContext for the request we are processing
- component - UIComponent we are checking for correctness
- value - the value to validate

Throws:

- ValidatorException - if validation fails
- NullPointerException - if context or component is null
javax.xml.bind Interface Validator

Deprecated. since JAXB 2.0

JAXB 2.0

Validator

XML  Java  Unmarshaller.setValidating
javadoc JAXB 1.0

Java  Java
JAXB 1.0

Java  Java  JAXB  JAXB
Validator  Unmarshaller  Marshaller
Unmarshaller /
Validator

Validator  Unmarshaller  Marshaller
Marshaller

validateunmarshal  marshal
Vali

setEventHandler  API  ValidatorUnmarshaller  Marshaller

Unmarshaller  /  Validator
As of JAXB 2.0, this class is deprecated and optional.

The `Validator` class is responsible for controlling the validation of content trees during runtime.

**Three Forms of Validation**

**Unmarshal-Time Validation**

This form of validation enables a client application to receive information about validation errors and warnings detected while unmarshalling XML data into a Java content tree and is completely orthogonal to the other types of validation. To enable or disable it, see the javadoc for `Unmarshaller.setValidating`. All JAXB 1.0 Providers are required to support this operation.
On-Demand Validation

This form of validation enables a client application to receive information about validation errors and warnings detected in the Java content tree. At any point, client applications can call the `Validator.validate` method on the Java content tree (or any sub-tree of it). All JAXB 1.0 Providers are required to support this operation.

Fail-Fast Validation

This form of validation enables a client application to receive immediate feedback about modifications to the Java content tree that violate type constraints on Java Properties as defined in the specification. JAXB Providers are not required support this type of validation. Of the JAXB Providers that do support this type of validation, some may require you to decide at schema compile time whether or not a client application will be allowed to request fail-fast validation at runtime.

The `Validator` class is responsible for managing On-Demand Validation. The `Unmarshaller` class is responsible for managing Unmarshal-Time Validation during the unmarshal operations. Although there is no formal method of enabling validation during the marshal operations, the `Marshaller` may detect errors, which will be reported to the `ValidationEventHandler` registered on it.

Using the Default EventHandler

If the client application does not set an event handler on their `Validator`, `Unmarshaller`, or `Marshaller` prior to calling the `validate`, `unmarshal`, or `marshal` methods, then a default event handler will receive notification of any errors or warnings encountered. The default event handler will cause the current operation to halt after encountering the first error or fatal error (but will attempt to continue after receiving warnings).

Handling Validation Events

There are three ways to handle events encountered during the unmarshal, validate, and marshal operations:
Use the default event handler
The default event handler will be used if you do not specify one via the setEventHandler API's on Validator, Unmarshaller, or Marshaller.

Implement and register a custom event handler
Client applications that require sophisticated event processing can implement the ValidationEventHandler interface and register it with the Unmarshaller and/or Validator.

Use the ValidationEventCollector utility
For convenience, a specialized event handler is provided that simply collects any ValidationEvent objects created during the unmarshal, validate, and marshal operations and returns them to the client application as a java.util.Collection.

Validation and Well-Formedness

Validation events are handled differently depending on how the client application is configured to process them as described in the previous section. However, there are certain cases where a JAXB Provider indicates that it is no longer able to reliably detect and report errors. In these cases, the JAXB Provider will set the severity of the ValidationEvent to FATAL_ERROR to indicate that the unmarshal, validate, or marshal operations should be terminated. The default event handler and ValidationEventCollector utility class must terminate processing after being notified of a fatal error. Client applications that supply their own ValidationEventHandler should also terminate processing after being notified of a fatal error. If not, unexpected behaviour may occur.

Supported Properties

There currently are not any properties required to be supported by all JAXB Providers on Validator. However, some providers may support their own set of provider specific properties.

Since:
JAXB1.0

Version:
$Revision: 1.4 $ $Date: 2005/07/29 20:56:02 $
Author:

- Ryan Shoemaker, Sun Microsystems, Inc.
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Joe Fialli, Sun Microsystems, Inc.

See Also:

JAXBContext, Unmarshaller, ValidationEventHandler, ValidationEvent, ValidationEventCollector

---

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getEventHandler()</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
<tr>
<td><code>getProperty(String name)</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
<tr>
<td><code>setEventHandler(ValidationEventHandler handler)</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
<tr>
<td><code>setProperty(String name, Object value)</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
<tr>
<td><code>validate(Object subrootObj)</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
<tr>
<td><code>validateRoot(Object rootObj)</code></td>
<td>Deprecated. since JAXB2.0</td>
</tr>
</tbody>
</table>

## Method Detail

```java
public void setEventHandler(ValidationEventHandler handler) throws JAXBException
```

```java
validate JAXB
null Validator
handler
```
setEvent.Handler

void setEvent.Handler(ValidationEventHandler handler)
throws JAXBException

Deprecated. since JAXB2.0

Allow an application to register a validation event handler.

The validation event handler will be called by the JAXB Provider if any validation errors are encountered during calls to validate. If the client application does not register a validation event handler before invoking the validate method, then validation events will be handled by the default event handler which will terminate the validate operation after the first error or fatal error is encountered.

Calling this method with a null parameter will cause the Validator to revert back to the default default event handler.

Parameters:
handler - the validation event handler

Throws:
JAXBException - if an error was encountered while setting the event handler

public ValidationEventHandler getEvent.Handler() throws JAXBException

return ValidationEventHandler

Throws JAXBException:

deprecated JAXB 2.0
getEventHandler

ValidationEventHandler getEventHandler() throws JAXBException

Deprecated. since JAXB2.0

Return the current event handler or the default event handler if one hasn't been set.

Returns:
the current ValidationEventHandler or the default event handler if it hasn't been set

Throws:
JAXBException - if an error was encountered while getting the current event handler

public boolean validate(Object subrootObj) throws JAXBException

subrootObj Java

Java subrootObj

Throws JAXBException:

Throws ValidationException:

Throws ValidationEventHandler
handleEvent false Validator subrootObj

Throws IllegalArgumentException:
return subrootObj null
return subrootObj true false
deprecated JAXB 2.0

validate

boolean validate(Object subrootObj) throws JAXBException
**Deprecated. since JAXB2.0**

Validate the Java content tree starting at `subrootObj`.

Client applications can use this method to validate Java content trees on-demand at runtime. This method can be used to validate any arbitrary subtree of the Java content tree. Global constraint checking **will not** be performed as part of this operation (i.e. ID/IDREF constraints).

**Parameters:**
- `subrootObj` - the obj to begin validation at

**Returns:**
- true if the subtree rooted at `subrootObj` is valid, false otherwise

**Throws:**
- `JAXBException` - if any unexpected problem occurs during validation
- `ValidationException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Validator` is unable to validate the content tree rooted at `subrootObj`
- `IllegalArgumentException` - If the `subrootObj` parameter is null

```
public boolean validateRoot(Object rootObj) throws 
JAXBException
  
rootObj   Java
```

```java
Java
  
rootObj
Throws  JAXBException:
  
Throws  ValidationException:  
  
Throws  IllegalArgumentException:  rootObj null
  
return  rootObj true false
deprecated  JAXB 2.0
```
**validateRoot**

```java
boolean validateRoot(Object rootObj)
    throws JAXBException
```

**Deprecated. since JAXB2.0**

Validate the Java content tree rooted at `rootObj`.

Client applications can use this method to validate Java content trees on-demand at runtime. This method is used to validate an entire Java content tree. Global constraint checking will be performed as part of this operation (i.e. ID/IDREF constraints).

**Parameters:**
- `rootObj` - the root obj to begin validation at

**Returns:**
- true if the tree rooted at `rootObj` is valid, false otherwise

**Throws:**
- `JAXBException` - if any unexpected problem occurs during validation
- `ValidationException` - If the `ValidationEventHandler` returns false from its `handleEvent` method or the `Validator` is unable to validate the content tree rooted at `rootObj`
- `IllegalArgumentException` - If the `rootObj` parameter is null

---

**public void setProperty(String name, Object value) throws PropertyException**

- **Validator**
  - JAXB
  - PropertyException

- **Throws**
  - PropertyException:
  - `IllegalArgumentException`: name null
  - `deprecated` JAXB 2.0

**setProperty**
void setProperty(String name, Object value) throws PropertyException

*Deprecated. since JAXB2.0*

Set the particular property in the underlying implementation of Validator. This method can only be used to set one of the standard JAXB defined properties above or a provider specific property. Attempting to set an undefined property will result in a PropertyException being thrown. See [Supported Properties](#).

**Parameters:**
- name - the name of the property to be set. This value can either be specified using one of the constant fields or a user supplied string.
- value - the value of the property to be set

**Throws:**
- PropertyException - when there is an error processing the given property or value
- IllegalArgumentException - If the name parameter is null

```java
public Object getProperty(String name) throws PropertyException

*Deprecated. since JAXB2.0*
```
Get the particular property in the underlying implementation of `Validator`. This method can only be used to get one of the standard JAXB defined properties above or a provider specific property. Attempting to get an undefined property will result in a `PropertyException` being thrown. See [Supported Properties](#).

**Parameters:**
- `name` - the name of the property to retrieve

**Returns:**
- the value of the requested property

**Throws:**
- `PropertyException` - when there is an error retrieving the given property or value property name
- `IllegalArgumentException` - If the name parameter is null

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.faces.webapp  Class ValidatorELTag

java.lang.Object
  ↓ java.servlet.jsp.tagext.TagSupport
    ↓ javax.faces.webapp.ValidatorELTag

All Implemented Interfaces:
  Serializable, IterationTag, JspTag, Tag

public abstract class ValidatorELTag
  extends TagSupport

Extends: TagSupport

ValidatorELTag JSP  UICOMPONENTTag
EditableValueHolder  Validator  UICOMPONENTTagBase
UICOMPONENT  Validator

Validator

cREATEVALIDATOR()  Validator  Validator
createVALIDATOR()

ValidatorELTag is a base class for all JSP custom actions that create and register a Validator instance on the EditableValueHolder associated with our most immediate surrounding instance of a tag whose implementation class is a subclass of UICOMPONENTTag. To avoid creating duplicate instances when a page is redisplayed, creation and registration of a Validator occurs only if the corresponding UICOMPONENT was created (by the owning UICOMPONENTTagBase) during the execution of the current page.
This class must be used as a base class for tag instances that support specific `Validator` subclasses.

Subclasses of this class must implement the `createValidator()` method, which creates and returns a `Validator` instance. Any configuration properties that specify the limits to be enforced by this `Validator` must have been set by the `createValidator()` method. Generally, this occurs by copying corresponding attribute values on the tag instance.

This tag creates no output to the page currently being created. It is used solely for the side effect of `Validator` creation.

See Also:
- Serialized Form

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class javax.servlet.jsp.tagext.<code>TagSupport</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>id, pageContext</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.servlet.jsp.tagext.<code>IterationTag</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>EVAL_BODY_AGAIN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fields inherited from interface javax.servlet.jsp.tagext.<code>Tag</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE</td>
</tr>
</tbody>
</table>

**Constructor Summary**

`ValidatorELTag()`

**Method Summary**

<table>
<thead>
<tr>
<th>protected abstract <code>createValidator()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Create and return a new <code>Validator</code> to be registered on</td>
</tr>
</tbody>
</table>
Validator

Create a new instance of the specified Validator class, and register it with the UIComponent instance associated with our most immediately surrounding UIComponentTagBase instance, if the UIComponent instance was created by this execution of the containing JSP page.

| Methods inherited from class javax.servlet.jsp.tagext.TagSupport |
|-------------------|-----------------|-------------------|
| doAfterBody, doEndTag, findAncestorWithClass, getId, getParent, getValue, getValues, release, removeValue, setId, setPageContext, setParent, setValue |

| Methods inherited from class java.lang.Object |
|-------------------|-----------------|-------------------|
| clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait |

**Constructor Detail**

public ValidatorELTag()

**Method Detail**

public int doStartTag() throws JspException

<table>
<thead>
<tr>
<th>Validator</th>
<th>JSP</th>
<th>UIComponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIComponentTagBase</td>
<td>UIComponent</td>
<td></td>
</tr>
</tbody>
</table>

Throws JspException: JSP
doStartTag

public int doStartTag() throws JspException

Create a new instance of the specified Validator class, and register it with the UICOMPONENT instance associated with our most immediately surrounding UICOMPONENT instance, if the UICOMPONENT instance was created by this execution of the containing JSP page.

Specified by: 
doStartTag in interface Tag
Overrides: 
doStartTag in class TagSupport
Returns: 
SKIP_BODY
Throws: 
JspException - if a JSP error occurs
See Also: 
Tag.doStartTag()

abstract protected Validator createValidator() throws JspException

Validator UICOMPONENT

Throws JspException:

createValidator

protected abstract Validator createValidator() throws JspException

Create and return a new Validator to be registered on our
surrounding UIComponent.

Throws:

```
JspException - if a new instance cannot be created
```

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.validator  

Class ValidatorException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ java.lang.RuntimeException
              └ javax.faces.FacesException
                  └ javax.faces.validator.ValidatorException

All Implemented Interfaces:
  Serializable

public class ValidatorException

extends FacesException

Extends: Throwable > Exception > RuntimeException > FacesException

ValidatorException  Validator  validate()

A ValidatorException is an exception thrown by the validate() method of a Validator to indicate that validation failed.

See Also:
  Serialized Form

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValidatorException(FacesMessage  message)</td>
<td>Construct a new exception with the specified message and no root cause.</td>
</tr>
<tr>
<td>ValidatorException(FacesMessage  message, Throwable  cause)</td>
<td>Construct a new exception with the specified detail message and root cause.</td>
</tr>
</tbody>
</table>
## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getFacesMessage()</code></td>
<td>Returns the FacesMessage associated with the exception.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.faces.FacesException`

- `getCause`

### Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

**public** `ValidatorException(FacesMessage message)`

*message*

## ValidatorException

**public** `ValidatorException(FacesMessage message)`

Construct a new exception with the specified message and no root cause.
public ValidatorException(FacesMessage message, Throwable cause)

Construct a new exception with the specified detail message and root cause.

Parameters:
message - The detail message for this exception
cause - The root cause for this exception

Method Detail

public FacesMessage getFacesMessage()

getFacesMessage

public FacesMessage getFacesMessage()

Returns the FacesMessage associated with the exception.
javax.faces.webapp  **Class ValidatorTag**

**java.lang.Object**
- javax.servlet.jsp.tagext.TagSupport
- javax.faces.webapp.ValidatorTag

**All Implemented Interfaces:**
- Serializable, IterationTag, JspTag, Tag

---

**Deprecated.** *This has been partially replaced by ValidatorELTag. The remainder of the functionality, namely, the binding facility and the implementation of the createValidator() method, is now an implementation detail.*

**Extends:** TagSupport

**ValidatorTag**  JSP  **UIComponentTag**
**EditableValueHolder**  Validator  **UIComponentTag**
**UIComponent**  **Validator**

<table>
<thead>
<tr>
<th>id</th>
<th>ID</th>
<th>Validator</th>
<th>id</th>
<th>ID</th>
<th>faces-config.xml</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

createValidator()  Validator  Validator
createValidator()

**deprecated**

ValidatorELTag  #createValidator

**public class**  **ValidatorTag**

extends TagSupport

**ValidatorTag** is a base class for all JSP custom actions that create and register a Validator instance on the EditableValueHolder associated with
our most immediate surrounding instance of a tag whose implementation class is a subclass of \texttt{UIComponentTag}. To avoid creating duplicate instances when a page is redisplayed, creation and registration of a \texttt{Validator} occurs \textbf{only} if the corresponding \texttt{UIComponent} was created (by the owning \texttt{UIComponentTag}) during the execution of the current page.

This class may be used directly to implement a generic validator registration tag (based on the validator-id specified by the \texttt{id} attribute), or as a base class for tag instances that support specific \texttt{Validator} subclasses. This \texttt{id} attribute must refer to one of the well known validator-ids, or a custom validator-id as defined in a \texttt{faces-config.xml} file.

Subclasses of this class must implement the \texttt{createValidator()} method, which creates and returns a \texttt{Validator} instance. Any configuration properties that specify the limits to be enforced by this \texttt{Validator} must have been set by the \texttt{createValidator()} method. Generally, this occurs by copying corresponding attribute values on the tag instance.

This tag creates no output to the page currently being created. It is used solely for the side effect of \texttt{Validator} creation.

\textbf{See Also:}

\texttt{Serialized Form}

\begin{center}
\textbf{Field Summary}
\end{center}

\begin{tabular}{|l|}
\hline
\textbf{Fields inherited from class javax.servlet.jsp.tagext.TagSupport} \\
\texttt{id, pageContext} \\
\hline
\textbf{Fields inherited from interface javax.servlet.jsp.tagext.IterationTag} \\
\texttt{EVAL_BODY_AGAIN} \\
\hline
\textbf{Fields inherited from interface javax.servlet.jsp.tagext.Tag} \\
\texttt{EVAL_BODY_INCLUDE, EVAL_PAGE, SKIP_BODY, SKIP_PAGE} \\
\hline
\end{tabular}
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ValidatorTag()</code></td>
<td>Deprecated.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createValidator()</code></td>
<td>Create and return a new <code>Validator</code> to be registered on our surrounding <code>UIComponent</code>. Deprecated.</td>
</tr>
<tr>
<td><code>doStartTag()</code></td>
<td>Create a new instance of the specified <code>Validator</code> class, and register it with the <code>UIComponent</code> instance associated with our most immediately surrounding <code>UIComponentTag</code> instance, if the <code>UIComponent</code> instance was created by this execution of the containing JSP page. Deprecated.</td>
</tr>
<tr>
<td><code>release()</code></td>
<td>Release references to any acquired resources. Deprecated.</td>
</tr>
<tr>
<td><code>setBinding(String binding)</code></td>
<td>Set the expression that will be used to create a <code>ValueExpression</code> that references a backing bean property of the <code>Validator</code> instance to be created. Deprecated.</td>
</tr>
<tr>
<td><code>setValidatorId(String validatorId)</code></td>
<td>Set the identifier of the <code>Validator</code> instance to be created. Deprecated.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `javax.servlet.jsp.tagext.TagSupport`

- `doAfterBody`, `doEndTag`, `findAncestorWithClass`, `getId`, `getParent`, `getValue`, `getValues`, `removeValue`, `setId`, `setPageContext`, `setParent`, `setValue`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`
public ValidatorTag()

ValidatorTag
public ValidatorTag()

Deprecated.

Method Detail
public void setValidatorId(String validatorId)

Validator

validatorId

setValidatorId
public void setValidatorId(String validatorId)

Deprecated.

Set the identifier of the Validator instance to be created.

Parameters:
validatorId - The new identifier of the validator instance to be created.

public void setBinding(String binding) throws JspException
**ValueExpression**  

**Validator** (backing) Bean

`binding`  

**Throws**  

**JspException**: JSP

---

**setBinding**

```java
class Validator {
    public void setBinding(String binding) throws JspException {
        // Deprecated.
        Set the expression that will be used to create a ValueExpression that references a backing bean property of the validator instance to be created.

        **Parameters:**
        binding - The new expression

        **Throws:**
        JspException - if a JSP error occurs
    }
}
```

---

**doStartTag()** throws **JspException**

```java
public int doStartTag() throws JspException {
    // Deprecated.
    // JSP UIComponent
    // UIComponentTag
    // Throws
    // JspException: JSP
}
```

---

**doStartTag**

```java
public int doStartTag() throws JspException {
    // Deprecated.
}
```
Create a new instance of the specified \texttt{Validator} class, and register it with the \texttt{UIComponent} instance associated with our most immediately surrounding \texttt{UIComponentTag} instance, if the \texttt{UIComponent} instance was created by this execution of the containing JSP page.

\textbf{Specified by:}
\begin{itemize}
  \item \texttt{doStartTag} in interface \texttt{Tag}
\end{itemize}

\textbf{Overrides:}
\begin{itemize}
  \item \texttt{doStartTag} in class \texttt{TagSupport}
\end{itemize}

\textbf{Returns:}
\begin{itemize}
  \item \texttt{SKIP\_BODY}
\end{itemize}

\textbf{Throws:}
\begin{itemize}
  \item \texttt{JspException} - if a JSP error occurs
\end{itemize}

\textbf{See Also:}
\begin{itemize}
  \item \texttt{Tag\_doStartTag()}
\end{itemize}

---

\textbf{public void release()}

\textbf{release}

\textbf{public void release()}

\textbf{Deprecated.}

Release references to any acquired resources.

\textbf{Specified by:}
\begin{itemize}
  \item \texttt{release} in interface \texttt{Tag}
\end{itemize}

\textbf{Overrides:}
\begin{itemize}
  \item \texttt{release} in class \texttt{TagSupport}
\end{itemize}

\textbf{See Also:}
\begin{itemize}
  \item \texttt{Tag\_release()}
\end{itemize}

---
protected `Validator` `createValidator()` throws `JspException`

`Validator` `UIComponent`

Throws `JspException`: `createValidator`

protected `Validator` `createValidator()`

throws `JspException`

Deprecated.

Create and return a new `validator` to be registered on our surrounding `UIComponent`.

Throws:

`JspException` - if a new instance cannot be created

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

Submit a bug or feature

PS:
**Class ValueBinding**

```
javax.faces.el
  Object
   javax.faces.el.ValueBinding
```

**Deprecated. This has been replaced by ValueExpression.**

The `ValueBinding` class is an object that can be used to access the property represented by an action or value binding expression. An immutable `ValueBinding` for a particular value binding can be acquired by calling the `createValueBinding()` method of the `Application` instance for this web application.

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>ValueBinding()</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>getExpressionString()</td>
</tr>
<tr>
<td>Material: String</td>
</tr>
<tr>
<td>Deprecated. Return the (possibly null) expression String, including the delimiters, from which this ValueBinding was built.</td>
</tr>
<tr>
<td>getType(FacesContext context)</td>
</tr>
<tr>
<td>Material: String</td>
</tr>
<tr>
<td>Returns the type represented by this ValueExpression</td>
</tr>
</tbody>
</table>
### Class

**Deprecated.** Return the type of the property represented by this `ValueBinding`, relative to the specified `FacesContext`.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>getValue</strong> (<code>FacesContext context</code>)</td>
<td>Deprecated. Return the value of the property represented by this <code>ValueBinding</code>, relative to the specified <code>FacesContext</code>.</td>
</tr>
<tr>
<td><strong>isReadOnly</strong> (<code>FacesContext context</code>)</td>
<td>Deprecated. Return <code>true</code> if the specified property of the specified property is known to be immutable; otherwise, return <code>false</code>.</td>
</tr>
<tr>
<td><strong>setValue</strong> (<code>FacesContext context, Object value</code>)</td>
<td>Deprecated. Set the value of the property represented by this <code>ValueBinding</code>, relative to the specified <code>FacesContext</code>.</td>
</tr>
</tbody>
</table>

### Constructor Detail

**public ValueBinding()**

**ValueBinding**

**public ValueBinding()**

*Deprecated.*

### Method Detail

**abstract public Object getValue** (`FacesContext context`)  
**throws** `EvaluationException`, `PropertyNotFoundException`
**FacesContext**  
**ValueBinding**

```
class ValueBinding
{
    public abstract Object getValue(FacesContext context)
        throws EvaluationException, PropertyNotFoundException;

    abstract public void setValue(FacesContext context, Object value)
        throws EvaluationException, PropertyNotFoundException;
}
```
Throws `EvaluationException`:
Throws `NullPointerException`: context null
Throws `PropertyNotFoundException`:

**setValue**

```java
public abstract void setValue(FacesContext context,
                              Object value)
throws EvaluationException,
       PropertyNotFoundException
```

*Deprecated.*

Set the value of the property represented by this `ValueBinding`, relative to the specified `FacesContext`.

**Parameters:**
- context - `FacesContext` for the current request
- value - The new value to be set

**Throws:**
- `EvaluationException` - if an exception is thrown while setting the value (the thrown exception must be included as the `cause` property of this exception)
- `NullPointerException` - if `context` is null
- `PropertyNotFoundException` - if a specified property name does not exist, or is not writeable

---

```java
abstract public boolean isReadOnly(FacesContext context)
throws EvaluationException, PropertyNotFoundException
```

true  false

```java
context FacesContext
Throws EvaluationException:
Throws NullPointerException: context null
Throws PropertyNotFoundException:
```
isReadOnly

```java
public abstract boolean isReadOnly(FacesContext context)
throws EvaluationException, PropertyNotFoundException
```

**Deprecated.**

Return `true` if the specified property of the specified property is known to be immutable; otherwise, return `false`.

**Parameters:**
- `context` - `FacesContext` for the current request

**Throws:**
- `EvaluationException` - if an exception is thrown while getting the description of the property (the thrown exception must be included as the `cause` property of this exception)
- `NullPointerException` - if `context` is null
- `PropertyNotFoundException` - if a specified property name does not exist

---

getType

```java
abstract public Class<T> getType(FacesContext context)
throws EvaluationException, PropertyNotFoundException
```

```java
context ValueBinding

<table>
<thead>
<tr>
<th>context</th>
<th>FacesContext</th>
</tr>
</thead>
<tbody>
<tr>
<td>Throws</td>
<td><code>EvaluationException</code>:</td>
</tr>
<tr>
<td>Throws</td>
<td><code>NullPointerException</code>: context null</td>
</tr>
<tr>
<td>Throws</td>
<td><code>PropertyNotFoundException</code>:</td>
</tr>
</tbody>
</table>
```

getType

```java
public abstract Class getType(FacesContext context)
throws EvaluationException,
```
PropertyNotFoundException

Deprecated.

Return the type of the property represented by this ValueBinding, relative to the specified FacesContext.

Parameters:

context - FacesContext for the current request

Throws:

EvaluationException - if an exception is thrown while getting the description of the property (the thrown exception must be included as the cause property of this exception)

NullPointerException - if context is null

PropertyNotFoundException - if a specified property name does not exist

public String getExpressionString()

    ValueBinding String null

getExpressionString

public String getExpressionString()

    Deprecated.

    Return the (possibly null) expression String, including the delimiters, from which this ValueBinding was built.
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.faces.event Class ValueChangeEvent

java.lang.Object
     | java.util.EventObject
     |    | javax.faces.event.FacesEvent
     |    | javax.faces.event.ValueChangeEvent

All Implemented Interfaces:
    Serializable

public class ValueChangeEvent
  extends FacesEvent

Extends: java.util.EventObject > FacesEvent

ValueChangeEvent

A ValueChangeEvent is a notification that the local value of the source component has been change as a result of user interface activity. It is not fired unless validation of the new value was completed successfully.

See Also:
    Serialized Form

---

Field Summary

Fields inherited from class java.util.EventObject
source

Constructor Summary

ValueChangeEvent(UIComponent component, Object oldValue,
Construct a new event object from the specified source component, old value, and new value.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td>Return the current local value of the source <strong>UIComponent</strong>.</td>
</tr>
<tr>
<td><strong>Object</strong></td>
</tr>
<tr>
<td>Return the previous local value of the source <strong>UIComponent</strong>.</td>
</tr>
<tr>
<td><strong>boolean</strong></td>
</tr>
<tr>
<td>Return true if this <strong>FacesListener</strong> is an instance of a listener class that this event supports.</td>
</tr>
<tr>
<td><strong>void</strong></td>
</tr>
<tr>
<td>Broadcast this <strong>FacesEvent</strong> to the specified <strong>FacesListener</strong>, by whatever mechanism is appropriate.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.faces.event.**FacesEvent**

getComponent, getPhaseId, queue, setPhaseId

Methods inherited from class java.util.**EventObject**

g Source, toString

Methods inherited from class java.lang.**Object**

clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

**Constructor Detail**

public ValueChangeEvent(**UIComponent** component, Object oldValue, Object newValue)
ValueChangeEvent

public ValueChangeEvent(UIComponent component, Object oldValue, Object newValue)

Construct a new event object from the specified source component, old value, and new value.

The default PhaseId for this event is PhaseId.ANY_PHASE.

Parameters:
- component - Source UIComponent for this event
- oldValue - The previous local value of this UIComponent
- newValue - The new local value of this UIComponent

Throws:
- IllegalArgumentException - if component is null

Method Detail

public Object getOldValue()

UIComponent

ggetOldValue

public Object getOldValue()
public Object getNewValue()

    UICOMPONENT

getNewValue

public Object getNewValue()

    Return the current local value of the source UICOMPONENT.

public boolean isAppropriateListener(FacesListener listener)

isAppropriateListener

public boolean isAppropriateListener(FacesListener listener)

Description copied from class: FacesEvent

Return true if this FacesListener is an instance of a listener class that this event supports. Typically, this will be accomplished by an "instanceof" check on the listener class.

Specified by:
    isAppropriateListener in class FacesEvent

Parameters:
    listener - FacesListener to evaluate

public void processListener(FacesListener listener)

    AbortProcessingException: NullPointerException
    JavaServer
public void processListener(FacesListener listener)

Description copied from class: FacesEvent

Broadcast this FacesEvent to the specified FacesListener, by whatever mechanism is appropriate. Typically, this will be accomplished by calling an event processing method, and passing this FacesEvent as a parameter.

Specified by:

processListener in class FacesEvent

Parameters:

listener - FacesListener to send this FacesEvent to

Throws:

AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.event Interface ValueChangeListener

All Superinterfaces: 
EventListener, FacesListener

All Known Implementing Classes: 
MethodExpressionValueChangeListener

public interface ValueChangeListener 
extends FacesListener

Implements: FacesListener
Implemented by: MethodExpressionValueChangeListener

ValueChangeEvent addValueChangeListener()
UIComponent

A listener interface for receiving ValueChangeEvent s. A class that is interested in receiving such events implements this interface, and then registers itself with the source UIComponent of interest, by calling addValueChangeListener().

Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void processValueChange(ValueChangeEvent event)</td>
<td>Invoked when the value change described by the specified ValueChangeEvent occurs.</td>
</tr>
</tbody>
</table>

Method Detail
public void processValueChange(ValueChangeEvent event) throws AbortProcessingException

Throws
AbortProcessingException: JavaServer Faces

Invoked when the value change described by the specified ValueChangeEvent occurs.

Parameters:
  event - The ValueChangeEvent that has occurred

Throws:
AbortProcessingException - Signal the JavaServer Faces implementation that no further processing on the current event should be performed

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.el Class ValueExpression

java.lang.Object
   └─javax.el.Expression
      └─javax.el.ValueExpression

All Implemented Interfaces:
   Serializable

public abstract class ValueExpression
   extends Expression

Extends: Expression

API     ValueExpression  l-value  r-value  r-value  l-value
EL      l-value          "$\{1+1\}$"   "$\{firstName\} \$\{lastName\}"

ExpressionFactory#createValueExpression       ValueExpression
   FunctionMapper
   ELContext

   #getValue#setValue#isReadOnly #getType   ELContext
ELResolver . [ ] 4       ELResolver#getValue
base     ValueExpression
ELResolver#getValueELResolver#setValueELResolver#isReadOnly
ELResolver#getType     ValueExpression

Expression javadoc
since JSP 2.1
See javax.el.ELResolver, javax.el.Expression,
also javax.el.ExpressionFactory

An Expression that can get or set a value.
In previous incarnations of this API, expressions could only be read. ValueExpression objects can now be used both to retrieve a value and to set a value. Expressions that can have a value set on them are referred to as l-value expressions. Those that cannot are referred to as r-value expressions. Not all r-value expressions can be used as l-value expressions (e.g. "${1+1}" or "${firstName} ${lastName}"). See the EL Specification for details. Expressions that cannot be used as l-values must always return true from isReadOnly().

The ExpressionFactory.createValueExpression(javax.el.ELContext, java.lang.String, java.lang.Class) method can be used to parse an expression string and return a concrete instance of ValueExpression that encapsulates the parsed expression. The FunctionMapper is used at parse time, not evaluation time, so one is not needed to evaluate an expression using this class. However, the ELContext is needed at evaluation time.

The getValue(javax.el.ELContext), setValue(javax.el.ELContext, java.lang.Object), isReadOnly(javax.el.ELContext) and getType(javax.el.ELContext) methods will evaluate the expression each time they are called. The ELResolver in the ELContext is used to resolve the top-level variables and to determine the behavior of the . and [] operators. For any of the four methods, the ELResolver.getValue(javax.el.ELContext, java.lang.Object, java.lang.Object) method is used to resolve all properties up to but excluding the last one. This provides the base object. At the last resolution, the ValueExpression will call the corresponding ELResolver.getValue(javax.el.ELContext, java.lang.Object, java.lang.Object), ELResolver.setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object), ELResolver.isReadOnly(javax.el.ELContext, java.lang.Object, java.lang.Object) or ELResolver.getType(javax.el.ELContext, java.lang.Object, java.lang.Object) method, depending on which was called on the ValueExpression.

See the notes about comparison, serialization and immutability in the Expression javadocs.

Since:
    JSP 2.1
See Also:

ELResolver, Expression, ExpressionFactory, Serialized Form

## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>ValueExpression()</th>
</tr>
</thead>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract Class&lt;? getExpectedType()</td>
<td>Returns the type the result of the expression will be coerced to after evaluation.</td>
</tr>
<tr>
<td>abstract Class&lt;? getType(ELContext context)</td>
<td>Evaluates the expression relative to the provided context, and returns the most general type that is acceptable for an object to be passed as the value parameter in a future call to the setValue(javax.el.ELContext, java.lang.Object) method.</td>
</tr>
<tr>
<td>abstract Object getValue(ELContext context)</td>
<td>Evaluates the expression relative to the provided context, and returns the resulting value.</td>
</tr>
<tr>
<td>abstract boolean isReadOnly(ELContext context)</td>
<td>Evaluates the expression relative to the provided context, and returns true if a call to setValue(javax.el.ELContext, java.lang.Object) will always fail.</td>
</tr>
<tr>
<td>abstract void setValue(ELContext context, Object value)</td>
<td>Evaluates the expression relative to the provided context, and sets the result to the provided value.</td>
</tr>
</tbody>
</table>

Methods inherited from class javax.el.Expression

equals, getExpressionString, hashCode, isLiteralText

Methods inherited from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, toString, wait, wait
### Constructor Detail

```java
class ValueExpression {
    public ValueExpression() {
    }
}
```

### Method Detail

#### abstract public Object getValue(ELContext context)

```java
public abstract Object getValue(ELContext context)
```

- Evaluates the expression relative to the provided context, and returns the resulting value.
- The resulting value is automatically coerced to the type returned by `getExpectedType()`, which was provided to the `ExpressionFactory` when this expression was created.

```java
ExpressionFactory getExpectedType()
```

- `context`:
- `return`:
- `Throws`:
  - `NullPointerException`: context null
  - `PropertyNotFoundException`: 
  - `ELException`: cause
Parameters:
   context - The context of this evaluation.

Returns:
   The result of the expression evaluation.

 Throws:
   NullPointerException - if context is null.
   PropertyNotFoundException - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.
   ELException - if an exception was thrown while performing property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

abstract public void setValue(ELContext context, Object value)

   context
   value

   Throws
   NullPointerException: context null
   PropertyNotFoundException:
   PropertyNotWritableException:
   ELException: cause

setValue

public abstract void setValue(ELContext context, Object value)

   Evaluates the expression relative to the provided context, and sets the result to the provided value.

Parameters:
   context - The context of this evaluation.
   value - The new value to be set.

 Throws:
   NullPointerException - if context is null.
**PropertyNotFoundException** - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.

**PropertyNotWritableException** - if the final variable or property resolution failed because the specified variable or property is not writable.

**ELException** - if an exception was thrown while attempting to set the property or variable. The thrown exception must be included as the cause property of this exception, if available.

---

abstract public boolean **isReadOnly**(ELContext context)  

    #setValue    true

    context

    return true false

    Throws NullPointerException: context null

    Throws PropertyNotFoundException:

    Throws ELException: cause

        NullPointerException context null

**isReadOnly**

public abstract boolean **isReadOnly**(ELContext context)

    Evaluates the expression relative to the provided context, and returns true if a call to setValue(javax.el.ELContext, java.lang.Object) will always fail.

**Parameters:**

    context - The context of this evaluation.

**Returns:**

    true if the expression is read-only or false if not.

**Throws:**

    NullPointerException - if context is null.

    PropertyNotFoundException - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.
**ELException** - if an exception was thrown while performing property or variable resolution. The thrown exception must be included as the cause property of this exception, if available. * @throws NullPointerException if context is null

```java
abstract public Class<T> getType(ELContext context)
```

```java
getValue().getClass() getType
```

**Throws**
- NullPointerException: context null
- PropertyNotFoundException:
- ELException: cause

```
public abstract <T> Class<T> getType(ELContext context)
```

Evaluates the expression relative to the provided context, and returns the most general type that is acceptable for an object to be passed as the value parameter in a future call to the setValue(javax.el.ELContext, java.lang.Object) method.

This is not always the same as getValue().getClass(). For example, in the case of an expression that references an array element, the getType method will return the element type of the array, which might be a superclass of the type of the actual element that is currently in the specified array element.

**Parameters:**
- context: The context of this evaluation.

**Returns:**
- the most general acceptable type; otherwise undefined.

**Throws:**
NullPointerException - if context is null.
PropertyNotFoundException - if one of the property resolutions failed because a specified variable or property does not exist or is not readable.
ELException - if an exception was thrown while performing property or variable resolution. The thrown exception must be included as the cause property of this exception, if available.

abstract public Class<T> getExpectedType()

    return ValueExpression ExpressionFactory.createValueExpression(expectedType)

getExpectedType

public abstract Class<?<> getExpectedType()

    Returns the type the result of the expression will be coerced to after evaluation.

    Returns:
    the expectedType passed to the
    ExpressionFactory.createValueExpression method that created this ValueExpression.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.faces.component Interface ValueHolder

All Known Subinterfaces:
   EditableValueHolder

All Known Implementing Classes:

public interface ValueHolder

Implemented by: EditableValueHolder, UOutput

ValueHolder String UComponent

ValueHolder is an interface that may be implemented by any concrete UComponent that wishes to support a local value, as well as access data in the model tier via a value expression, and support conversion between String and the model tier data’s native data type.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getConverter()</td>
<td>Return the Converter (if any) that is registered for this UComponent.</td>
</tr>
<tr>
<td>getLocalValue()</td>
<td>Return the local value of this UComponent (if any), without evaluating any associated ValueExpression.</td>
</tr>
</tbody>
</table>
### Method Detail

#### public Object getLocalValue()

This method returns the local value of this UIComponent (if any), without evaluating any associated ValueExpression.

#### public Object getValue()

This method gets the value of this UIComponent.
Gets the value of this `UIComponent`. First, consult the local value property of this component. If non-null return it. If null, see if we have a `ValueExpression` for the `value` property. If so, return the result of evaluating the property, otherwise return null. Note that because the specification for `UIComponent.setValueBinding(java.lang.String, javax.faces.el.ValueBinding)` requires a call through to `UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression)`, legacy tags will continue to work.

```
public void setValue(Object value)

   UIComponent

   value

setValue

void setValue(Object value)

   Set the value of this `UIComponent` (if any).

   Parameters:
      value - The new local value
```

```
public Converter getConverter()

   UIComponent   Converter

getConverter

   Converter getConverter()

   Return the `Converter` (if any) that is registered for this `UIComponent`.
```
public void setConverter(Converter converter)

setConverter

void setConverter(Converter converter)

Set the Converter (if any) that is registered for this UIComponent.

Parameters:

converter - New Converter (or null)
javax.servlet.jsp.tagext  **Class VariableInfo**

java.lang.Object  

javax.servlet.jsp.tagext.VariableInfo

public class **VariableInfo**  

extends **Object**

/ TagExtraInfo  JSP

   AT_BEGINNESTED  AT_END

   (VariableInfo.getClassName) ""  

java.lang.Integer

Web  CLASSPATH  Servlet 2.4  -  WEB-INF/lib  
WEB-INF/classes

VariableInfo import  Web  CLASSPATH  
Servlet 2.4  -  WEB-INF/lib  WEB-INF/classes

JAR  JSP  J2EE

JSP import

getVariableInfo  VariableInfo  VariableInfo

JSP 2.0
- NESTED
- AT_BEGIN
- AT_END

<table>
<thead>
<tr>
<th></th>
<th>doStartTag()</th>
<th>doInitBody()</th>
<th>doAfterBody()</th>
<th>doEndTag()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag</td>
<td>AT_BEGIN</td>
<td>NESTED</td>
<td></td>
<td>AT_BEGIN</td>
</tr>
<tr>
<td>IterationTag</td>
<td>AT_BEGIN</td>
<td>NESTED</td>
<td>AT_BEGIN</td>
<td>AT_END</td>
</tr>
<tr>
<td>BodyTag</td>
<td>AT_BEGIN</td>
<td>NESTED¹</td>
<td>AT_BEGIN</td>
<td>AT_END</td>
</tr>
<tr>
<td>SimpleTag</td>
<td></td>
<td></td>
<td></td>
<td>AT_END</td>
</tr>
</tbody>
</table>

¹ EVAL_BODY_INCLUDE  

doStartTag()  
doInitBody()

TLD

<tag>  <variable>  JSP

Information on the scripting variables that are created/modified by a tag (at run-time). This information is provided by TagExtraInfo classes and it is used by the translation phase of JSP.

Scripting variables generated by a custom action have an associated scope of either AT_BEGIN, NESTED, or AT_END.

The class name (VariableInfo.getClassName) in the returned objects is used to determine the types of the scripting variables. Note that because scripting variables are assigned their values from scoped attributes which cannot be of primitive types, "boxed" types such as java.lang.Integer must be used instead of primitives.

The class name may be a Fully Qualified Class Name, or a short class
name.

If a Fully Qualified Class Name is provided, it should refer to a class that should be in the CLASSPATH for the Web Application (see Servlet 2.4 specification - essentially it is WEB-INF/lib and WEB-INF/classes). Failure to be so will lead to a translation-time error.

If a short class name is given in the VariableInfo objects, then the class name must be that of a public class in the context of the import directives of the page where the custom action appears. The class must also be in the CLASSPATH for the Web Application (see Servlet 2.4 specification - essentially it is WEB-INF/lib and WEB-INF/classes). Failure to be so will lead to a translation-time error.

**Usage Comments**

Frequently a fully qualified class name will refer to a class that is known to the tag library and thus, delivered in the same JAR file as the tag handlers. In most other remaining cases it will refer to a class that is in the platform on which the JSP processor is built (like Java EE). Using fully qualified class names in this manner makes the usage relatively resistant to configuration errors.

A short name is usually generated by the tag library based on some attributes passed through from the custom action user (the author), and it is thus less robust: for instance a missing import directive in the referring JSP page will lead to an invalid short name class and a translation error.

**Synchronization Protocol**

The result of the invocation on getVariableInfo is an array of VariableInfo objects. Each such object describes a scripting variable by providing its name, its type, whether the variable is new or not, and what its scope is. Scope is best described through a picture:

The JSP 2.0 specification defines the interpretation of 3 values:

- NESTED, if the scripting variable is available between the start tag
and the end tag of the action that defines it.

- **AT_BEGIN**, if the scripting variable is available from the start tag of
  the action that defines it until the end of the scope.
- **AT_END**, if the scripting variable is available after the end tag of the
  action that defines it until the end of the scope.

The scope value for a variable implies what methods may affect its value and
thus where synchronization is needed as illustrated by the table below. **Note:** the synchronization of the variable(s) will occur *after* the respective method has been called.

<table>
<thead>
<tr>
<th>Variable Synchronization Points</th>
<th>doStartTag()</th>
<th>doInitBody()</th>
<th>doAfterBody()</th>
<th>doEndTag()</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tag</strong></td>
<td>AT_BEGIN, NESTED</td>
<td></td>
<td>AT_BEGIN, AT_END</td>
<td></td>
</tr>
<tr>
<td><strong>IterationTag</strong></td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, AT_END</td>
<td></td>
</tr>
<tr>
<td><strong>BodyTag</strong></td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, AT_END</td>
</tr>
<tr>
<td><strong>SimpleTag</strong></td>
<td>AT_BEGIN, NESTED</td>
<td>AT_BEGIN, NESTED</td>
<td></td>
<td>AT_BEGIN, AT_END</td>
</tr>
</tbody>
</table>

1 Called after doStartTag() if EVAL_BODY_INCLUDE is returned, or after doInitBody() otherwise.

**Variable Information in the TLD**

Scripting variable information can also be encoded directly for most cases into the Tag Library Descriptor using the `<variable>` subelement of the `<tag>` element. See the JSP specification.

---

**Field Summary**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static int AT_BEGIN</strong></td>
<td>Scope information that scripting variable is visible after start tag.</td>
</tr>
<tr>
<td><strong>int AT_END</strong></td>
<td></td>
</tr>
</tbody>
</table>
**Constructor Summary**

```java
VariableInfo(String varName, String className, boolean declare, int scope)
```

Constructor

These objects can be created (at translation time) by the TagExtraInfo instances.

**Method Summary**

- `String getClassName()`
  - Returns the type of this variable.

- `boolean getDeclare()`
  - Returns whether this is a new variable.

- `int getScope()`
  - Returns the lexical scope of the variable.

- `String getVarName()`
  - Returns the name of the scripting variable.

**Methods inherited from class java.lang.Object**

- `clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

**Field Detail**

**NESTED**

```java
public static final int NESTED
```
Scope information that scripting variable is visible only within the start/end tags.

See Also:
Constant Field Values

AT_BEGIN

public static final int AT_BEGIN

Scope information that scripting variable is visible after start tag.

See Also:
Constant Field Values

AT_END

public static final int AT_END

Scope information that scripting variable is visible after end tag.

See Also:
Constant Field Values

Constructor Detail

public VariableInfo(String varName, String className, boolean declare, int scope)

TagExtraInfo

varName
className
VariableInfo

public VariableInfo(String varName, String className, boolean declare, int scope)

Constructor These objects can be created (at translation time) by the TagExtraInfo instances.

Parameters:
- varName - The name of the scripting variable
- className - The type of this variable
- declare - If true, it is a new variable (in some languages this will require a declaration)
- scope - Indication on the lexical scope of the variable

Method Detail

public String getVarName()

    return

getVarName

public String getVarName()

    Returns the name of the scripting variable.

    Returns:
    the name of the scripting variable
public String getClassName()

    return

getClassName

public String getClassName()

    Returns the type of this variable.

    Returns:
    the type of this variable

public boolean getDeclare()

    return

getDeclare

public boolean getDeclare()

    Returns whether this is a new variable. If so, in some languages this will require a declaration.

    Returns:
    whether this is a new variable.

public int getScope()

    return

    See also

    AT_BEGIN, AT_END, NESTED
getScope

public int getScope()

Returns the lexical scope of the variable.

Returns: the lexical scope of the variable, either AT_BEGIN, AT_END, or NESTED.

See Also: AT_BEGIN, AT_END, NESTED
javax.el  **Class VariableMapper**

`java.lang.Object`  
└ `javax.el.VariableMapper`  

```java
public abstract class VariableMapper
    extends Object
```

**EL**  

```
since JSP 2.1
```

The interface to a map between EL variables and the EL expressions they are associated with.

**Since:**  
JSP 2.1

**Constructor Summary**

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>VariableMapper()</code></td>
</tr>
</tbody>
</table>

**Method Summary**

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>abstract ValueExpression resolveVariable(String variable)</code></td>
</tr>
</tbody>
</table>
| `abstract ValueExpression setVariable(String variable, ValueExpression expression)`  
Assign a ValueExpression to an EL variable, replacing any previously assignment to the same variable. |

**Methods inherited from class java.lang.Object**
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

## Constructor Detail

public VariableMapper()

VariableMapper

public VariableMapper()

## Method Detail

abstract public ValueExpression resolveVariable(String variable)

    variable
    return ValueExpression null

resolveVariable

public abstract ValueExpression resolveVariable(String variable)

    Parameters:
    variable - The variable name

    Returns:
    the ValueExpression assigned to the variable, null if there is no previous assignment to this variable.

abstract public ValueExpression setVariable(String variable, ValueExpression expression)

ValueExpression EL
setVariable

public abstract ValueExpression setVariable(String variable, ValueExpression expression)

Assign a ValueExpression to an EL variable, replacing any previously assignment to the same variable. The assignment for the variable is removed if the expression is null.

Parameters:
- variable - The variable name
- expression - The ValueExpression to be assigned to the variable.

Returns:
The previous ValueExpression assigned to this variable, null if there is no previous assignment to this variable.
javax.faces.el  Class VariableResolver

java.lang.Object  
  javax.faces.el.VariableResolver

Deprecated. This has been replaced by ELResolver when operating with a null base argument.

VariableResolver

deprecated  null base  javax.el.ELResolver

public abstract class VariableResolver
extends Object

VariableResolver represents a pluggable mechanism for resolving a top-level variable reference at evaluation time.

Constructor Summary

<table>
<thead>
<tr>
<th>ConstructorResolver()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated.</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>abstract Object resolveVariable(FacesContext context, String name)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated. Resolve the specified variable name, and return the corresponding object, if any; otherwise, return null.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait
Constructor Detail

public VariableResolver()

VariableResolver

public VariableResolver()

Deprecated.

Method Detail

abstract public Object resolveVariable(FacesContext context, String name) throws EvaluationException

null

context
name

Throws EvaluationException:

Throws NullPointerException: context name null

resolveVariable

public abstract Object resolveVariable(FacesContext context, String name)

throws EvaluationException

Deprecated.

Resolve the specified variable name, and return the corresponding object, if any; otherwise, return null.
Parameters:
  context - FacesContext against which to resolve this variable name
  name - Name of the variable to be resolved

Throws:
  EvaluationException - if an exception is thrown while resolving the variable name (the thrown exception must be included as the cause property of this exception)
  NullPointerException - if context or name is null

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Interface VariableResolver

Deprecated. As of JSP 2.1, replaced by ELResolver

ExpressionEvaluator ""

since 2.0

public interface VariableResolver

This class is used to customize the way an ExpressionEvaluator resolves variable references at evaluation time. For example, instances of this class can implement their own variable lookup mechanisms, or introduce the notion of "implicit variables" which override any other variables. An instance of this class should be passed when evaluating an expression.

An instance of this class includes the context against which resolution will happen.

Since:
JSP 2.0

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolveVariable(String pName)</td>
</tr>
<tr>
<td>Deprecated. Resolves the specified variable.</td>
</tr>
</tbody>
</table>

| Method Detail |
public Object resolveVariable(String pName) throws ELException
null

  pName
  return
  Throws

resolvéVariable

Object resolveVariable(String pName)

throws ELException

Deprecated.
Resolves the specified variable. Returns null if the variable is not found.

Parameters:
  pName - the name of the variable to resolve

Returns:
  the result of the variable resolution

Throws:
  ELException - if a failure occurred while trying to resolve the given variable
This annotation specifies the version field or property of an entity class that serves as its optimistic lock value. The version is used to ensure integrity when performing the merge operation and for optimistic concurrency control.

Only a single `Version` property or field should be used per class; applications that use more than one `Version` property or field will not be portable.

The `Version` property should be mapped to the primary table for the entity
class; applications that map the Version property to a table other than the primary table will not be portable.

The following types are supported for version properties: int, Integer, short, Short, long, Long, Timestamp.

Example:

```java
@Version
@Column(name="OPTLOCK")
protected int getVersionNum() { return versionNum; }
```

Since:
Java Persistence 1.0

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.registry.infomodel Interface Versionable

All Known Subinterfaces:
   ClassificationScheme, ExtrinsicObject, RegistryEntry,
   RegistryPackage, Service

public interface Versionable

Implemented by: RegistryEntry

Versionable RegistryEntry Versionable

See also javax.xml.registry.infomodel.RegistryEntry

The Versionable interface defines the behavior common to classes that are capable of creating versions of their instances. At present all RegistryEntry classes are required to implement the Versionable interface.

Author:
   Farrukh S. Najmi

See Also:
   RegistryEntry

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>int getMajorVersion()</td>
<td>Gets the major revision number for this version of the Versionable object.</td>
</tr>
<tr>
<td>int getMinorVersion()</td>
<td>Gets the minor revision number for this version of the Versionable object.</td>
</tr>
<tr>
<td>String getUserVersion()</td>
<td>Gets the user-specified revision number for this version of the Versionable object.</td>
</tr>
</tbody>
</table>
**Method Detail**

```java
public int getMajorVersion() throws JAXRException

Versionable

```

1

```java
return

Throws JAXRException: JAXR

**getMajorVersion**

```java
int getMajorVersion() throws JAXRException

```

Gets the major revision number for this version of the Versionable object.

**Capability Level: 1**

**Returns:**

the major version for this object

**Throws:**

JAXRException - If the JAXR provider encounters an internal error
public void setMajorVersion(int majorVersion) throws JAXRException
Versionable

1

majorVersion

Throws JAXRException: JAXR

setMajorVersion

void setMajorVersion(int majorVersion)
throws JAXRException

Sets the major revision number for this version of the Versionable object.

Capability Level: 1

Parameters:
   majorVersion - the major version number

Throws:
   JAXRException - If the JAXR provider encounters an internal error

public int getMinorVersion() throws JAXRException
Versionable

1

return

Throws JAXRException: JAXR
getMinorVersion

```java
int getMinorVersion() throws JAXRException
```

Gets the minor revision number for this version of the Versionable object.

**Capability Level: 1**

**Returns:**
- the minor version for this object

**Throws:**
- `JAXRException` - If the JAXR provider encounters an internal error

---

**public void setMinorVersion(int minorVersion) throws JAXRException**

**Versionable**

1

```java
minorVersion
```

**Throws**
- `JAXRException`: JAXR

---

setMinorVersion

```java
void setMinorVersion(int minorVersion) throws JAXRException
```

Sets the minor revision number for this version of the Versionable object.

**Capability Level: 1**

**Parameters:**
- `minorVersion` - the minor version number
Throws:
JAXRException - If the JAXR provider encounters an internal error

public String getUserVersion() throws JAXRException
Versionable

1

return
Throws JAXRException: JAXR

getUserVersion

String getUserVersion() throws JAXRException

Gets the user-specified revision number for this version of the Versionable object.

Capability Level: 1

Returns: the user-defined version number

Throws: JAXRException - If the JAXR provider encounters an internal error

public void setUserVersion(String userVersion) throws JAXRException
Versionable

1

userVersion
Throws: **JAXRException**: JAXR

### setUserVersion

```java
void setUserVersion(String userVersion)
throws JAXRException
```

Sets the user specified revision number for this version of the Versionable object.

**Capability Level: 1**

**Parameters:**
- `userVersion` - the user-defined version number

**Throws:**
- **JAXRException** - If the JAXR provider encounters an internal error

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.faces.application   Class ViewExpiredException

java.lang.Object  
   java.lang.Throwable  
      java.lang.Exception  
         java.lang.RuntimeException  
            javax.faces.FacesException  
               javax.faces.application.ViewExpiredException

All Implemented Interfaces:
   Serializable

public class ViewExpiredException  
   extends FacesException

Extends: Throwable > Exception > RuntimeException > FacesException

restoreView(javax.faces.context.FacesContext, String, String)  
   FacesException

since          1.2

Implementations must throw this FacesException when attempting to restore the view
StateManager.restoreView(javax.faces.context.FacesContext, String, String) results in failure on postback.

Since:  
   1.2

See Also:  
   Serialized Form

---

Constructor Summary
**ViewExpiredException()**  
Construct a new exception with no detail message or root cause.

**ViewExpiredException(String viewId)**  
Construct a new exception with the specified view identifier.

**ViewExpiredException(String message, String viewId)**  
Construct a new exception with the specified detail message and no root cause.

**ViewExpiredException(String message, Throwable cause, String viewId)**  
Construct a new exception with the specified detail message and root cause.

**ViewExpiredException(Throwable cause, String viewId)**  
Construct a new exception with the specified root cause.

---

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>String getMessage()</code></td>
<td>Return the message for this exception prepended with the view identifier if the view identifier is not <code>null</code>, otherwise, return the message.</td>
</tr>
<tr>
<td><code>String getViewId()</code></td>
<td>Return the view identifier of this exception, or <code>null</code> if the view identifier is nonexistent or unknown.</td>
</tr>
</tbody>
</table>

---

### Methods inherited from class `javax.faces.FacesException`

- `getCause`

### Methods inherited from class `java.lang.Throwable`

- `fillInStackTrace`, `getLocalizedMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

### Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
public ViewExpiredException()

ViewExpiredException

public ViewExpiredException()

Construct a new exception with no detail message or root cause.

public ViewExpiredException(String viewId)

viewId

ViewExpiredException

public ViewExpiredException(String viewId)

Construct a new exception with the specified view identifier.

Parameters:

    viewId - The view identifier for this exception

public ViewExpiredException(String message, String viewId)
ViewExpiredException

public ViewExpiredException(String message, String viewId)

Construct a new exception with the specified detail message and no root cause.

Parameters:
message - The detail message for this exception
viewId - The view identifier for this exception

public ViewExpiredException(Throwable cause, String viewId)

(cause == null ? null : cause.toString())

cause
viewId

ViewExpiredException

public ViewExpiredException(Throwable cause, String viewId)

Construct a new exception with the specified root cause. The detail message will be set to (cause == null ? null : cause.toString())

Parameters:
cause - The root cause for this exception
viewId - The view identifier for this exception
public ViewExpiredException(String message, Throwable cause, String viewId)

Construct a new exception with the specified detail message and root cause.

Parameters:
message - The detail message for this exception
cause - The root cause for this exception
viewId - The view identifier for this exception

Method Detail

public String getViewId()

null

ggetViewId

public String getViewId()

Return the view identifier of this exception, or null if the view
identifier is nonexistent or unknown.

public String getMessage()

null

getMessage

public String getMessage()

    Return the message for this exception prepended with the view identifier if the view identifier is not null, otherwise, return the message.

    Overrides:
        getMessage in class Throwable
javax.faces.application  **Class ViewHandler**

java.lang.Object
   ↓ javax.faces.application.ViewHandler

**Direct Known Subclasses:**  
ViewHandlerWrapper

---

```java
public abstract class ViewHandler extends Object

Extended by: ViewHandlerWrapper

ViewHandler  JavaServer Faces

ViewHandler StateManager StateManager

ViewHandler is the pluggability mechanism for allowing implementations of or applications using the JavaServer Faces specification to provide their own handling of the activities in the Render Response and Restore View phases of the request processing lifecycle. This allows for implementations to support different response generation technologies, as well as alternative strategies for saving and restoring the state of each view.

Please seeStateManager for information on how the ViewHandler interacts theStateManager.
```

---

**Field Summary**

<table>
<thead>
<tr>
<th>static String CHARACTER_ENCODING_KEY</th>
</tr>
</thead>
<tbody>
<tr>
<td>The key, in the session's attribute set, under which the response character encoding may be stored and retrieved.</td>
</tr>
<tr>
<td>static String DEFAULT_SUFFIX</td>
</tr>
<tr>
<td>-----------------------------</td>
</tr>
<tr>
<td>static String DEFAULT_SUFFIX_PARAM_NAME</td>
</tr>
</tbody>
</table>

## Constructor Summary

`ViewHandler()`

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>calculateCharacterEncoding(FacesContext context)</code></td>
<td>Returns the correct character encoding to be used for this request.</td>
</tr>
<tr>
<td><code>calculateLocale(FacesContext context)</code></td>
<td>Returns an appropriate <code>Locale</code> to use for this and subsequent requests for the current client.</td>
</tr>
<tr>
<td><code>calculateRenderKitId(FacesContext context)</code></td>
<td>Return an appropriate <code>renderKitId</code> for this and subsequent requests from the current client.</td>
</tr>
<tr>
<td><code>createView(FacesContext context, String viewId)</code></td>
<td>Create and return a new <code>UIViewRoot</code> instance initialized with information from the argument <code>FacesContext</code> and <code>viewId</code>.</td>
</tr>
<tr>
<td><code>getActionURL(FacesContext context, String viewId)</code></td>
<td>Return a URL suitable for rendering (after optional encoding performed by the <code>encodeActionURL()</code> method of <code>ExternalContext</code>) that selects the specified view identifier.</td>
</tr>
<tr>
<td><code>getResourceURL(FacesContext context, String path)</code></td>
<td>Return a URL suitable for rendering (after optional encoding performed by the <code>encodeResourceURL()</code> method of <code>ExternalContext</code>) that selects the specified web application resource.</td>
</tr>
<tr>
<td><code>initView(FacesContext context)</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Signature</td>
</tr>
<tr>
<td>--------</td>
<td>-----------</td>
</tr>
<tr>
<td>Initialize</td>
<td>void</td>
</tr>
<tr>
<td>Render View</td>
<td>abstract void <code>renderView(FacesContext context, UIViewRoot viewToRender)</code></td>
</tr>
<tr>
<td>Restore View</td>
<td>abstract <code>UIViewRoot restoreView(FacesContext context, String viewId)</code></td>
</tr>
<tr>
<td>Write State</td>
<td>abstract void <code>writeState(FacesContext context)</code></td>
</tr>
</tbody>
</table>

Methods inherited from class `java.lang.Object`:
- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

### Field Detail

**CHARACTER_ENCODING_KEY**

```java
public static final String CHARACTER_ENCODING_KEY
```

The key, in the session's attribute set, under which the response character encoding may be stored and retrieved.

**See Also:**
- [Constant Field Values](#)
**DEFAULT SUFFIX PARAM_NAME**

`public static final String DEFAULT_SUFFIX_PARAM_NAME`

Allow the web application to define an alternate suffix for pages containing JSF content. If this init parameter is not specified, the default value is taken from the value of the constant `DEFAULT_SUFFIX`.

**See Also:**
- [Constant Field Values](#)

---

**DEFAULT SUFFIX**

`public static final String DEFAULT_SUFFIX`

The value to use for the default extension if the webapp is using url extension mapping.

**See Also:**
- [Constant Field Values](#)

---

### Constructor Detail

**public ViewHandler()**

**ViewHandler**

**public ViewHandler()**

### Method Detail

`abstract public java.util.Locale calculateLocale(FacesContext context)`
Locale

context

Throws

FacesContext

NullPointerException: context null

calculateLocale

public abstract Locale calculateLocale(FacesContext context)

Returns an appropriate Locale to use for this and subsequent requests for the current client.

Parameters:

context - FacesContext for the current request

Throws:

NullPointerException - if context is null

calculateCharacterEncoding

public String calculateCharacterEncoding(FacesContext context)

- Content-Type charset

- charset getSession(boolean) false

ture ExternalContext#getSessionMap Map

ViewHandler#CHARACTER_ENCODING_KEY String

- null
**calculateCharacterEncoding**

```java
public String calculateCharacterEncoding(FacesContext context)
```

Returns the correct character encoding to be used for this request.

The following algorithm is employed.

- Examine the `Content-Type` request header. If it has a `charset` parameter, extract it and return that as the encoding.

- If no `charset` parameter was found, check for the existence of a session by calling `ExternalContext.getSession(boolean)` passing `false` as the argument. If that method returns `true`, get the session Map by calling `ExternalContext.getSessionMap()` and look for a value under the key given by the value of the symbolic constant `CHARACTER_ENCODING_KEY`. If present, return the value, converted to String.

- Otherwise, return `null`

**abstract public String calculateRenderKitId(FacesContext context)**

```java
renderKitId null
default javax.faces.render.RenderKitFactory#HTML_BASIC_RENDER_KIT
```

Throws `NullPointerException: context null`

**calculateRenderKitId**

```java
public abstract String calculateRenderKitId(FacesContext context)
```

Return an appropriate `renderKitId` for this and subsequent requests
from the current client. It is an error for this method to return null.

The default return value is RenderKitFactory.HTML_BASIC_RENDER_KIT.

Parameters:
  context - FacesContext for the current request

Throws:
  NullPointerException - if context is null

abstract public UIViewRoot createView(FacesContext context, String viewId)

FacesContext  viewId  UIViewRoot

FacesContext  ViewRoot  locale  renderKitId
#calculateLocale  #calculateRenderKitId  UIViewRoot
locale  renderKitId

Throws  NullPointerException:  context  null

createView

public abstract UIViewRoot createView(FacesContext context, String viewId)

Create and return a new UIViewRoot instance initialized with information from the argument FacesContext and viewId.

If there is an existing ViewRoot available on the FacesContext, this method must copy its locale and renderKitId to this new view root. If not, this method must call calculateLocale(javax.faces.context.FacesContext) and calculateRenderKitId(javax.faces.context.FacesContext), and store the results as the values of the locale and renderKitId, properties, respectively, of the newly created UIViewRoot.
**Throws:**

*NullPointerException* - if context is null

**abstract public String getActionURL(FacesContext context, String viewId)**

**URL**

*Web*  
*ExternalContext*  
*encodeActionURL()*

**Parameters:**

- **context** - *FacesContext* for this request
- **viewId** - View identifier of the desired view

**Throws:**

*IllegalArgumentException* - if viewId is not valid for this ViewHandler.

*NullPointerException* - if context or viewId is null.

**abstract public String getResourceURL(FacesContext context, String path)**

**URL**

*Web*  
*ExternalContext*  
*encodeResourceURL()*
getResourceURL

public abstract String getResourceURL(FacesContext context, String path)

Return a URL suitable for rendering (after optional encoding performed by the encodeResourceURL() method of ExternalContext) that selects the specified web application resource. If the specified path starts with a slash, it must be treated as context relative; otherwise, it must be treated as relative to the action URL of the current view.

Parameters:
context - FacesContext for the current request
path - Resource path to convert to a URL

Throws:
IllegalArgumentException - if viewId is not valid for this ViewHandler.
NullPointerException - if context or path is null.

public void initView(FacesContext context) throws FacesException

#calculateCharacterEncoding null
**initView**

public void **initView**(FacesContext context) throws FacesException

Initialize the view for the request processing lifecycle.

This method must be called at the beginning of the Restore View Phase of the Request Processing Lifecycle. It is responsible for performing any per-request initialization necessary to the operation of the lifecycle.

The default implementation calls calculateCharacterEncoding(javax.faces.context.FacesContext) and passes the result, if non-null into the ExternalContext.setRequestCharacterEncoding(java.lang.String) method.

**Throws:**

FacesException - if a problem occurs setting the encoding, such as the UnsupportedEncodingException thrown by the underlying Servlet or Portlet technology when the encoding is not supported.

---

**abstract public void renderView**(FacesContext context, UIViewRoot viewToRender) throws java.io.IOException, FacesException

FacesContext context

viewToRender
Throws: java.io.IOException: /
Throws: NullPointerException: context viewToRender null
Throws: FacesException: servlet

renderView

public abstract void renderView(FacesContext context, UIViewRoot viewToRender)
throws IOException, FacesException

Perform whatever actions are required to render the response view to the response object associated with the current FacesContext.

Parameters:
context - FacesContext for the current request
viewToRender - the view to render

Throws:
IOException - if an input/output error occurs
NullPointerException - if context or viewToRender is null
FacesException - if a servlet error occurs

abstract public UIViewRoot restoreView(FacesContext context, String viewId)

FacesContext viewId StateManager restoreView
viewId null

context FacesContext
viewId

Throws: NullPointerException: context null
Throws: FacesException: servlet

restoreView
public abstract UIViewRoot restoreView(FacesContext context, String viewId)

Perform whatever actions are required to restore the view associated with the specified FacesContext and viewId. It may delegate to the restoreView of the associatedStateManager to do the actual work of restoring the view. If there is no available state for the specified viewId, return null.

Parameters:
- context - FacesContext for the current request
- viewId - the view identifier for the current request

Throws:
- NullPointerException - if context is null
- FacesException - if a servlet error occurs

abstract public void writeState(FacesContext context) throws java.io.IOException

StateManager#writeState

context - FacesContext

Throws java.io.IOException: /

Throws NullPointerException: context null

writeState

public abstract void writeState(FacesContext context) throws IOException

Take any appropriate action to either immediately write out the current state information (by callingStateManager.writeState(javax.faces.context.FacesContext, java.lang.Object), or noting where state information should later be written.

Parameters:
context - FacesContext for the current request

Throws:

IOException - if an input/output error occurs

NullPointerException - if context is null
public abstract class `ViewHandlerWrapper` extends `ViewHandler`

Extends: `ViewHandler`

Provides a simple implementation of `ViewHandler` that can be subclassed by developers wishing to provide specialized behavior to an existing `ViewHandler` instance. The default implementation of all methods is to call through to the wrapped `ViewHandler`.

Usage: extend this class and override `getWrapped()` to return the instance we are wrapping.

Since: 1.2

---

### Field Summary

Fields inherited from class `javax.faces.application.ViewHandler`

<table>
<thead>
<tr>
<th>Field</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHARACTER_ENCODING_KEY</td>
</tr>
<tr>
<td>DEFAULT_SUFFIX</td>
</tr>
<tr>
<td>DEFAULT_SUFFIX_PARAM_NAME</td>
</tr>
</tbody>
</table>
### Constructor Summary

**ViewHandlerWrapper()**

### Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>calculateCharacterEncoding()</code></td>
<td>(<code>FacesContext context</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.calculateCharacterEncoding(javax.faces.context.FacesContext)</code> on the wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>calculateLocale()</code></td>
<td>(<code>FacesContext context</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.calculateLocale(javax.faces.context.FacesContext)</code> wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>calculateRenderKitId()</code></td>
<td>(<code>FacesContext context</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.calculateRenderKitId(javax.faces.context.FacesContext)</code> the wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>createView()</code></td>
<td>(<code>FacesContext context</code>, <code>String viewId</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.createView(javax.faces.context.FacesContext, String)</code> wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>getActionURL()</code></td>
<td>(<code>FacesContext context</code>, <code>String viewId</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.getActionURL(javax.faces.context.FacesContext, String)</code> the wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>getResourceURL()</code></td>
<td>(<code>FacesContext context</code>, <code>String path</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.getResourceURL(javax.faces.context.FacesContext, String)</code> the wrapped <code>ViewHandler</code> object.</td>
</tr>
<tr>
<td><code>getWrapped()</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>initView()</code></td>
<td>(<code>FacesContext context</code>)</td>
<td>The default behavior of this method is to call <code>ViewHandler.initView(javax.faces.context.FacesContext)</code> on the wrapped <code>ViewHandler</code> object.</td>
</tr>
</tbody>
</table>
void renderView(FacesContext context, UIViewRoot viewToRender)
   The default behavior of this method is to call
   ViewHandler.renderView(javax.faces.context.FacesContext,
   javax.faces.component.UIViewRoot) on the wrapped ViewHandler object.

UIViewRoot restoreView(FacesContext context, String viewId)
   The default behavior of this method is to call
   ViewHandler.restoreView(javax.faces.context.FacesContext,
   wrapped ViewHandler) object.

   writeState(FacesContext context)
   The default behavior of this method is to call
   ViewHandler.writeState(javax.faces.context.FacesContext) on
   ViewHandler object.

---

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

---

Constructor Detail

public ViewHandlerWrapper()

---

ViewHandlerWrapper

public ViewHandlerWrapper()

---

Method Detail

abstract protected ViewHandler getWrapped()
   return getWrapped
protected abstract ViewHandler getWrapped()

    Returns:
    the instance that we are wrapping.

public String calculateCharacterEncoding(FacesContext context)

    ViewHandler calculateCharacterEncoding(javax.faces.context.FacesContext)

    since 1.2
    See also calculateCharacterEncoding(javax.faces.context.FacesContext)

calculateCharacterEncoding

public String calculateCharacterEncoding(FacesContext context)

    The default behavior of this method is to call
    ViewHandler.calculateCharacterEncoding(javax.faces.context.FacesContext)
on the wrapped ViewHandler object.

    Overrides: calculateCharacterEncoding in class ViewHandler
    Since: 1.2
    See Also: ViewHandler.calculateCharacterEncoding(javax.faces.context.FacesContext)

public java.util.Locale calculateLocale(FacesContext context)

    ViewHandler calculateLocale(javax.faces.context.FacesContext)
**calculateLocale**

```java
public Locale calculateLocale(FacesContext context)
```

The default behavior of this method is to call `ViewHandler.calculateLocale(javax.faces.context.FacesContext)` on the wrapped `ViewHandler` object.

**Specified by:**
- `calculateLocale` in class `ViewHandler`

**Parameters:**
- `context` - `FacesContext` for the current request

**Since:**
- 1.2

**See Also:**
- `ViewHandler.calculateLocale(javax.faces.context.FacesContext)`

---

**public String calculateRenderKitId(FacesContext context)**

```java
ViewHandler calculateRenderKitId(javax.faces.context.FacesContext)
```

**since**
- 1.2

**See also**
- `calculateRenderKitId(javax.faces.context.FacesContext)`

---

**calculateRenderKitId**

```java
public String calculateRenderKitId(FacesContext context)
```

The default behavior of this method is to call `ViewHandler.calculateRenderKitId(javax.faces.context.FacesContext)` on the wrapped `ViewHandler` object.
public UIViewRoot createView(FacesContext context, String viewId)

ViewHandler createView(javax.faces.context.FacesContext, String)

Since: 1.2
See also createView(javax.faces.context.FacesContext, String)

cREATEVIEW

public UIViewRoot createView(FacesContext context, String viewId)

The default behavior of this method is to call ViewHandler.createView(javax.faces.context.FacesContext, String) on the wrapped ViewHandler object.

Specified by:
createView in class ViewHandler

Since: 1.2
See Also: ViewHandler.createView(javax.faces.context.FacesContext, String)

public String getActionURL(FacesContext context, String
public String getResourceURL(FacesContext context, String path)

ViewHandler
getResourceURL(javax.faces.context.FacesContext, String)

since 1.2
See also getResourceURL(javax.faces.context.FacesContext, String)
getResourceURL

public String getResourceURL(FacesContext context, String path)

The default behavior of this method is to call ViewHandler.getResourceURL(javax.faces.context.FacesContext, String) on the wrapped ViewHandler object.

Specified by: getResourceURL in class ViewHandler

Parameters:
- context - FacesContext for the current request
- path - Resource path to convert to a URL

Since: 1.2

See Also: ViewHandler.getResourceURL(javax.faces.context.FacesContext, String)

public void initView(FacesContext context) throws FacesException

ViewHandler initView

since 1.2

See also initView

initView

public void initView(FacesContext context) throws FacesException

The default behavior of this method is to call ViewHandler.initView(javax.faces.context.FacesContext) on the
wrapped ViewHandler object.

Overrides:
initView in class ViewHandler

Throws:
FacesException - if a problem occurs setting the encoding, such as the UnsupportedEncodingException thrown by the underlying Servlet or Portlet technology when the encoding is not supported.

Since:
1.2

See Also:
ViewHandler.initView(javax.faces.context.FacesContext)

public void renderView(FacesContext context, UIViewRoot viewToRender) throws java.io.IOException, FacesException

ViewHandler renderView(javax.faces.context.FacesContext, javax.faces.component.UIViewRoot)

since 1.2

See also renderView(javax.faces.context.FacesContext, java.util.FacesContext, javax.faces.component.UIViewRoot)

renderView

public void renderView(FacesContext context, UIViewRoot viewToRender) throws IOException, FacesException

The default behavior of this method is to call ViewHandler.renderView(javax.faces.context.FacesContext, javax.faces.component.UIViewRoot) on the wrapped ViewHandler object.
Specified by:
   renderView in class ViewHandler

Parameters:
   context - FacesContext for the current request
   viewToRender - the view to render

Throws:
   IOException - if an input/output error occurs
   FacesException - if a servlet error occurs

Since:
   1.2

See Also:
   ViewHandler.renderView(javax.faces.context.FacesContext,
                           javax.faces.componentUIViewRoot)

---

class UIViewRoot

publicUIViewRoot restoreView(FacesContext context, String viewId)

   ViewHandler
   restoreView(javax.faces.context.FacesContext,FacesContext,String)

since 1.2

See also restoreView(javax.faces.context.FacesContext, String)

---

_restoreView_

public UIViewRoot _restoreView_(FacesContext context, String viewId)

The default behavior of this method is to call
ViewHandler._restoreView_(javax.faces.context.FacesContext,
String) on the wrapped ViewHandler object.

Specified by:
   _restoreView_ in class ViewHandler

Parameters:
   context - FacesContext for the current request
   viewId - the view identifier for the current request
public void writeState(FacesContext context) throws java.io.IOException

ViewHandler
writeState(java.faces.context.FacesContext)

since 1.2
See also writeState(java.faces.context.FacesContext)

writeState

public void writeState(FacesContext context)
throws IOException

The default behavior of this method is to call
ViewHandler.writeState(java.faces.context.FacesContext) on the
wrapped ViewHandler object.

Specified by:
writeState in class ViewHandler

Parameters:
context - FacesContext for the current request

Throws:
IOException - if an input/output error occurs

Since: 1.2
See Also:
ViewHandler.writeState(java.faces.context.FacesContext)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  

Class W3CDomHandler

java.lang.Object  
  javax.xml.bind.annotation.W3CDomHandler

All Implemented Interfaces:
  DomHandler<Element,DOMResult>

public class W3CDomHandler
  extends Object
  implements DomHandler<Element,DOMResult>

Implements: ">DomHandler

W3C DOM  org.w3c.dom  DomHandler
  since  JAXB2.0

DomHandler implementation for W3C DOM (org.w3c.dom package.)

Since:  JAXB2.0
Author:  Kohsuke Kawaguchi

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3CDomHandler()</td>
<td>Default constructor.</td>
</tr>
<tr>
<td>W3CDomHandler(DocumentBuilder builder)</td>
<td>Constructor that allows applications to specify which DOM implementation to be used.</td>
</tr>
</tbody>
</table>

Method Summary
When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a Result object.

Once the portion is sent to the Result.

This method is called when a JAXB provider needs to marshal an element to XML.

Default constructor. It is up to a JAXB provider to decide which DOM implementation to use or how that is configured.
W3CDomHandler

public W3CDomHandler(DocumentBuilder builder)

Constructor that allows applications to specify which DOM implementation to be used.

Parameters:
  builder - must not be null. JAXB uses this DocumentBuilder to create a new element.

Method Detail

public javax.xml.parsers.DocumentBuilder getBuilder()

getBuilder

public DocumentBuilder getBuilder()

public void setBuilder(javax.xml.parsers.DocumentBuilder builder)

setBuilder

public void setBuilder(DocumentBuilder builder)

public javax.xml.transform.dom.DOMResult
createUnmarshaller(ValidationEventHandler errorHandler)

createUnmarshaller

public DOMResult createUnmarshaller(ValidationEventHandler errorHandler)

Description copied from interface: DomHandler
When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a Result object.

A JAXB provider will then send a portion of the XML into the given result. Such a portion always form a subtree of the whole XML document rooted at an element.

Specified by:
createUnmarshaller in interface DomHandler<Element, DOMResult>

Parameters:
errorHandler - if any error happens between the invocation of this method and the invocation of DomHandler.getElement(Result), they must be reported to this handler. The caller must provide a non-null error handler. The Result object created from this method may hold a reference to this error handler.

Returns:
null if the operation fails. The error must have been reported to the error handler.

public org.w3c.dom.Element
getElement(javax.xml.transform.dom.DOMResult r)

g getElement

public Element getElement(DOMResult r)
Description copied from interface: **DomHandler**
Once the portion is sent to the **Result**. This method is called by a JAXB provider to obtain the unmarshalled element representation.

Multiple invocations of this method may return different objects. This method can be invoked only when the whole sub-tree are fed to the **Result** object.

**Specified by:**  
**getElement** in interface DomHandler\<Element, DOMResult\>

**Parameters:**
- r - The **Result** object created by  
  DomHandler.createUnmarshaller(ValidationEventHandler).

**Returns:**
null if the operation fails. The error must have been reported to the error handler.

```
public javax.xml.transform.Source marshal(org.w3c.dom.Element element,  
  ValidationEventHandler errorHandler)
```

**marshal**

```
public Source marshal(Element element,  
  ValidationEventHandler errorHandler)
```

Description copied from interface: **DomHandler**
This method is called when a JAXB provider needs to marshal an element to XML.

If non-null, the returned **Source** must contain a whole document rooted at one element, which will then be weaved into a bigger document that the JAXB provider is marshalling.

**Specified by:**  
**marshal** in interface DomHandler\<Element, DOMResult\>
errorHandler - Receives any errors happened during the
process of converting an element into a Source. The caller must provide a non-null error handler.

**Returns:**
null if there was an error. The error should have been reported to the handler.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/permissions/).

PS:
javax.xml.ws Annotation Type WebEndpoint

```java
@Target(value=METHOD)
@Retention(value=RUNTIME)
@Documented
public @interface WebEndpoint
```

**Implements:** Annotation
@Target(value=METHOD)
@Retention(value=RUNTIME)
@Documented

```java
getPortName()
```

wsdl:service    wsdl:port    WebServiceClient
since           JAX-WS 2.0
See also        javax.xml.ws.WebServiceClient

Used to annotate the `getPortName()` methods of a generated service interface.

The information specified in this annotation is sufficient to uniquely identify a `wsdl:port` element inside a `wsdl:service`. The latter is determined based on the value of the `WebServiceClient` annotation on the generated service interface itself.

**Since:**
JAX-WS 2.0

**See Also:**
WebServiceClient

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>The local name of the endpoint.</td>
</tr>
</tbody>
</table>
abstract public String name()

name

public abstract String name

The local name of the endpoint.

Default: 

"""
javax.xml.ws  Annotation Type WebFault

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented  
public @interface WebFault

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented

bean

since  JAX-WS 2.0

Used to annotate service specific exception classes to customize to the local and namespace name of the fault element and the name of the fault bean.

**Since:**  
JAX-WS 2.0

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>faultBean</td>
<td>Fault bean name.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>Elements local name.</td>
</tr>
<tr>
<td>String</td>
<td>targetNamespace</td>
<td>Elements namespace name.</td>
</tr>
</tbody>
</table>

abstract public String name()
name

public abstract String name

Elements local name.

Default:
   """

abstract public String targetNamespace()

targetNamespace

public abstract String targetNamespace

Elements namespace name.

Default:
   """

abstract public String faultBean()

bean

faultBean

public abstract String faultBean

Fault bean name.

Default:
   """
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jws Annotation Type WebMethod

@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface WebMethod

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=METHOD)

Web Service  JAX-RPC 1.1  5
java.rmi.RemoteException

Customizes a method that is exposed as a Web Service operation. The associated method must be public and its parameters return value, and exceptions must follow the rules defined in JAX-RPC 1.1, section 5. The method is not required to throw java.rmi.RemoteException.

Author:
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>action</strong></td>
</tr>
<tr>
<td><strong>exclude</strong></td>
</tr>
<tr>
<td><strong>operationName</strong></td>
</tr>
</tbody>
</table>

abstract public String operationName()
wsdl:operation  specdefault  Java
operationName

public abstract String operationName

    Name of the wsdl:operation matching this method.

    Default: 


abstract public String action()

SOAP  soap

action

public abstract String action

    The action for this operation.

    For SOAP bindings, this determines the value of the soap action.

    Default: 


abstract public boolean exclude()

web

web

    since 2.0
public abstract boolean exclude

Marks a method to NOT be exposed as a web method.

Used to stop an inherited method from being exposed as part of this web service. If this element is specified, other elements MUST NOT be specified for the @WebMethod.

*This member-value is not allowed on endpoint interfaces.*

**Since:**
2.0

**Default:**
false
javax.jws Annotation Type WebParam

@Retention(value=RUNTIME)
@Target(value=PARAMETER)
public @interface WebParam

Implements: Annotation
Inner classes: WebParam.Mode
@Retention(value=RUNTIME)
@Target(value=PARAMETER)

Web Service  XML

Customizes the mapping of an individual parameter to a Web Service message part and XML element.

Author:
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

<table>
<thead>
<tr>
<th>Optional Element Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Header</strong></td>
</tr>
<tr>
<td><strong>boolean</strong> header</td>
</tr>
<tr>
<td>If true, the parameter is pulled from a message header rather than the message body.</td>
</tr>
</tbody>
</table>

| **Mode**                 |
| **WebParam.Mode** mode   |
| The direction in which the parameter is flowing (One of IN, OUT, or INOUT). |

| **Name**                 |
| **String** name          |
| Name of the parameter.   |

| **PartName**             |
| **String** partName      |
| The name of the wsdl:part representing this parameter. |

| **TargetNamespace**      |
| **String** targetNamespace |
| The XML namespace for the parameter. |
abstract public String name()

rpc @WebParam.partName wsdl:part XML

BARE OUT INOUT

specdefault BARE argN N arg0

@WebMethod

name

public abstract String name

Name of the parameter.

If the operation is rpc style and @WebParam.partName has not been specified, this is name of the wsdl:part representing the parameter.

If the operation is document style or the parameter maps to a header, this is the local name of the XML element representing the parameter.

A name MUST be specified if the operation is document style, the parameter style is BARE, and the mode is OUT or INOUT.

Default:

abstract public String partName()
wsdl:part
rpc BARE
partName

public abstract String partName

   The name of the wsdl:part representing this parameter.

   This is only used if the operation is rpc style or if the operation is document style and the parameter style is BARE.

   Since:
      2.0
   Default:
     ""

abstract public String targetNamespace()

XML

??

   specdefault             WRAPPED
   Web Service  targetNamespace

**targetNamespace**

public abstract String targetNamespace

   The XML namespace for the parameter.

   Only used if the operation is document style or the parameter maps to a header. If the target namespace is set to ??, this represents the empty namespace.
abstract public WebParam.Mode mode()
INOUT  INOUT

OUT  INOUT  Holder  JAX-WS 2.0 [5]  2.3.3
Holder  OUT  INOUT
    specdefault
    Holder  INOUT
    Holder  IN

mode

public abstract WebParam.Mode mode

The direction in which the parameter is flowing (One of IN, OUT, or INOUT).

The OUT and INOUT modes may only be specified for parameter types that conform to the definition of Holder types (JAX-WS 2.0 [5], section 2.3.3). Parameters that are Holder Types MUST be OUT or INOUT.

Default:
IN

abstract public boolean header()
true

header

public abstract boolean header
If true, the parameter is pulled from a message header rather than the message body.

Default:
false
javax.jws  **Enum WebParam.Mode**

**java.lang.Object**  
**java.lang.Enum<WebParam.Mode>**  
**javax.jws.WebParam.Mode**

**All Implemented Interfaces:**  
**Serializable, Comparable<WebParam.Mode>**

**Enclosing class:**  
**WebParam**

---

```java
public static enum WebParam.Mode
extends Enum<WebParam.Mode>
```

**Extends:** Enum<E>  
**Contained within:** WebParam

---

The direction in which the parameter flows

---

**Enum Constant Summary**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IN</td>
<td></td>
</tr>
<tr>
<td>INOUT</td>
<td></td>
</tr>
<tr>
<td>OUT</td>
<td></td>
</tr>
</tbody>
</table>

---

**Method Summary**

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>valueOf</strong>(String name)</td>
<td></td>
</tr>
</tbody>
</table>
static WebParam.Mode Returns the enum constant of this type with the specified name.

static WebParam.Mode[] values() Returns an array containing the constants of this enum type, in the order they're declared.

Methods inherited from class java.lang.Enum
clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

Methods inherited from class java.lang.Object
finalize, getClass, notify, notifyAll, wait, wait, wait

Enum Constant Detail

IN
public static final WebParam.Mode IN

OUT
public static final WebParam.Mode OUT

INOUT
public static final WebParam.Mode INOUT
**Method Detail**

**final** public static [WebParam.Mode][] values()  

**values**  

public static final WebParam.Mode[] values()  

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(WebParam.Mode c : WebParam.Mode.values())
    System.out.println(c);
```

**Returns:**

an array containing the constants of this enum type, in the order they're declared

---

**public static WebParam.Mode valueOf(String name)**  

**valueOf**  

public static WebParam.Mode valueOf(String name)  

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**

name - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**
IllegalArgumentException - if this enum type has no constant with the specified name

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FRAMES</th>
<th>NO FRAMES</th>
</tr>
</thead>
<tbody>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR</td>
</tr>
</tbody>
</table>
javax.security.jacc  Class WebResourcePermission

java.lang.Object  
   | java.security.Permission
   | javax.security.jacc.WebResourcePermission

All Implemented Interfaces:
   Serializable, Guard

public final class WebResourcePermission

extends Permission
implements Serializable

Extends: java.security.Permission
Implements: java.io.Serializable

Servlet Web WebResourcePermission

WebResourcePermission  Web

newPermissionCollection

version |
See also  java.security.Permission

Class for Servlet web resource permissions. A WebResourcePermission is a named permission and has actions.

The name of a WebResourcePermission (also referred to as the target name) identifies the Web resources to which the permission pertains.

Implementations of this class MAY implement newPermissionCollection or inherit its implementation from the super class.

Version:

%I% %E%
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WebResourcePermission(HttpServletRequest request)</code></td>
<td>Creates a new <code>WebResourcePermission</code> from the <code>HttpServletRequest</code> object.</td>
</tr>
<tr>
<td><code>WebResourcePermission(String name, String actions)</code></td>
<td>Creates a new <code>WebResourcePermission</code> with the specified name and actions.</td>
</tr>
<tr>
<td><code>WebResourcePermission(String urlPatternSpec, String[] HTTPMethods)</code></td>
<td>Creates a new <code>WebResourcePermission</code> with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods.</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean equals(Object o)</code></td>
<td>Checks two <code>WebResourcePermission</code> objects for equality.</td>
</tr>
<tr>
<td><code>String getActions()</code></td>
<td>Returns a canonical String representation of the actions of this <code>WebResourcePermission</code>.</td>
</tr>
<tr>
<td><code>int hashCode()</code></td>
<td>Returns the hash code value for this <code>WebResourcePermission</code>.</td>
</tr>
<tr>
<td><code>boolean implies(Permission permission)</code></td>
<td>Determines if the argument <code>Permission</code> is &quot;implied by&quot; this <code>WebResourcePermission</code>.</td>
</tr>
</tbody>
</table>

Methods inherited from class `java.security.Permission`: `checkGuard`, `getName`, `newPermissionCollection`, `toString`
Methods inherited from class java.lang.Object
clone, finalize, getClass, notify, notifyAll, wait, wait, wait

Constructor Detail

public WebResourcePermission(String name, String actions)
WebResourcePermission

Web URLPatternSpec

URLPatternList ::= URLPattern | URLPatternList colon URLPattern

URLPatternSpec ::= null | URLPattern | URLPattern colon URLPatternSpec

null URLPatternSpec  URLPattern ("/") exact
URLPattern URLPatternSpec  exact URLPattern
URLPatternSpec  URLPattern exactpath-prefixextension

Java Servlet URLPatternSpec

URLPatternList URLPatternList

• URLPatternList
• path-prefix URLPatternList exact path-prefix
• extension URLPatternList exact path-prefix
• "/" URLPatternList
• exact URLPatternList URLPatternSpec

actions HTTP actions
ExtensionMethod ::= any token as defined by RFC 2616
1*[ CTL ]

HTTPMethod ::= "GET" | "POST" | "PUT" | "DELETE" | "HEAD" | "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethod

HTTPMethodExceptionList ::= exclamationPoint HTTPMethodList

HTTPMethodSpec ::= null | HTTPMethodExceptionList | HTTPMethodList

HTTPMethodSpec
null HTTPMethodSpec URL HTTP
HTTPMethodSpec HTTPMethodExceptionList exclamationPoint

name
Web URLPatternSpecURLPatternSpec URLPattern
Web URLPatternSpec URLPattern
null URLPatternSpec URLPattern ("/")

actions HTTP null HTTP

WebResourcePermission

public WebResourcePermission(String name, String actions)

Creates a new WebResourcePermission with the specified name and actions.

The name contains a URLPatternSpec that identifies the web resources to which the permissions applies. The syntax of a
URLPatternSpec is as follows:

\[
\text{URLPatternList} ::= \text{URLPattern} \mid \text{URLPatternList colon } \text{URLPattern}
\]
\[
\text{URLPatternSpec} ::= \text{null} \mid \text{URLPattern} \mid \text{URLPattern colc}
\]

A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor. The empty string is an exact URLPattern, and may occur anywhere in a URLPatternSpec that an exact URLPattern may occur. The first URLPattern in a URLPatternSpec may be any of the pattern types, exact, path-prefix, extension, or default as defined in the Java Servlet Specification. When a URLPatternSpec includes a URLPatternList, the patterns of the URLPatternList identify the resources to which the permission does NOT apply and depend on the pattern type and value of the first pattern as follows:

- No pattern may exist in the URLPatternList that matches the first pattern.
- If the first pattern is a path-prefix pattern, only exact patterns matched by the first pattern and path-prefix patterns matched by, but different from, the first pattern may occur in the URLPatternList.
- If the first pattern is an extension pattern, only exact patterns that are matched by the first pattern and path-prefix patterns may occur in the URLPatternList.
- If the first pattern is the default pattern, "/", any pattern except the default pattern may occur in the URLPatternList.
- If the first pattern is an exact pattern a URLPatternList must not be present in the URLPatternSpec.

The actions parameter contains a comma seperated list of HTTP methods. The syntax of the actions parameter is defined as follows:

\[
\text{ExtensionMethod ::= any token as defined by RFC 2616 (that is, 1*[any CHAR except CTLs or separat}
\]
\[
\text{HTTPMethod ::= "GET" | "POST" | "PUT" | "DELETE" | "HE "OPTIONS" | "TRACE" | ExtensionMethod}
\]
If duplicates occur in the HTTPMethodSpec they must be eliminated by the permission constructor.

A null or empty string HTTPMethodSpec indicates that the permission applies to all HTTP methods at the resources identified by the URL pattern.

If the HTTPMethodSpec contains an HTTPMethodExceptionList (i.e., it begins with an exclamationPoint), the permission pertains to all methods except those occurring in the exception list.

**Parameters:**
- **name** - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor.

- **actions** - identifies the HTTP methods to which the permission pertains. If the value passed through this parameter is null or the empty string, then the permission pertains to all the possible HTTP methods.

```java
public WebResourcePermission(String urlPatternSpec, String[] HTTPMethods)
WebResourcePermission URLPatternSpec HTTP
```
WebResourcePermission

public WebResourcePermission(String urlPatternSpec, String[] HTTPMethods)

Creates a new WebResourcePermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods.

Parameters:
urlPatternSpec - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor.

HTTPMethods - an array of strings each element of which contains the value of an HTTP method. If the value passed through this parameter is null or is an array with no elements, then the permission pertains to all the possible HTTP methods.

public WebResourcePermission(HttpServletRequest request)

HttpServletRequest WebResourcePermission

Servlet HttpServletRequest contextPath
(HttpServletRequest.getContextPath()) requestURI
request (HttpServletRequest.getRequestURI()) "/"
WebResourcePermission

public WebResourcePermission(HttpServletRequest request)

Creates a new WebResourcePermission from the HttpServletRequest object.

Parameters:
request - the HttpServletRequest object corresponding to the Servlet operation to which the permission pertains. The permission name is the substring of the requestURI (HttpServletRequest.getRequestURI()) that begins after the contextPath (HttpServletRequest.getContextPath()). When the substring operation yields the string "/", the permission is constructed with the empty string as its name. The permission's actions field is obtained from HttpServletRequest.getMethod().

Method Detail

public boolean equals(Object o)
WebResourcePermission WebResourcePermission
URLPatternSpec URLPatternSpec
URLPatternSpec URLPatternSpec
URLPatternList URLPatternList

P1.implies(P2) && P2.implies(P1)  Permission P1 P2

o  WebResourcePermission WebResourcePermission
return WebResourcePermission WebResourcePermission true
equals

public boolean equals(Object o)

Checks two WebResourcePermission objects for equality. WebResourcePermission objects are equivalent if their URLPatternSpec and (canonicalized) actions values are equivalent. The URLPatternSpec of a reference permission is equivalent to that of an argument permission if their first patterns are equivalent, and the patterns of the URLPatternList of the reference permission collectively match exactly the same set of patterns as are matched by the patterns of the URLPatternList of the argument permission.

Two Permission objects, P1 and P2, are equivalent if and only if P1.implies(P2) && P2.implies(P1).

Specified by:
equals in class Permission

Parameters:
o - the WebResourcePermission object being tested for equality with this WebResourcePermission.

Returns:
true if the argument WebResourcePermission object is equivalent to this WebResourcePermission.

public String getActions()
WebResourcePermission HTTP
HTTP null

return WebResourcePermission null

getActions
public String getActions()

Returns a canonical String representation of the actions of this WebResourcePermission. In the canonical form, predefined methods precede extension methods, and within each method classification the corresponding methods occur in ascending lexical order. There may be no duplicate HTTP methods in the canonical form, and the canonical form of the set of all HTTP methods is the value null.

Specified by: getActions in class Permission

Returns: a String containing the canonicalized actions of this WebResourcePermission (or the null value).

public int hashCode()

WebResourcePermission

- Java WebResourcePermission hashCode
  WebResourcePermission hashCode
- equals WebResourcePermission
  Permission hashCode
  return

hashCode

public int hashCode()

Returns the hash code value for this WebResourcePermission. The properties of the returned hash code must be as follows:

- During the lifetime of a Java application, the hashCode method must return the same integer value, every time it is called on a WebResourcePermission object. The value returned by
hashCode for a particular WebResourcePermission need not remain consistent from one execution of an application to another.

- If two WebResourcePermission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result (within an application).

Specified by:

hashCode in class Permission

Returns:

the integer hash code value for this object.

---

public boolean implies(java.security.Permission
permission)
WebResourcePermission "" Permission
true

- WebResourcePermission
- permission URLPattern URLPattern
- permission URLPattern URLPatternSpec
  URLPatternList URLPattern
- permission URLPattern URLPatternSpec
  URLPattern URLPatternSpec URLPatternList
  URLPattern permission URLPatternList
  URLPattern
- permission HTTP HTTP

URLPattern Servlet URL

- String
- path-prefix "/*"
- path-prefix "/" "/*" 2
  "/"
- extension "*."
**implies**

```java
public boolean implies(Permission permission)
```

Determines if the argument Permission is "implied by" this WebResourcePermission. For this to be the case, all of the following must be true:

- The argument is an instanceof WebResourcePermission
- The first URLPattern in the name of the argument permission is matched by the first URLPattern in the name of this permission.
- The first URLPattern in the name of the argument permission is NOT matched by any URLPattern in the URLPatternList of the URLPatternSpec of this permission.
- If the first URLPattern in the name of the argument permission matches the first URLPattern in the URLPatternSpec of this permission, then every URLPattern in the URLPatternList of the URLPatternSpec of this permission is matched by a URLPattern in the URLPatternList of the argument permission.
- The HTTP methods represented by the actions of the argument permission are a subset of the HTTP methods represented by the actions of this permission.

URLPattern matching is performed using the *Servlet matching rules* where two URL patterns match if they are related as follows:

- their pattern values are String equivalent, or
- this pattern is the path-prefix pattern "/*/", or
- this pattern is a path-prefix pattern (that is, it starts with "/" and
ends with "/*/" and the argument pattern starts with the substring of this pattern, minus its last 2 characters, and the next character of the argument pattern, if there is one, is "/", or

- this pattern is an extension pattern (that is, it starts with "*.") and the argument pattern ends with this pattern, or
- the reference pattern is the special default pattern, "/", which matches all argument patterns.

All of the comparisons described above are case sensitive.

**Specified by:**

`implies` in class `Permission`

**Parameters:**

`permission` - "this" WebResourcePermission is checked to see if it implies the argument permission.

**Returns:**

true if the specified permission is implied by this object, false if not.
javax.jws Annotation Type WebResult

@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface WebResult

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value=METHOD)

WSDL  XML

Customizes the mapping of the return value to a WSDL part and XML element.

**Author:**
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean</td>
<td>header</td>
<td>If true, the result is pulled from a message header rather than the message body.</td>
</tr>
<tr>
<td>String</td>
<td>name</td>
<td>Name of return value.</td>
</tr>
<tr>
<td>String</td>
<td>partName</td>
<td>The name of the wsdl:part representing this return value.</td>
</tr>
<tr>
<td>String</td>
<td>targetNamespace</td>
<td>The XML namespace for the return value.</td>
</tr>
</tbody>
</table>

**abstract public String name()**
public abstract String name

Name of return value.

If the operation is rpc style and @WebResult.partName has not been specified, this is the name of the wsdl:part representing the return value.
If the operation is document style or the return value maps to a header, this is the local name of the XML element representing the return value.

Default: ""

abstract public String partName()

wsdl:part

rpc BARE

specdefault since {@code

partName

public abstract String partName

The name of the wsdl:part representing this return value.
This is only used if the operation is rpc style, or if the operation is document style and the parameter style is BARE.

**Since:**
2.0

**Default:**
""

```java
abstract public String targetNamespace()
XML
```

**targetNamespace**

```java
public abstract String targetNamespace
```

The XML namespace for the return value.

Only used if the operation is document style or the return value maps to a header. If the target namespace is set to `??`, this represents the empty namespace.

**Default:**
""

```java
abstract public boolean header()
true
```

**header**

```java
since 2.0
```
public abstract boolean header

If true, the result is pulled from a message header rather than the message body.

Since:
2.0
Default:
false

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.security.jacc Class WebRoleRefPermission

java.lang.Object  
  - java.security.Permission  
    - javax.security.jacc.WebRoleRefPermission

All Implemented Interfaces: Serializable, Guard

public final class WebRoleRefPermission

extends Permission
implements Serializable

Extends: java.security.Permission
Implements: java.io.Serializable

Servlet isUserInRole (String reference) WebRoleRefPermission

WebRoleRefPermission servlet Web servlet
.isUserInRole (String reference)

WebRoleRefPermission WebRoleRefPermission

newPermissionCollection

See also java.security.Permission

Class for Servlet isUserInRole (String reference) permissions. A WebRoleRefPermission is a named permission and has actions.

The name of an WebRoleRefPermission (also referred to as the target name) identifies a Web resource by the servlet name (in the deployment descriptor corresponding to the component from which the call to isUserInRole (String reference) is being made.
The actions of a WebRoleRefPermission identifies the role reference to which the permission applies. A WebRoleRefPermission is checked to determine if the subject is a member of the role identified by the reference.

Implementations of this class MAY implement newPermissionCollection or inherit its implementation from the super class.

**Author:**
Ron Monzillo, Gary Ellison

**See Also:**
Permission, Serialized Form

---

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebRoleRefPermission(String name, String actions)</td>
<td>Creates a new WebRoleRefPermission with the specified name and actions.</td>
</tr>
</tbody>
</table>

---

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object o)</td>
<td>Checks two WebRoleRefPermission objects for equality.</td>
</tr>
<tr>
<td>String getActions()</td>
<td>Returns a canonical String representation of the actions of this WebRoleRefPermission.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Returns the hash code value for this WebRoleRefPermission.</td>
</tr>
<tr>
<td>boolean implies(Permission permission)</td>
<td>Determines if the argument Permission is &quot;implied by&quot; this WebRoleRefPermission.</td>
</tr>
</tbody>
</table>

---

Methods inherited from class java.security.Permission
checkGuard, getName, newPermissionCollection, toString
Methods inherited from class java.lang.**Object**
clone, finalize, getClass, notify, notifyAll, wait, wait, wait

**Constructor Detail**

public **WebRoleRefPermission**(String name, String actions)

WebRoleRefPermission

**Parameters:**
- **name** - the servlet-name that identifies the application specific web resource in whose context the role references are to be evaluated.

- **actions** - identifies the role reference to which the permission pertains. The role reference is scoped to the Web resource identified in the name parameter. The value of the role reference must not be null or the empty string.

**Method Detail**

public boolean equals(Object o)
WebRoleRefPermission  WebRoleRefPermission

P1.implies(P2) && P2.implies(P1)  Permission P1
P2

o  WebRoleRefPermission  WebRoleRefPermission
return  EJBRoleRefPermission  EJBRoleRefPermission
true

equals

public boolean equals(Object o)

Checks two WebRoleRefPermission objects for equality. WebRoleRefPermission objects are equivalent if they have case equivalent name and actions values.

Two Permission objects, P1 and P2, are equivalent if and only if P1.implies(P2) && P2.implies(P1).

The name and actions comparisons described above are case sensitive.

Specified by:  
equals in class Permission

Parameters:
  o - the WebRoleRefPermission object being tested for equality with this WebRoleRefPermission.

Returns:
  true if the argument WebRoleRefPermission object is equivalent to this WebRoleRefPermission.
public String getActions()
WebRoleRefPermission

return WebRoleRefPermission

getActions

public String getActions()

Returns a canonical String representation of the actions of this WebRoleRefPermission.

Specified by:
getActions in class Permission

Returns:
a String containing the canonicalized actions of this WebRoleRefPermission.

public int hashCode()
WebRoleRefPermission

• Java WebRoleRefPermission hashCode
WebRoleRefPermission hashCode
• equals WebRoleRefPermission Permission hashCode

return

hashCode

public int hashCode()
Returns the hash code value for this WebRoleRefPermission. The properties of the returned hash code must be as follows:

- During the lifetime of a Java application, the hashCode method must return the same integer value, every time it is called on a WebRoleRefPermission object. The value returned by hashCode for a particular WebRoleRefPermission need not remain consistent from one execution of an application to another.
- If two WebRoleRefPermission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result (within an application).

Specified by:
hashCode in class Permission

Returns:
the integer hash code value for this object.

public boolean implies(java.security.Permission permission)
WebRoleRefPermission “” Permission

- WebRoleRefPermission
- WebRoleRefPermission
- WebRoleRefPermission

permission "this" WebRoleRefPermission
return true false

implies
public boolean implies(Permission permission)

Determines if the argument Permission is "implied by" this WebRoleRefPermission. For this to be the case,

- The argument must be an instance of WebRoleRefPermission
- with name equivalent to this WebRoleRefPermission, and
- with role reference equivalent to this WebRoleRefPermission (as defined in their actions).

The comparisons described above are case sensitive.

Specified by:
implies in class Permission

Parameters:
permission - "this" WebRoleRefPermission is checked to see if it implies the argument permission.

Returns:
true if the specified permission is implied by this object, false if not.
javax.jws Annotation Type WebService

```
@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface WebService
```

**Implements:** Annotation
```
@Retention(value=RUNTIME)
@Target(value=TYPE)
```

Java Web Service Java Web Service

Marks a Java class as implementing a Web Service, or a Java interface as defining a Web Service interface.

**Author:**
Copyright (c) 2004 by BEA Systems, Inc. All Rights Reserved.

---

**Optional Element Summary**

<table>
<thead>
<tr>
<th>String</th>
<th>endpointInterface</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The complete name of the service endpoint interface defining the service's abstract Web Service contract.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The name of the Web Service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>portName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The port name of the Web Service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>serviceName</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>The service name of the Web Service.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>targetNamespace</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If the @WebService.targetNamespace annotation is on a service endpoint interface, the targetNamespace is used for the namespace for the wsdl:portType (and associated XML elements).</td>
</tr>
</tbody>
</table>
String wsdlLocation

The location of a pre-defined WSDL describing the service.

abstract public String name()

Web Service

WSDL 1.1  wsdl:portType

specdefault  Java

name

public abstract String name

The name of the Web Service.

Used as the name of the wsdl:portType when mapped to WSDL 1.1.

Default:

abstract public String targetNamespace()

@WebService.targetNamespace  targetNamespace

wsdl:portType  XML

endpointInterface

bean  targetNamespace  wsdl:portType

wsdl:service  XML

@WebService.targetNamespace  endpointInterface

bean  targetNamespace  wsdl:service

XML

specdefault  JAX-WS 2.0 [5]  3.2
targetNamespace

public abstract String targetNamespace

If the @WebService.targetNamespace annotation is on a service endpoint interface, the targetNamespace is used for the namespace for the wsdl:portType (and associated XML elements).

If the @WebService.targetNamespace annotation is on a service implementation bean that does NOT reference a service endpoint interface (through the endpointInterface attribute), the targetNamespace is used for both the wsdl:portType and the wsdl:service (and associated XML elements).

If the @WebService.targetNamespace annotation is on a service implementation bean that does reference a service endpoint interface (through the endpointInterface attribute), the targetNamespace is used for only the wsdl:service (and associated XML elements).

Default:

""

abstract public String serviceName()

Web Service

WSDL 1.1 wsdl:service

specdefault Java class + Service

serviceName
public abstract String serviceName

The service name of the Web Service.
Used as the name of the wsdl:service when mapped to WSDL 1.1.

*This member-value is not allowed on endpoint interfaces.*

**Default:**

```
""
```

---

abstract public String portName()

**Web Service**

**WSDL 1.1  wsdl:port**

specdefault @WebService.name+?Port?.

since 2.0

**portName**

public abstract String portName

The port name of the Web Service.
Used as the name of the wsdl:port when mapped to WSDL 1.1.

*This member-value is not allowed on endpoint interfaces.*

**Since:**

2.0

**Default:**

```
""
```
abstract public String wsdlLocation()
WSDL

wsdlLocation WSDL URLwsdlLocation
bean WSDL bean WSDL portType
JSR-181 WSDL portType bean Web
Service portType

wsdlLocation

public abstract String wsdlLocation

The location of a pre-defined WSDL describing the service.

The wsdlLocation is a URL (relative or absolute) that refers to a pre-existing WSDL file. The presence of a wsdlLocation value indicates that the service implementation bean is implementing a pre-defined WSDL contract. The JSR-181 tool MUST provide feedback if the service implementation bean is inconsistent with the portType and bindings declared in this WSDL. Note that a single WSDL file might contain multiple portTypes and multiple bindings. The annotations on the service implementation bean determine the specific portType and bindings that correspond to the Web Service.

Default: 

""

abstract public String endpointInterface()
Web

WSDL portType Java WSDL JSR-181
Bean
Web Bean
endpointInterface

```java
public abstract String endpointInterface
```

The complete name of the service endpoint interface defining the service's abstract Web Service contract.

This annotation allows the developer to separate the interface contract from the implementation. If this annotation is present, the service endpoint interface is used to determine the abstract WSDL contract (portType and bindings). The service endpoint interface MAY include JSR-181 annotations to customize the mapping from Java to WSDL.

The service implementation bean MAY implement the service endpoint interface, but is not REQUIRED to do so.

If this member-value is not present, the Web Service contract is generated from annotations on the service implementation bean. If a service endpoint interface is required by the target environment, it will be generated into an implementation-defined package with an implementation-defined name.

*This member-value is not allowed on endpoint interfaces.*

**Default:**

```
PS:
javax.xml.ws Annotation Type WebServiceClient

@Target(value=TYPE)
@Retention(value=RUNTIME)
@Documented
public @interface WebServiceClient

**Implements:** Annotation
@Target(value=TYPE)
@Retention(value=RUNTIME)
@Documented

WSDL  wsdl:service  wsdl:service Web
      since        JAX-WS 2.0

Used to annotate a generated service interface.

The information specified in this annotation is sufficient to uniquely identify a `wsdl:service` element inside a WSDL document. This `wsdl:service` element represents the Web service for which the generated service interface provides a client view.

**Since:**
JAX-WS 2.0

Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
<th>The local name of the Web service.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>targetNamespace</td>
<td>The namespace for the Web service.</td>
</tr>
<tr>
<td>String</td>
<td>wsdlLocation</td>
<td>The location of the WSDL document for the service (a</td>
</tr>
</tbody>
</table>
abstract public String name()
Web

name

public abstract String name

    The local name of the Web service.

    Default: 
    

abstract public String targetNamespace()
Web
targetNamespace

public abstract String targetNamespace

    The namespace for the Web service.

    Default: 
    

abstract public String wsdlLocation()
WSDL (URL)

wsdlLocation

public abstract String wsdlLocation
The location of the WSDL document for the service (a URL).

**Default:**

"""
javax.xml.ws Interface WebServiceContext

public interface WebServiceContext

WebServiceContext Web

since JAX-WS 2.0
See also javax.annotation.Resource

A WebServiceContext makes it possible for a web service endpoint implementation class to access message context and security information relative to a request being served. Typically a WebServiceContext is injected into an endpoint implementation class using the Resource annotation.

Since: JAX-WS 2.0
See Also: Resource

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getMessageContext()</td>
<td>Returns the MessageContext for the request being served at the time this method is called.</td>
</tr>
<tr>
<td>getUserPrincipal()</td>
<td>Returns the Principal that identifies the sender of the request currently being serviced.</td>
</tr>
<tr>
<td>boolean isUserInRole(String role)</td>
<td>Returns a boolean indicating whether the authenticated user is included in the specified logical role.</td>
</tr>
</tbody>
</table>
public MessageContext getMessageContext()
MessageContext APPLICATION

    return MessageContext

    Throws IllegalStateException:

See also javax.xml.ws.handler.MessageContext,
javax.xml.ws.handler.MessageContext.Scope,
java.lang.IllegalStateException

getMessageContext

    MessageContext getMessageContext()

    Returns the MessageContext for the request being served at the
time this method is called. Only properties with APPLICATION scope
will be visible to the application.

    Returns: MessageContext The message context.

    Throws: IllegalStateException - This exception is thrown if the method
is called while no request is being serviced.

See Also: MessageContext, MessageContext.Scope, IllegalStateException

public java.security.Principal getUserPrincipal()
Principal

    return Principal

    Throws IllegalStateException:

See also java.security.Principal, java.lang.IllegalStateException

getUserPrincipal
**Principal** `getUserPrincipal()`

Returns the Principal that identifies the sender of the request currently being serviced. If the sender has not been authenticated, the method returns `null`.

**Returns:**
Principal The principal object.

**Throws:**
`IllegalStateException` - This exception is thrown if the method is called while no request is being serviced.

**See Also:**
Principal, `IllegalStateException`

---

**public boolean isUserInRole(String role)**

`false`

```java
role String
return boolean
Throws IllegalArgumentException: IllegalStateException:
```

**isUserInRole**

boolean `isUserInRole(String role)`

Returns a boolean indicating whether the authenticated user is included in the specified logical role. If the user has not been authenticated, the method returns false.

**Parameters:**
role - A String specifying the name of the role

**>Returns:**
a boolean indicating whether the sender of the request belongs to a given role

**Throws:**
`IllegalStateException` - This exception is thrown if the method is called while no request is being serviced.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Java class `WebServiceException` is the base exception class for all JAX-WS API runtime exceptions.

The `WebServiceException` class is the base exception class for all JAX-WS API runtime exceptions.

### Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WebServiceException()</code></td>
<td>Constructs a new exception with <code>null</code> as its detail message.</td>
</tr>
</tbody>
</table>
### Constructor Detail

**public WebServiceException()**

nullcause

**WebServiceException**

**public WebServiceException()**

Constructs a new exception with null as its detail message. The cause is not initialized.
public WebServiceException(String message)

Constructs a new exception with the specified detail message. The cause is not initialized.

Parameters:
message - The detail message which is later retrieved using the getMessage method

public WebServiceException(String message, Throwable cause)

Constructs a new exception with the specified detail message and cause.

Parameters:
message - The detail message which is later retrieved using the getMessage method
cause - The cause which is saved for the later retrieval throw by the getCause method
public WebServiceException(Throwable cause)

cause (cause==null ? null : cause.toString)

WebServiceException

cause WebServiceException

cause cause getCause

WebServiceException

public WebServiceException((Throwable cause)

Constructs a new WebServiceException with the specified cause and a detail message of (cause==null ? null : cause.toString()) (which typically contains the class and detail message of cause).

Parameters:

cause - The cause which is saved for the later retrieval throw by the getCause method. (A null value is permitted, and indicates that the cause is nonexistent or unknown.)

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
</tr>
<tr>
<td>FRAMES</td>
<td>NO FRAMES</td>
</tr>
<tr>
<td>DETAIL: FIELD</td>
<td>CONSTR I METHOD</td>
</tr>
</tbody>
</table>
javax.xml.ws  **Class WebServicePermission**

`java.lang.Object`  
  `java.security.Permission`  
    `java.security.BasicPermission`  
      `javax.xml.ws.WebServicePermission`

**All Implemented Interfaces:**  
  Serializable, Guard

```java
public final class WebServicePermission
extends BasicPermission

Extends: java.security.Permission > java.security.BasicPermission

Web
Web ""

publishEndpoint

publishEndpoint    javax.xml.ws.Endpoint    publish Web

See also    javax.xml.ws.Endpoint, java.security.BasicPermission,  
            java.security.Permission, java.security.Permissions,  
            java.lang.SecurityManager
```

This class defines web service permissions.

Web service Permissions are identified by name (also referred to as a  
"target name") alone. There are no actions associated with them.

The following permission target name is defined:

publishEndpoint
The `publishEndpoint` permission allows publishing a web service endpoint using the `publish` methods defined by the `javax.xml.ws.Endpoint` class.

See Also:

- `Endpoint`, `BasicPermission`, `Permission`, `Permissions`, `SecurityManager`, `Serialized Form`
WebServicePermission

public WebServicePermission(String name)

Creates a new permission with the specified name.

Parameters:
name - the name of the WebServicePermission

public WebServicePermission(String name, String actions)

name - WebServicePermission
actions - null

WebServicePermission

public WebServicePermission(String name, String actions)

Creates a new permission with the specified name and actions. The actions parameter is currently unused and it should be null.

Parameters:
name - the name of the WebServicePermission
actions - should be null
PS:
javax.xml.ws Annotation Type WebServiceProvider

@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented  
public @interface WebServiceProvider

**Implements:** Annotation  
@Target(value=TYPE)  
@Retention(value=RUNTIME)  
@Documented

Provider

**since** JAX-WS 2.0  
**See also**  
[javax.xml.ws.Provider](http://docs.oracle.com/javase/8/docs/api/javax/xml/ws/Provider.html)

Used to annotate a Provider implementation class.

**Since:**  
JAX-WS 2.0  
**See Also:**  
[javax.xml.ws.Provider](http://docs.oracle.com/javase/8/docs/api/javax/xml/ws/Provider.html)

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>portName</td>
<td>Port name.</td>
</tr>
<tr>
<td>String</td>
<td>serviceName</td>
<td>Service name.</td>
</tr>
<tr>
<td>String</td>
<td>targetNamespace</td>
<td>Target namespace for the service</td>
</tr>
<tr>
<td>String</td>
<td>wsdlLocation</td>
<td>Location of the WSDL description for the service.</td>
</tr>
</tbody>
</table>
abstract public String wsdlLocation()

wsdlLocation

public abstract String wsdlLocation

Location of the WSDL description for the service.

Default: ""

abstract public String serviceName()

serviceName

public abstract String serviceName

Service name.

Default: ""

abstract public String targetNamespace()

targetNamespace

public abstract String targetNamespace

Target namespace for the service
abstract public String portName()

portName

public abstract String portName

Port name.

Default:

""

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.ws  Annotation Type WebServiceRef

@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
@Documented
public @interface WebServiceRef

Implements: Annotation
@Target(value={TYPE, METHOD, FIELD})
@Retention(value=RUNTIME)
@Documented

WebServiceRef Web

since JAX-WS 2.0
See also javax.annotation.Resource

The WebServiceRef annotation is used to define a reference to a web service and (optionally) an injection target for it. Web service references are resources in the Java EE 5 sense.

Since:
JAX-WS 2.0
See Also:
Resource

Optional Element Summary

<table>
<thead>
<tr>
<th>Optional Element</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>mappedName</td>
<td>A product specific name that this resource should be mapped to.</td>
</tr>
<tr>
<td>name</td>
<td>The JNDI name of the resource.</td>
</tr>
<tr>
<td>type</td>
<td>The Java type of the resource.</td>
</tr>
<tr>
<td>value</td>
<td></td>
</tr>
<tr>
<td><strong>Class</strong></td>
<td>The service class, always a type extending javax.xml.ws.Service.</td>
</tr>
<tr>
<td>-----------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>String</strong></td>
<td><code>wsdlLocation</code> A URL pointing to the WSDL document for the web service.</td>
</tr>
</tbody>
</table>

```java
abstract public String name()
JNDI  JavaBean

name
```

```java
public abstract String name

The JNDI name of the resource. For field annotations, the default is the field name. For method annotations, the default is the JavaBeans property name corresponding to the method. For class annotations, there is no default and this must be specified.

**Default:**
""
```

```java
abstract public Class<T> type()
Java  JavaBean

type
```

```java
public abstract Class type

The Java type of the resource. For field annotations, the default is the type of the field. For method annotations, the default is the type of the JavaBeans property. For class annotations, there is no default and this must be specified.

**Default:**
java.lang.Object.class
```
A product specific name that this resource should be mapped to. The name of this resource, as defined by the `name` element or defaulted, is a name that is local to the application component using the resource. (It's a name in the JNDI `java:comp/env` namespace.) Many application servers provide a way to map these local names to names of resources known to the application server. This mapped name is often a *global* JNDI name, but may be a name of any form.

Application servers are not required to support any particular form or type of mapped name, nor the ability to use mapped names. The mapped name is product-dependent and often installation-dependent. No use of a mapped name is portable.

**Default:**

```
"
```
The service class, always a type extending javax.xml.ws.Service. This element must be specified whenever the type of the reference is a service endpoint interface.

Default:
java.lang.Object.class

abstract public String wsdlLocation()
Web WSDL URL WSDL

wsdlLocation

public abstract String wsdlLocation

A URL pointing to the WSDL document for the web service. If not specified, the WSDL location specified by annotations on the resource type is used instead.

Default:


javax.xml.ws  Annotation Type WebServiceRefs

@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface WebServiceRefs

**Implements:** Annotation
@Documented
@Retention(value=RUNTIME)
@Target(value=TYPE)

WebServiceRefs  web

<table>
<thead>
<tr>
<th>since</th>
<th>2.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>See also</td>
<td>javax.xml.ws.WebServiceRef</td>
</tr>
</tbody>
</table>

The WebServiceRefs annotation allows multiple web service references to be declared at the class level.

**Since:** 2.0

**See Also:**
WebServiceRef

---

### Required Element Summary

<table>
<thead>
<tr>
<th>WebServiceRef[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Array used for multiple web service reference declarations.</td>
</tr>
</tbody>
</table>

---

### Element Detail

abstract public **WebServiceRef[]** value()
web

value

public abstract WebServiceRef[] value

Array used for multiple web service reference declarations.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>OVERVIEW</th>
<th>PACKAGE</th>
<th>TREE</th>
<th>DEPRECATED</th>
<th>INDEX</th>
<th>HELP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>SUMMARY: NESTED</td>
<td>FIELD</td>
<td>CONSTR</td>
<td>METHOD</td>
</tr>
</tbody>
</table>
javax.security.jacc Class WebUserDataPermission

java.lang.Object
  ↓ java.security.Permission
  ↓ javax.security.jacc.WebUserDataPermission

All Implemented Interfaces:
    Serializable, Guard

public final class WebUserDataPermission
    extends Permission
    implements Serializable

Extends: java.security.Permission
Implements: java.io.Serializable

Servlet Web WebUserDataPermission

WebUserDataPermission URL Web
   See also                java.security.Permission

Class for Servlet Web user data permissions. A WebUserDataPermission is a named permission and has actions.

The name of a WebUserDataPermission (also referred to as the target name) identifies a Web resource by its context path relative URL pattern.

Author:
    Ron Monzillo, Gary Ellison

See Also:
    Permission, Serialized Form

Constructor Summary
<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WebUserDataPermission(HttpServletRequest request)</td>
<td>Creates a new WebUserDataPermission from the HttpServletRequest object.</td>
</tr>
<tr>
<td>WebUserDataPermission(String name, String actions)</td>
<td>Creates a new WebUserDataPermission with the specified name and actions.</td>
</tr>
<tr>
<td>WebUserDataPermission(String urlPatternSpec, String[] HTTPMethods, String transportType)</td>
<td>Creates a new WebUserDataPermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods and the transport type.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean equals(Object o)</td>
<td>Checks two WebUserDataPermission objects for equality.</td>
</tr>
<tr>
<td>String getActions()</td>
<td>Returns a canonical String representation of the actions of this WebUserDataPermission.</td>
</tr>
<tr>
<td>int hashCode()</td>
<td>Returns the hash code value for this WebUserDataPermission.</td>
</tr>
<tr>
<td>boolean implies(Permission permission)</td>
<td>Determines if the argument Permission is &quot;implied by&quot; this WebUserDataPermission.</td>
</tr>
</tbody>
</table>

Methods inherited from class java.security.Permission:
- checkGuard
- getName
- newPermissionCollection
- toString

Methods inherited from class java.lang.Object:
- clone
- finalize
- getClass
- notify
- notifyAll
- wait
- wait
- wait

### Constructor Detail
public WebUserDataPermission(String name, String actions)  
WebUserDataPermission

Web  URLPatternSpec

URLPatternList ::= URLPattern | URLPatternList colon URLPattern

URLPatternSpec ::= null | URLPattern | URLPattern colon URLPattern

null URLPatternSpec  URLPattern ("/") exact
URLPattern URLPatternSpec exact URLPattern
URLPatternSpec  URLPattern exactpathprefixextension  

Java Servlet  URLPatternSpec
URLPatternList URLPatternList

- URLPatternList
- path-prefix URLPatternList exact
  path-prefix
- extension URLPatternList exact path-prefix
- "/" URLPatternList
- exact URLPatternList URLPatternSpec

actions HTTP HTTP transportType

ExtensionMethod ::= any token as defined by RFC 2616
1*[ CTL ]

HTTPMethod ::= "Get" | "POST" | "PUT" | "DELETE" | "HEAD" | "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethodList
HTTPMethodExceptionList ::= ! HTTPMethodList

HTTPMethodSpec ::= emptyString | HTTPMethodExceptionList | HTTPMethodList

transportType ::= "INTEGRAL" | "CONFIDENTIAL" | "NONE"

actions ::= null | HTTPMethodSpec |

HTTPMethodSpec colon transportType

HTTPMethodSpec

HTTPMethodSpec HTTP

HTTPMethodSpec HTTPMethodExceptionList exclamationPoint

transportType "NONE" TransportType

transportType "NONE"

Web URLPatternSpecURLPatternSpec URLPattern
Web URLPatternSpec NULLPatternnull
URLPatternSpec URLPattern("/")

name actions HTTP null HTTP transportType "NONE"

WebUserDataPermission

public WebUserDataPermission(String name,
 String actions)

Creates a new WebUserDataPermission with the specified name and actions.
The name contains a URLPatternSpec that identifies the web resources to which the permissions applies. The syntax of a URLPatternSpec is as follows:

\[
\begin{align*}
\text{URLPatternList} & ::= \text{URLPattern} \mid \text{URLPatternList} \colon \text{URLPattern} \\
\text{URLPatternSpec} & ::= \text{null} \mid \text{URLPattern} \mid \text{URLPattern colc}
\end{align*}
\]

A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor. The empty string is an exact URLPattern, and may occur anywhere in a URLPatternSpec that an exact URLPattern may occur. The first URLPattern in a URLPatternSpec may be any of the pattern types, exact, path-prefix, extension, or default as defined in the Java Servlet Specification. When a URLPatternSpec includes a URLPatternList, the patterns of the URLPatternList identify the resources to which the permission does NOT apply and depend on the pattern type and value of the first pattern as follows:

- No pattern may exist in the URLPatternList that matches the first pattern.
- If the first pattern is a path-prefix pattern, only exact patterns matched by the first pattern and path-prefix patterns matched by, but different from, the first pattern may occur in the URLPatternList.
- If the first pattern is an extension pattern, only exact patterns that are matched by the first pattern and path-prefix patterns may occur in the URLPatternList.
- If the first pattern is the default pattern, "/", any pattern except the default pattern may occur in the URLPatternList.
- If the first pattern is an exact pattern a URLPatternList must not be present in the URLPatternSpec.

The actions parameter contains a comma separated list of HTTP methods that may be followed by a transportType separated from the HTTP method by a colon.

\[
\text{ExtensionMethod} ::= \text{any token as defined by RFC 2616} \\
\quad \text{(that is, } 1^*\text{[any CHAR except CTLs or separator}}
\]
HTTPMethod ::= "Get" | "POST" | "PUT" | "DELETE" | "HEAD" | "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethod

HTTPMethodExceptionList ::= exclamationPoint HTTPMethodList

HTTPMethodSpec ::= emptyString | HTTPMethodExceptionList | HTTPMethodList

transportType ::= "INTEGRAL" | "CONFIDENTIAL" | "NONE"

actions ::= null | HTTPMethodSpec | HTTPMethodSpec colon transportType

If duplicates occur in the HTTPMethodSpec they must be eliminated by the permission constructor.

An empty string HTTPMethodSpec is a shorthand for a List containing all the possible HTTP methods.

If the HTTPMethodSpec contains an HTTPMethodExceptionList (i.e., it begins with an exclamationPoint), the permission pertains to all methods except those occurring in the exception list.

An actions string without a transportType is a shorthand for a actions string with the value "NONE" as its TransportType.

A granted permission representing a transportType of "NONE", indicates that the associated resources may be accessed using any conection type.

Parameters:
name - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor.
actions - identifies the HTTP methods and transport type to which the permission pertains. If the value passed through this parameter is null or the empty string, then the permission is constructed with actions corresponding to all the possible HTTP methods and transportType "NONE".

```
public WebUserDataPermission(String urlPatternSpec, String[] HTTPMethods, String transportType)
WebUserDataPermission URLPatternSpec HTTP
```

```
urlPatternSpec
Web URLPatternSpec
URLPattern Spec
URLPatternnull
URLPatternSpec
URLPattern ("/")
```

```
HTTPMethods
HTTP null HTTP
```

```
transportType
transportType String null transportType
"NONE"
```

**WebUserDataPermission**

```
public WebUserDataPermission(String urlPatternSpec,
String[] HTTPMethods,
String transportType)
```

Creates a new WebUserDataPermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods and the transport type.

**Parameters:**

- **urlPatternSpec** - the URLPatternSpec that identifies the application specific web resources to which the permission pertains. All URLPatterns in the URLPatternSpec are relative to the context path of the deployed web application module, and the same URLPattern must not occur more than once in a
URLPatternSpec. A null URLPatternSpec is translated to the default URLPattern, "/", by the permission constructor.

HTTPMethods - an array of strings each element of which contains the value of an HTTP method. If the value passed through this parameter is null or is an array with no elements, then the permission is constructed with actions corresponding to all the possible HTTP methods.

transportType - a String whose value is a transportType. If the value passed through this parameter is null, then the permission is constructed with actions corresponding to transportType "NONE".

```java
public WebUserDataPermission(HttpServletRequest request) {
    HttpServletRequest request = (HttpServletRequest) request;
    Servlet contextPath = request.getContextPath();
    String requestURI = request.getRequestURI();
    String[] HTTPMethods = request.getMethod().split(\s+);
    String transportType = request.isSecure() ? "SECURED" : "NONE";
    // Create WebUserDataPermission
}
```

WebUserDataPermission

```java
public WebUserDataPermission(HttpServletRequest request) {
    // Create WebUserDataPermission
}
```

Creates a new WebUserDataPermission from the HttpServletRequest object.

Parameters:

request - the HttpServletRequest object corresponding to the Servlet operation to which the permission pertains. The permission name is the substring of the requestURI (HttpServletRequest.getRequestURI()) that begins after the
contextPath (HttpServletRequest.getContextPath()). When the substring operation yields the string "/", the permission is constructed with the empty string as its name. The HTTP method component of the permission's actions is as obtained from HttpServletRequest.getMethod(). The TransportType component of the permission's actions is determined by calling HttpServletRequest.isSecure().

**Method Detail**

```java
class WebUserDataPermission

public boolean equals(Object o)

WebUserDataPermission
WebUserDataPermission URLPatternSpec
URLPatternSpec URLPatternSpec
URLPatternList URLPatternList

P1.implies(P2) && P2.implies(P1) Permission P1
P2

    o  WebUserDataPermission WebUserDataPermission

    return WebUserDataPermission WebUserDataPermission
    true

equals

class WebUserDataPermission

public boolean equals(Object o)

Checks two WebUserDataPermission objects for equality. WebUserDataPermission objects are equivalent if their URLPatternSpec and (canonicalized) actions values are equivalent. The URLPatternSpec of a reference permission is equivalent to that of an argument permission if their first patterns are equivalent, and the patterns of the URLPatternList of the reference permission collectively match exactly the same set of patterns as are matched
by the patterns of the URLPatternList of the argument permission.

Two Permission objects, P1 and P2, are equivalent if and only if 
P1.implies(P2) && P2.implies(P1).

**Specified by:**
equals in class Permission

**Parameters:**
- the WebUserDataPermission object being tested for equality with this WebUserDataPermission.

**Returns:**
true if the argument WebUserDataPermission object is 
equivalent to this WebUserDataPermission.

```java
public String getActions()
WebUserDataPermission
WebUserDataPermission

ExtensionMethod ::= any token as defined by RFC 2616
  1*[ CTL ]

HTTPMethod ::= "GET" | "POST" | "PUT" | "DELETE" | "HEAD" | "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethod

HTTPMethodExceptionList ::= exclamationPoint HTTPMethodList

HTTPMethodSpec ::= emptyString | HTTPMethodExceptionList | HTTPMethodList

transportType ::= "INTEGRAL" | "CONFIDENTIAL" | "NONE"

actions ::= null | HTTPMethodList | HTTPMethodSpec colon transportType
```
getActions

public String getActions()

Returns a canonical String representation of the actions of this WebUserDataPermission. The canonical form of the actions of a WebUserDataPermission is described by the following syntax description.

```
ExtensionMethod ::= any token as defined by RFC 2616
                 (that is, 1*[any CHAR except CTLs or separatec

HTTPMethod ::= "GET" | "POST" | "PUT" | "DELETE" | "HE
             "OPTIONS" | "TRACE" | ExtensionMethod

HTTPMethodList ::= HTTPMethod | HTTPMethodList comma HTTPMethod

HTTPMethodExceptionList ::= exclamationPoint HTTPMethodList

HTTPMethodSpec ::= emptyString | HTTPMethodExceptionList
                 HTTPMethodList

transportType ::= "INTEGRAL" | "CONFIDENTIAL" | "NONE"

actions ::= null | HTTPMethodList |
           HTTPMethodSpec colon transportType
```
If the permission's HTTP methods correspond to the entire HTTP method set and the permission's transport type is "INTEGRAL" or "CONFIDENTIAL", the HTTP methods shall be represented in the canonical form by an emptyString HTTPMethodSpec. If the permission's HTTP methods correspond to the entire HTTP method set, and the permission's transport type is not "INTEGRAL" or "CONFIDENTIAL", the canonical actions value shall be the null value.

If the permission's methods do not correspond to the entire HTTP method set, duplicates must be eliminated and the remaining elements must be ordered such that the predefined methods precede the extension methods, and such that within each method classification the corresponding methods occur in ascending lexical order. The resulting (non-emptyString) HTTPMethodSpec must be included in the canonical form, and if the permission's transport type is not "INTEGRAL" or "CONFIDENTIAL", the canonical actions value must be exactly the resulting HTTPMethodSpec.

**Specified by:**

`getActions` in class `Permission`

**Returns:**

a String containing the canonicalized actions of this WebUserDataPermission (or the null value).

```java
public int hashCode()

WebUserDataPermission

- Java WebUserDataPermission hashCode
  EJBMethode hashCode
- equals WebUserDataPermission
  Permission hashCode

  return
```
**hashCode**

```
public int hashCode()
```

Returns the hash code value for this WebUserDataPermission. The properties of the returned hash code must be as follows:

- During the lifetime of a Java application, the hashCode method shall return the same integer value every time it is called on a WebUserDataPermission object. The value returned by hashCode for a particular EJBMeth permission need not remain consistent from one execution of an application to another.
- If two WebUserDataPermission objects are equal according to the equals method, then calling the hashCode method on each of the two Permission objects must produce the same integer result (within an application).

**Specified by:**

hashCode in class Permission

**Returns:**

the integer hash code value for this object.

---

**public boolean implies(java.security.Permission permission)**

```
WebUserDataPermission "" Permission
```

true

- WebUserDataPermission
- permission URLPattern URLPattern
- permission URLPattern URLPatternSpec
  URLPatternList URLPattern
- permission URLPattern URLPatternSpec
  URLPattern URLPatternSpec URLPatternSpec URLPatternList
URLPattern  permission URLPatternList
URLPattern
URLPattern
permission HTTP HTTP
transportType "NONE" permission
transportType

URLPattern Servlet URL

- String
- path-prefix "/*"
- path-prefix "/*" 2
- extension "."
- "/*"

permission "this" WebUserDataPermission
return true false

implies

public boolean implies(Permission permission)

Determines if the argument Permission is "implied by" this WebUserDataPermission. For this to be the case all of the following must be true:

- The argument is an instance of WebUserDataPermission.
- The first URLPattern in the name of the argument permission is matched by the first URLPattern in the name of this permission.
- The first URLPattern in the name of the argument permission is NOT matched by any URLPattern in the URLPatternList of the URLPatternSpec of this permission.
- If the first URLPattern in the name of the argument permission matches the first URLPattern in the URLPatternSpec of this
permission, then every URLPattern in the URLPatternList of the
URLPatternSpec of this permission is matched by a URLPattern
in the URLPatternList of the argument permission.

- The HTTP methods represented by the actions of the argument
  permission are a subset of the HTTP methods represented by
  the actions of this permission.
- The transportType in the actions of this permission either
  corresponds to the value "NONE", or equals the transportType
  in the actions of the argument permission.

URLPattern matching is performed using theServlet matching rules
where two URL patterns match if they are related as follows:

- their pattern values are String equivalent, or
- this pattern is the path-prefix pattern "/*", or
- this pattern is a path-prefix pattern (that is, it starts with "/" and
  ends with "/*") and the argument pattern starts with the substring
  of this pattern, minus its last 2 characters, and the next
  character of the argument pattern, if there is one, is "/", or
- this pattern is an extension pattern (that is, it starts with "*."
  and the argument pattern ends with this pattern, or
- the reference pattern is the special default pattern, "/", which
  matches all argument patterns.

All of the comparisons described above are case sensitive.

**Specified by:**
implies in class Permission

**Parameters:**
permission - "this" WebUserDataPermission is checked to see if
it implies the argument permission.

**Returns:**
true if the specified permission is implied by this object, false if
not.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.resource.spi.work Interface Work

All Superinterfaces:
   Runnable

public interface Work
extends Runnable
Implements: Runnable

    Work         WorkManager
    version      1.0

This models a Work instance that would be executed by a WorkManager upon submission.

Version:
   1.0
Author:
   Ram Jeyaraman

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void</td>
<td>release()</td>
<td>The WorkManager might call this method to hint the active Work instance to complete execution as soon as possible.</td>
</tr>
</tbody>
</table>

Methods inherited from interface java.langRunnable

<table>
<thead>
<tr>
<th>Method</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>run</td>
<td></td>
</tr>
</tbody>
</table>

Method Detail
public void release()

release

void release()

The WorkManager might call this method to hint the active Work instance to complete execution as soon as possible. This would be called on a separate thread other than the one currently executing the Work instance.
public class WorkAdapter

extends Object
implements WorkListener

Implements: WorkListener

Version: 1.0
Author: Ram Jeyaraman

This class is provided as a convenience for easily creating WorkListener instances by extending this class and overriding only those methods of interest.

Constructor Summary

workAccepted(WorkEvent e)

Method Summary

workAccepted(WorkEvent e)
void Invoked when a work instance has been accepted.

void workCompleted(WorkEvent e)
Invoked when a work instance has completed execution.

void workRejected(WorkEvent e)
Invoked when a work instance has been rejected.

void workStarted(WorkEvent e)
Invoked when a work instance has started execution.

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail

public WorkAdapter()

WorkAdapter

public WorkAdapter()

Method Detail

public void workAccepted(WorkEvent e)
  Work

workAccepted

public void workAccepted(WorkEvent e)
  Invoked when a work instance has been accepted.
public void workRejected(WorkEvent e)

Invoked when a Work instance has been rejected.

Specified by:
workRejected in interface WorkListener

public void workStarted(WorkEvent e)

Invoked when a Work instance has started execution. This only means that a thread has been allocated.

Specified by:
workStarted in interface WorkListener

public void workCompleted(WorkEvent e)

workCompleted
public void workCompleted(WorkEvent e)

Invoked when a Work instance has completed execution.

**Specified by:**
workCompleted in interface WorkListener
javax.resource.spi.work Class WorkCompletedException

java.lang.Object  
   ▼ java.lang.Throwable  
      ▼ java.lang.Exception  
         ▼ javax.resource.ResourceException  
            ▼ javax.resource.spi.work.WorkException  
               ▼ javax.resource.spi.work.WorkCompletedException

All Implemented Interfaces:
   Serializable

public class WorkCompletedException
   extends WorkException

Extends: Throwable > Exception > ResourceException > WorkException

WorkManager Work
Work Work
Work.run()
Work.run()

WorkException.TX_RECREATE_FAILED
WorkException.TX_CONCURRENT_WORK_DISALLOWED
WorkException.UNDEFINED

version 1.0

This exception is thrown by a WorkManager to indicate that a submitted Work instance has completed with an exception.

This could be thrown only after the execution of a Work instance has started (that is, after a thread has been allocated for Work execution). The allocated thread sets up an execution context (if it has been specified), and then calls Work.run().
Any exception thrown during execution context setup or during `Work` execution (that is, during `Work.run()`) is chained within this exception.

An associated error code indicates the nature of the error condition. Possible error codes are `WorkException.TX_RECREATE_FAILED`, `WorkException.TX_CONCURRENT_WORK_DISALLOWED` or `WorkException.UNDEFINED`.

**Version:**
1.0

**Author:**
Ram Jeyaraman

**See Also:**
Serialized Form

---

**Field Summary**

<table>
<thead>
<tr>
<th>Fields inherited from class javax.resource.spi.work.<code>WorkException</code></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>INTERNAL</code>, <code>START_TIMED_OUT</code>, <code>TX_CONCURRENT_WORK_DISALLOWED</code>, <code>TX_RECREATE_FAILED</code>, <code>UNDEFINED</code></td>
</tr>
</tbody>
</table>

**Constructor Summary**

<table>
<thead>
<tr>
<th><code>WorkCompletedException()</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new instance with null as its detail message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><code>WorkCompletedException(String message)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new instance with the specified detail message.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><code>WorkCompletedException(String message, String errorCode)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new throwable with the specified detail message and an error code.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><code>WorkCompletedException(String message, Throwable cause)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new throwable with the specified detail message and cause.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><code>WorkCompletedException(Throwable cause)</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Constructs a new throwable with the specified cause.</td>
</tr>
</tbody>
</table>
### Method Summary

**Methods inherited from class javax.resource.ResourceException**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Throwable**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, setStackTrace, toString</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait</td>
</tr>
</tbody>
</table>

### Constructor Detail

**public WorkCompletedException()**

null

**WorkCompletedException**

**public WorkCompletedException()**

Constructs a new instance with null as its detail message.

**public WorkCompletedException(String message)**

_**message**_
WorkCompletedException

public WorkCompletedException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.

public WorkCompletedException(Throwables cause)

cause throwable

cause Throwable

WorkCompletedException

public WorkCompletedException(Throwables cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

public WorkCompletedException(String message, Throwable cause)

cause throwable

message

cause Throwable

WorkCompletedException

public WorkCompletedException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and
cause.

**Parameters:**
- `message` - the detail message.
- `cause` - a chained exception of type Throwable.

```java
public WorkCompletedException(String message, String errorCode)
```

`throwable`

```java
message
errorCode
```

## WorkCompletedException

**public WorkCompletedException(String message, String errorCode)**

Constructs a new throwable with the specified detail message and an error code.

**Parameters:**
- `message` - a description of the exception.
- `errorCode` - a string specifying the vendor specific error code.
javax.resource.spi.work  Class WorkEvent

java.lang.Object  
   java.util.EventObject  
      javax.resource.spi.work.WorkEvent

All Implemented Interfaces:
   Serializable

public class WorkEvent
extends EventObject

Extends: java.util.EventObject

Work

version 1.0

This class models the various events that occur during the processing of a Work instance.

Version:
   1.0

Author:
   Ram Jeyaraman

See Also:
   Serialized Form

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
</table>
| static int WORK_ACCEPTED  
   Indicates Work instance has been accepted. |
| static int WORK_COMPLETED  
   Indicates Work instance has completed execution. |
<p>| WORK_REJECTED |</p>
<table>
<thead>
<tr>
<th>static int</th>
<th>Indicates work instance has been rejected.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static int</td>
<td>WORK_STARTED Indicates work instance has started execution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Fields inherited from class java.util.EventObject</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
</tr>
</tbody>
</table>

## Constructor Summary

### WorkEvent (Object source, int type, Work work, WorkException exc)
Constructor.

### WorkEvent (Object source, int type, Work work, WorkException exc, long startDuration)
Constructor.

## Method Summary

### WorkException

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getException()</td>
<td>Return the WorkException.</td>
</tr>
</tbody>
</table>

### long

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getStartDuration()</td>
<td>Return the start interval duration.</td>
</tr>
</tbody>
</table>

### int

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getType()</td>
<td>Return the type of this event.</td>
</tr>
</tbody>
</table>

### Work

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>getWork()</td>
<td>Return the Work instance which is the cause of the event.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.util.EventObject

getSource, toString

## Methods inherited from class java.lang.Object

clon, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait
WORK_ACCEPTED
public static final int WORK_ACCEPTED

Indicates Work instance has been accepted.

See Also:
Constant Field Values

WORK_REJECTED
public static final int WORK_REJECTED

Indicates Work instance has been rejected.

See Also:
Constant Field Values

WORK_STARTED
public static final int WORK_STARTED

Indicates Work instance has started execution.

See Also:
Constant Field Values
public static final int WORK_COMPLETED

Indicates work instance has completed execution.

See Also:
Constant Field Values

Constructor Detail

public WorkEvent(Object source, int type, Work work, WorkException exc)

source  
  type  
  work  Work
  exc  Work

WorkEvent

public WorkEvent(Object source, 
  int type, 
  Work work, 
  WorkException exc)

Constructor.

Parameters:
    source - The object on which the event initially occurred.
    type - The event type.
    work - The Work object on which the event occurred.
    exc - The exception that occurred during Work processing.

public WorkEvent(Object source, int type, Work work, 
  WorkException exc, long startDuration)
WorkEvent

public WorkEvent(Object source, int type, Work work, WorkException exc, long startDuration)

Constructor.

Parameters:
- source - The object on which the event initially occurred.
- type - The event type.
- work - The Work object on which the event occurred.
- exc - The exception that occurred during Work processing.
- startDuration - The start delay duration (in milliseconds).

Method Detail

public int getType()

    return

getType

public int getType()

    Return the type of this event.
Returns:
the event type.

public Work getWork()
Work work
return work

getWork

public Work getWork()

Return the work instance which is the cause of the event.

Returns:
the work instance.

public long getStartDuration()
return Work work -1

getStartDuration

public long getStartDuration()

Return the start interval duration.

Returns:
the time elapsed (in milliseconds) since the work was accepted, until the work execution started. Note, this does not offer real-time guarantees. It is valid to return -1, if the actual start interval duration is unknown.

public WorkException getException()
public WorkException getException()

Return the WorkException. The actual WorkException subtype returned depends on the type of the event.

Returns:
  a WorkRejectedException or a WorkCompletedException, if any.
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td>DETAIL</td>
<td>FIELD</td>
</tr>
</tbody>
</table>
javax.resource.spi.work Class WorkException

java.lang.Object
   ▼ java.lang.Throwable
      ▼ java.lang.Exception
         ▼ javax.resource.ResourceException
            ▼ javax.resource.spi.work.WorkException

All Implemented Interfaces:
   Serializable

Direct Known Subclasses:
   WorkCompletedException, WorkRejectedException

public class WorkException
   extends ResourceException

Extends: Throwable > Exception > ResourceException
Extended by: WorkCompletedException, WorkRejectedException

   Work

   version 1.0

A common base class for all work processing related exceptions.

Version:
   1.0
Author:
   Ram Jeyaraman
See Also:
   Serialized Form

Field Summary

   INTERNAL
| static String | Indicates an internal error condition. |
| static String | **START_TIMED_OUT** Indicates start timeout expiration. |
| static String | **TX_CONCURRENT_WORK_DISALLOWED** Indicates that concurrent work within a transaction is disallowed. |
| static String | **TX_RECREATE_FAILED** Indicates a failure in recreating the specified transaction context. |
| static String | **UNDEFINED** Undefined error code. |

### Constructor Summary

**WorkException()**

Constructs a new instance with null as its detail message.

**WorkException(String message)**

Constructs a new instance with the specified detail message.

**WorkException(String message, String errorCode)**

Constructs a new throwable with the specified detail message and an error code.

**WorkException(String message, Throwable cause)**

Constructs a new throwable with the specified detail message and cause.

**WorkException(Throwable cause)**

Constructs a new throwable with the specified cause.

### Method Summary

Methods inherited from class `javax.resource.ResourceException`

getErrorCode, getLinkedException, getMessage, setErrorCode, setLinkedException

Methods inherited from class `java.lang.Throwable`
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString

Methods inherited from class java.lang.Object
clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

Field Detail

INTERNAL

public static final String INTERNAL

Indicates an internal error condition.

See Also:
Constant Field Values

UNDEFINED

public static final String UNDEFINED

Undefined error code.

See Also:
Constant Field Values

START_TIMED_OUT
public static final String START_TIMED_OUT

Indicates start timeout expiration.

See Also:
Constant Field Values

---

TX_CONCURRENT_WORK_DISALLOWED

public static final String TX_CONCURRENT_WORK_DISALLOWED

Indicates that concurrent work within a transaction is disallowed. That is, there is already another Work instance associated with the specified transaction context.

See Also:
Constant Field Values

---

TX_RECREATE_FAILED

public static final String TX_RECREATE_FAILED

Indicates a failure in recreating the specified transaction context.

See Also:
Constant Field Values

---

**Constructor Detail**

public WorkException()

null
**WorkException**

```java
public WorkException()

    Constructs a new instance with null as its detail message.
```

```java
public WorkException(String message)

    message
```

**WorkException**

```java
public WorkException(String message)

    Constructs a new instance with the specified detail message.

    Parameters:
        message - the detail message.
```

```java
public WorkException(Throwable cause)

    cause throwable
```

**WorkException**

```java
public WorkException(Throwable cause)

    Constructs a new throwable with the specified cause.

    Parameters:
        cause - a chained exception of type Throwable.
```

```java
public WorkException(String message, Throwable cause)
```
public WorkException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.

public WorkException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi.work Interface WorkListener

All Superinterfaces:
   EventListener

All Known Implementing Classes:
   WorkAdapter

public interface WorkListener
   extends EventListener

Implements: java.util.EventListener
Implemented by: WorkAdapter

<table>
<thead>
<tr>
<th>WorkListener</th>
<th>Work</th>
<th>WorkManager</th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkListener</td>
<td></td>
<td>version 1.0</td>
</tr>
</tbody>
</table>

This models a WorkListener instance which would be notified by the WorkManager when the various Work processing events (work accepted, work rejected, work started, work completed) occur. The WorkListener instance must not make any thread assumptions and must be thread-safe ie., a notification could occur from any arbitrary thread. Further, it must not make any assumptions on the ordering of notifications.

Version:
   1.0

Author:
   Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void workAccepted(WorkEvent e)</td>
</tr>
<tr>
<td>Invoked when a work instance has been accepted.</td>
</tr>
<tr>
<td>Method</td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td>void <code>workCompleted</code></td>
</tr>
<tr>
<td>void <code>workRejected</code></td>
</tr>
<tr>
<td>void <code>workStarted</code></td>
</tr>
</tbody>
</table>

### Method Detail

**public void workAccepted(WorkEvent e)**

Work

**workAccepted**

```java
void workAccepted(WorkEvent e)
```

Invoked when a work instance has been accepted.

**public void workRejected(WorkEvent e)**

Work

**workRejected**

```java
void workRejected(WorkEvent e)
```

Invoked when a work instance has been rejected.

**public void workStarted(WorkEvent e)**

Work
workStarted

void workStarted(WorkEvent e)

Invoked when a Work instance has started execution. This only means that a thread has been allocated.

workCompleted

public void workCompleted(WorkEvent e)

Invoked when a Work instance has completed execution.
public interface WorkManager

WorkManager Work Java

* Work Work WorkRejectedException
* Work Work WorkRejectedException
  WorkWorkRejectedException
* Work Work Work Work WorkRejectedException
  Work.run()
  Work.run()
* Work WorkManager Work
  WorkCompletedException

version 1.0

This interface models a WorkManager which provides a facility to submit Work instances for execution. This frees the user from having to create Java threads directly to do work. Further, this allows efficient pooling of thread resources and more control over thread usage. The various stages in Work processing are:

- **work submit**: A Work instance is being submitted for execution. The Work instance could either be accepted or rejected with a WorkRejectedException set to an appropriate error code.
- **work accepted**: The submitted Work instance has been accepted. The accepted Work instance could either start execution or could be rejected again with a WorkRejectedException set to an appropriate error code. There is no guarantee on when the execution would start unless a start timeout duration is specified. When a start timeout is specified, the Work execution must be started within the specified duration (not a real-time guarantee), failing which a
WorkRejectedException set to an error code (WorkRejected.TIMED_OUT) is thrown.

- **work rejected**: The Work instance has been rejected. The Work instance could be rejected during Work submittal or after the Work instance has been accepted (but before Work instance starts execution). The rejection could be due to internal factors or start timeout expiration. A WorkRejectedException with an appropriate error code (indicates the reason) is thrown in both cases.

- **work started**: The execution of the Work instance has started. This means that a thread has been allocated for its execution. But this does not guarantee that the allocated thread has been scheduled to run on a CPU resource. Once execution is started, the allocated thread sets up an appropriate execution context (transaction, security, etc) and calls Work.run(). Note, any exception thrown during execution context setup or Work.run() leads to completion of processing.

- **work completed**: The execution of the Work has been completed. The execution could complete with or without an exception. The WorkManager catches any exception thrown during Work processing (which includes execution context setup), and wraps it with a WorkCompletedException.

---

Version:
1.0

Author:
Ram Jeyaraman

---

### Field Summary

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static long</td>
<td>IMMEDIATE</td>
<td>A constant to indicate timeout duration.</td>
</tr>
<tr>
<td>static long</td>
<td>INDEFINITE</td>
<td>A constant to indicate timeout duration.</td>
</tr>
<tr>
<td>static long</td>
<td>UNKNOWN</td>
<td>A constant to indicate an unknown start delay duration or other unknown values.</td>
</tr>
</tbody>
</table>
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void doWork(Work work)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
<tr>
<td>void doWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
<tr>
<td>void scheduleWork(Work work)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
<tr>
<td>void scheduleWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
<tr>
<td>long startWork(Work work)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
<tr>
<td>long startWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener)</td>
<td>Accepts a Work instance for processing.</td>
</tr>
</tbody>
</table>

### Field Detail

**IMMEDIATE**

static final long IMMEDIATE

A constant to indicate timeout duration. A zero timeout value indicates an action be performed immediately.

**See Also:**

[Constant Field Values](#)

**INDEFINITE**
static final long INDEFINITE

A constant to indicate timeout duration. A maximum timeout value indicates that an action be performed arbitrarily without any time constraint.

See Also:
Constant Field Values

UNKNOWN

static final long UNKNOWN

A constant to indicate an unknown start delay duration or other unknown values.

See Also:
Constant Field Values

---

Method Detail

public void doWork(Work work) throws WorkException

<table>
<thead>
<tr>
<th>Work</th>
<th>Work</th>
<th>Work</th>
</tr>
</thead>
<tbody>
<tr>
<td>work</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Throws

WorkRejectedException: Work

Throws

WorkCompletedException: Work

doWork

void doWork(Work work)

throws WorkException

Accepts a Work instance for processing. This call blocks until the Work instance completes execution. There is no guarantee on when the
accepted work instance would start execution i.e., there is no time constraint to start execution.

**Parameters:**
- **work** - The unit of work to be done. Could be long or short-lived.

**Throws:**
- **WorkRejectedException** - indicates that a work instance has been rejected from further processing. This can occur due to internal factors.
- **WorkCompletedException** - indicates that a work instance has completed execution with an exception.
- **WorkException**

```java
public void doWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener) throws WorkException
    work
    startTimeout (WorkRejectedException.TIMED_OUT)  WorkRejectedException
    execContext
    workListener 
    Throws  WorkRejectedException:  Work
    Throws  WorkCompletedException:  Work
```

doWork

```java
void doWork(Work work,
            long startTimeout,
            ExecutionContext execContext,
            WorkListener workListener)
    throws WorkException
```

Accepts a work instance for processing. This call blocks until the work instance completes execution.
Parameters:
- **work**: The unit of work to be done. Could be long or short-lived.
- **startTimeTimeout**: A time duration (in milliseconds) within which the execution of the **Work** instance must start. Otherwise, the **Work** instance is rejected with a **WorkRejectedException** set to an appropriate error code (**WorkRejectedException.TIMED_OUT**).
  
  Note, this does not offer real-time guarantees.
- **execContext**: An object containing the execution context with which the submitted **Work** instance must be executed.
- **workListener**: An object which would be notified when the various **Work** processing events (work accepted, work rejected, work started, work completed) occur.

Throws:
- **WorkRejectedException** - indicates that a **Work** instance has been rejected from further processing. This can occur due to internal factors or start timeout expiration.
- **WorkCompletedException** - indicates that a **Work** instance has completed execution with an exception.
- **WorkException**

```java
public long startWork(Work work) throws WorkException {
    Work work
    return Work -1
    Throws WorkRejectedException:
}
```

### startWork

```java
long startWork(Work work) throws WorkException {
    Accepts a **Work** instance for processing. This call blocks until the **Work** instance starts execution but not until its completion. There is no guarantee on when the accepted **Work** instance would start execution i.e., there is no time constraint to start execution.

    Parameters:
```
work - The unit of work to be done. Could be long or short-lived.

Returns:
the time elapsed (in milliseconds) from work acceptance until start of execution. Note, this does not offer real-time guarantees. It is valid to return -1, if the actual start delay duration is unknown.

Throws:
WorkRejectedException - indicates that a work instance has been rejected from further processing. This can occur due to internal factors.
WorkException

```java
public long startWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener) throws WorkException {
    Work startWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener)
    
    return Work -1
    
    Throws WorkRejectedException:

    startWork

    long startWork(Work work,
    long startTimeout,
    ExecutionContext execContext,
    WorkListener workListener)
    throws WorkException
```

Accepts a Work instance for processing. This call blocks until the Work instance starts execution but not until its completion. There is no guarantee on when the accepted Work instance would start execution i.e., there is no time constraint to start execution.
Parameters:

- **work** - The unit of work to be done. Could be long or short-lived.
- **startTimeout** - a time duration (in milliseconds) within which the execution of the **Work** instance must start. Otherwise, the **Work** instance is rejected with a **WorkRejectedException** set to an appropriate error code (**WorkRejectedException.TIMED_OUT**).
  
  Note, this does not offer real-time guarantees.
- **execContext** - an object containing the execution context with which the submitted **Work** instance must be executed.
- **workListener** - an object which would be notified when the various **Work** processing events (work accepted, work rejected, work started, work completed) occur.

Returns:

- the time elapsed (in milliseconds) from **Work** acceptance until start of execution. Note, this does not offer real-time guarantees. It is valid to return -1, if the actual start delay duration is unknown.

Throws:

- **WorkRejectedException** - indicates that a **Work** instance has been rejected from further processing. This can occur due to internal factors or start timeout expiration.
- **WorkException**

---

```java
public void scheduleWork(Work work) throws WorkException
```

**Throws**  
**WorkRejectedException**

```java
void scheduleWork(Work work)
```

**Throws**  
**WorkException**

Accepts a **Work** instance for processing. This call does not block and returns immediately once a **Work** instance has been accepted for
There is no guarantee on when the submitted Work instance would start execution i.e., there is no time constraint to start execution.

**Parameters:**
- work - The unit of work to be done. Could be long or short-lived.

**Throws:**
- WorkRejectedException - indicates that a Work instance has been rejected from further processing. This can occur due to internal factors.
- WorkException

```
public void scheduleWork(Work work, long startTimeout, ExecutionContext execContext,WorkListener workListener) throws WorkException
```

**scheduleWork**

```
void scheduleWork(Work work, long startTimeout, ExecutionContext execContext, WorkListener workListener)
throws WorkException
```

Accepts a Work instance for processing. This call does not block and returns immediately once a Work instance has been accepted for processing.

**Parameters:**
work - The unit of work to be done. Could be long or short-lived.

startTime - a time duration (in milliseconds) within which the execution of the Work instance must start. Otherwise, the Work instance is rejected with a WorkRejectedException set to an appropriate error code (WorkRejectedException.TIMED_OUT). Note, this does not offer real-time guarantees.

execContext - an object containing the execution context with which the submitted Work instance must be executed.

workListener - an object which would be notified when the various Work processing events (work accepted, work rejected, work started, work completed) occur.

Throws:

WorkRejectedException - indicates that a Work instance has been rejected from further processing. This can occur due to internal factors.

WorkException

PREV CLASS NEXT CLASS
SUMMARY: NESTED | FIELD | CONSTR | METHOD
FRAMES NO FRAMES
DETAIL: FIELD | CONSTR | METHOD

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.resource.spi.work  Class WorkRejectedException

java.lang.Object
   ↓ java.lang.Throwable
      ↓ java.lang.Exception
         ↓ javax.resource.ResourceException
            ↓ javax.resource.spi.work.WorkException
                ↓ javax.resource.spi.work.WorkRejectedException

All Implemented Interfaces:
   Serializable

public class WorkRejectedException
   extends WorkException

Extends: Throwable > Exception > ResourceException > WorkException

WorkManager  Work

Work  Work

WorkException.START_TIMED_OUT  WorkException.INTERNAL
WorkException.UNDEFINED

version  1.0

This exception is thrown by a WorkManager to indicate that a submitted Work instance has been rejected. The rejection could be due to internal factors or start timeout expiration.

This could be thrown only before the execution of a Work instance starts (that is, before a thread has been allocated for Work execution).

An associated error code indicates the nature of the error condition. Possible error codes are WorkException.START_TIMED_OUT, WorkException.INTERNAL OR WorkException.UNDEFINED.
Field Summary

Fields inherited from class javax.resource.spi.work.WorkException
INTERNAL, START_TIMED_OUT, TX_CONCURRENT_WORK_DISALLOWED,
TX_RECREATE_FAILED, UNDEFINED

Constructor Summary

WorkRejectedException()
Constructs a new instance with null as its detail message.

WorkRejectedException(String message)
Constructs a new instance with the specified detail message.

WorkRejectedException(String message, String errorCode)
Constructs a new throwable with the specified detail message
and an error code.

WorkRejectedException(String message, Throwable cause)
Constructs a new throwable with the specified detail message
and cause.

WorkRejectedException(Throwable cause)
Constructs a new throwable with the specified cause.

Method Summary

Methods inherited from class javax.resource.ResourceException
getErrorCode, getLinkedException, getMessage, setErrorCode,
setLinkedException
Methods inherited from class java.lang.Throwable

- fillInStackTrace
- getCause
- getLocalizedMessage
- getStackTrace
- initCause
- printStackTrace
- printStackTrace
- printStackTrace
- setStackTrace
- toString

Methods inherited from class java.lang.Object

- clone
- equals
- finalize
- getClass
- hashCode
- notify
- notifyAll
- wait
- wait
- wait

Constructor Detail

public WorkRejectedException()
null

WorkRejectedException

public WorkRejectedException()

Constructs a new instance with null as its detail message.

public WorkRejectedException(String message)

message

WorkRejectedException

public WorkRejectedException(String message)

Constructs a new instance with the specified detail message.

Parameters:
message - the detail message.
public WorkRejectedException(Throwable cause)

Constructs a new throwable with the specified cause.

Parameters:
cause - a chained exception of type Throwable.

public WorkRejectedException(String message, Throwable cause)

Constructs a new throwable with the specified detail message and cause.

Parameters:
message - the detail message.
cause - a chained exception of type Throwable.
WorkRejectedException

public WorkRejectedException(String message, String errorCode)

Constructs a new throwable with the specified detail message and an error code.

Parameters:
message - a description of the exception.
errorCode - a string specifying the vendor specific error code.
javax.jms Interface XAConnection

All Superinterfaces:  
  Connection

All Known Subinterfaces:  
  XAQueueConnection, XATopicConnection

The XAConnection interface extends the capability of Connection by providing an XASession (optional).

The XAConnection interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:  
  1.1 February 2, 2002

Author:  
  Mark Hapner, Rich Burridge, Kate Stout
See Also:
XAQueueConnection, XATopicConnection

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session createSession</td>
<td>Creates a Session object.</td>
</tr>
<tr>
<td>XASession createXASession</td>
<td>Creates an XASession object.</td>
</tr>
</tbody>
</table>

## Methods inherited from interface javax.jms.Connection

- close, createConnectionConsumer, createDurableConnectionConsumer, getClientID, getExceptionListener, getMetaData, setClientID, setExceptionListener, start, stop

## Method Detail

```java
public XASession createXASession() throws JMSException
```

XASession

```
return XASession
```

Throws: JMSException: XAConnection XASession

since

1.1

createXASession

```
XASession createXASession() throws JMSException
```

Creates an XASession object.

**Returns:**
- a newly created XASession

**Throws:**
- JMSException - if the XAConnection object fails to create an
XASession due to some internal error.

Since:
   1.1

---

public Session createSession(boolean transacted, int acknowledgeMode) throws JMSException

Session
    transacted
    acknowledgeMode

return Session

Throws JMSException: XAConnection Session
since 1.1

createSession

Session createSession(boolean transacted, int acknowledgeMode) throws JMSException

Creates an Session object.

Specified by:
   createSession in interface Connection

Parameters:
    transacted - usage undefined
    acknowledgeMode - usage undefined

Returns:
    a Session object

Throws:
    JMSException - if the XAConnection object fails to create an Session due to some internal error.

Since:
    1.1

See Also:
   Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE,
   Session.DUPS_OK_ACKNOWLEDGE
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.jms  **Interface XAConnectionFactory**

**All Known Subinterfaces:**

`XAQueueConnectionFactory`, `XATopicConnectionFactory`

```java
public interface XAConnectionFactory
```

**Implemented by:** `XAQueueConnectionFactory`, `XATopicConnectionFactory`

The `XAConnectionFactory` interface is a base interface for the `XAQueueConnectionFactory` and `XATopicConnectionFactory` interfaces.

Some application servers provide support for grouping JTS capable resource use into a distributed transaction (optional). To include JMS API transactions in a JTS transaction, an application server requires a JTS aware JMS provider. A JMS provider exposes its JTS support using an `XAConnectionFactory` object, which an application server uses to create `XAConnection` objects.

`XAConnectionFactory` objects are JMS administered objects, just like
ConnectionFactory objects. It is expected that application servers will find them using the Java Naming and Directory Interface (JNDI) API.

The XAConnectionFactory interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:
   1.1 April 4, 2002
Author:
   Mark Hapner, Rich Burridge, Kate Stout
See Also:
   XAQueueConnectionFactory, XATopicConnectionFactory

### Method Summary

<table>
<thead>
<tr>
<th>XAConnection</th>
<th>createXAConnection()</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Creates an XAConnection with the default user identity.</td>
</tr>
<tr>
<td>XAConnection</td>
<td>createXAConnection(String userName, String password)</td>
</tr>
<tr>
<td></td>
<td>Creates an XA connection with the specified user identity.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public XAConnection createXAConnection() throws JMSException

   return XAConnection

   Throws JMSException: JMS XA
   Throws JMSSecurityException: since 1.1
```
createXAConnection

**XAC**
createXAConnection() throws JMSException

Creates an XAConnection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

**Returns:**
- a newly created XAConnection

**Throws:**
- JMSException - if the JMS provider fails to create an XA connection due to some internal error.
- JMSSecurityException - if client authentication fails due to an invalid user name or password.

**Since:**
1.1

---

public XAC createXAConnection(String userName, String password) throws JMSException

**userName**

**password**

**return**

**XAC**

**Throws**
- JMSException: JMS XA
- JMSSecurityException: since 1.1

createXAConnection

**XAC** createXAConnection(String userName, String password) throws JMSException

Creates an XA connection with the specified user identity. The connection is created in stopped mode. No messages will be
delivered until the `Connection.start` method is explicitly called.

**Parameters:**
- `userName` - the caller's user name
- `password` - the caller's password

**Returns:**
a newly created XA connection

**Throws:**
- `JMSException` - if the JMS provider fails to create an XA connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

**Since:**
1.1
javax.transaction.xa  Class XAException

java.lang.Object  
   ↓ java.lang.Throwable  
      ↓ java.lang.Exception  
         ↓ javax.transaction.xa.XAException

All Implemented Interfaces:
   Serializable

public class XAException

extends Exception

Extends: Throwable > Exception

(RM)  XAException

The XAException is thrown by the Resource Manager (RM) to inform the Transaction Manager of an error encountered by the involved transaction.

See Also:
   Serialized Form

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>int errorCode</td>
</tr>
<tr>
<td>static int XA_HEURCOM</td>
</tr>
<tr>
<td>static int XA_HEURHAZ</td>
</tr>
<tr>
<td>Static Int</td>
</tr>
<tr>
<td>------------</td>
</tr>
<tr>
<td>XA_HEURMIX</td>
</tr>
<tr>
<td>XA_HEURRB</td>
</tr>
<tr>
<td>XA_NOMIGRATE</td>
</tr>
<tr>
<td>XA_RBBASE</td>
</tr>
<tr>
<td>XA_RBCOMMFAIL</td>
</tr>
<tr>
<td>XA_RBDEADLOCK</td>
</tr>
<tr>
<td>XA_RBEND</td>
</tr>
<tr>
<td>XA_RBINTEGRITY</td>
</tr>
<tr>
<td>XA_RBOTHER</td>
</tr>
<tr>
<td>XA_RBPROTO</td>
</tr>
<tr>
<td>XA_RBROLLBACK</td>
</tr>
<tr>
<td>XA_RBTIMEOUT</td>
</tr>
<tr>
<td>XA_RBTRANSIENT</td>
</tr>
<tr>
<td>XA_RDONLY</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
<tr>
<td>static int</td>
</tr>
</tbody>
</table>

**Constructor Summary**

- **XAException()**
  
  Create an XAException.

- **XAException(int errcode)**
  
  Create an XAException with a given error code.

- **XAException(String s)**
  
  Create an XAException with a given string.

**Method Summary**

Methods inherited from class java.lang.Throwable

- fillInStackTrace, getCause, getLocalizedMessage, getMessage,
Field Detail

errorCode

public int errorCode

The error code with which to create the SystemException.

XA_RBBASE

public static final int XA_RBBASE

The inclusive lower bound of the rollback codes.

See Also:
Constant Field Values

XA_RBROLLBACK

public static final int XA_RBROLLBACK

Indicates that the rollback was caused by an unspecified reason.
See Also:  
Constant Field Values

XA_RBCOMMFAIL

public static final int XA_RBCOMMFAIL

Indicates that the rollback was caused by a communication failure.

See Also:  
Constant Field Values

XA_RBDEADLOCK

public static final int XA_RBDEADLOCK

A deadlock was detected.

See Also:  
Constant Field Values

XA_RBINTEGRITY

public static final int XA_RBINTEGRITY

A condition that violates the integrity of the resource was detected.

See Also:  
Constant Field Values
XA_RBOTHER

public static final int XA_RBOTHER

The resource manager rolled back the transaction branch for a reason not on this list.

See Also:
Constant Field Values

XA_RBPROTO

public static final int XA_RBPROTO

A protocol error occurred in the resource manager.

See Also:
Constant Field Values

XA_RBTIMEOUT

public static final int XA_RBTIMEOUT

A transaction branch took too long.

See Also:
Constant Field Values

XA_RBTRANSIENT
public static final int XA_RBTRANSIENT

May retry the transaction branch.

See Also:
Constant Field Values

XA_RBEND

public static final int XA_RBEND

The inclusive upper bound of the rollback error code.

See Also:
Constant Field Values

XA_NOMIGRATE

public static final int XA_NOMIGRATE

Resumption must occur where the suspension occurred.

See Also:
Constant Field Values

XA_HEURHAZ

public static final int XA_HEURHAZ

The transaction branch may have been heuristically completed.
XA_HEURCOM

public static final int XA_HEURCOM

The transaction branch has been heuristically committed.

See Also:
Constant Field Values

XA_HEURRB

public static final int XA_HEURRB

The transaction branch has been heuristically rolled back.

See Also:
Constant Field Values

XA_HEURMIX

public static final int XA_HEURMIX

The transaction branch has been heuristically committed and rolled back.

See Also:
Constant Field Values
XA_RETRY

public static final int XA_RETRY

    Routine returned with no effect and may be reissued.

    See Also:
    Constant Field Values

XA_RDONLY

public static final int XA_RDONLY

    The transaction branch was read-only and has been committed.

    See Also:
    Constant Field Values

XAER_ASYNC

public static final int XAER_ASYNC

    There is an asynchronous operation already outstanding.

    See Also:
    Constant Field Values

XAER_RMERR
public static final int XAER_RMERR

A resource manager error has occurred in the transaction branch.

See Also:
   Constant Field Values

XAER_NOTA

public static final int XAER_NOTA

The XID is not valid.

See Also:
   Constant Field Values

XAER_INVAL

public static final int XAER_INVAL

Invalid arguments were given.

See Also:
   Constant Field Values

XAER_PROTO

public static final int XAER_PROTO

Routine was invoked in an improper context.
See Also:  
Constant Field Values

XAER_RMFFAIL

public static final int XAER_RMFFAIL

Resource manager is unavailable.

See Also:  
Constant Field Values

XAER_DUPID

public static final int XAER_DUPID

The XID already exists.

See Also:  
Constant Field Values

XAER_OUTSIDE

public static final int XAER_OUTSIDE

The resource manager is doing work outside a global transaction.

See Also:  
Constant Field Values
public XAException()
XAException

XAException

public XAException()

Create an XAException.

public XAException(String s)
XAException

s
String

XAException

public XAException(String s)

Create an XAException with a given string.

Parameters:
s - The String object containing the exception message.

public XAException(int errcode)
XAException

errcode

XAException

public XAException(int errcode)
Create an XAException with a given error code.

**Parameters:**

errcode - The error code identifying the exception.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/legal/licenses/cd marc terms.html).

PS:
javax.jms Interface XAQueueConnection

All Superinterfaces:
   Connection, QueueConnection, XAConnection

public interface XAQueueConnection
   extends XAConnection, QueueConnection

Implements: XAConnection, QueueConnection

XAQueueConnection    QueueConnection    XAConnection

XAQueueConnection    JMS    JMS    XA

version            1.1 February 2 - 2002
See also            javax.jms.XAConnection

An XAQueueConnection provides the same create options as
QueueConnection (optional). The only difference is that an XAConnection is
by definition transacted.

The XAQueueConnection interface is optional. JMS providers are not
required to support this interface. This interface is for use by JMS
providers to support transactional environments. Client programs are
strongly encouraged to use the transactional support available in their
environment, rather than use these XA interfaces directly.

Version:
   1.1 February 2 - 2002
Author:
   Mark Hapner, Rich Burridge, Kate Stout
See Also:
   XAConnection
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueSession.createQueueSession(boolean transacted, int acknowledgeMode)</td>
<td>Creates an XAQueueSession object.</td>
</tr>
<tr>
<td>XAQueueSession.createXAQueueSession()</td>
<td>Creates an XAQueueSession object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.XAConnection
createSession, createXASession

Methods inherited from interface javax.jms.Connection
createConnectionConsumer, createDurableConnectionConsumer, getClientID, getExceptionListener, getMetaData, setClientID, setExceptionListener, start, stop

Methods inherited from interface javax.jms.QueueConnection
createConnectionConsumer

Method Detail

```
public XAQueueSession createXAQueueSession() throws JMSException

    XAQueueSession
    return XAQueueSession

    Throws JMSException: XAQueueConnection XA
```

createXAQueueSession
**createXAQueueSession()**

 Throws: *JMSException*

 Creates an XAQueueSession object.

 **Returns:**
 a newly created XAQueueSession

 **Throws:**
 *JMSException* - if the XAQueueConnection object fails to create an XA queue session due to some internal error.

---

**createQueueSession**

 **public QueueSession createQueueSession(boolean transacted, int acknowledgeMode)**

 Throws: *JMSException*

 Creates an XAQueueSession object.

 **Specified by:**
 *createQueueSession* in interface *QueueConnection*

 **Parameters:**
 - transacted - usage undefined
 - acknowledgeMode - usage undefined

 **Returns:**
 a newly created XAQueueSession

 **Throws:**
 *JMSException* - if the XAQueueConnection object fails to create an XA queue session due to some internal error.
See Also:

Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, Session.DUPS_OK_ACKNOWLEDGE

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface XAQueueConnectionFactory

All Superinterfaces: 
  ConnectionFactory, QueueConnectionFactory, XAConnectionFactory

public interface XAQueueConnectionFactory
  extends XAConnectionFactory, QueueConnectionFactory

Implements: XAConnectionFactory, QueueConnectionFactory

XAQueueConnectionFactory  QueueConnectionFactory
XATopicConnectionFactory  JMS   JMS   XA

version  1.1 - 8 April 2002
See  javax.jms.QueueConnectionFactory,
also  javax.jms.XAConnectionFactory

An XAQueueConnectionFactory provides the same create options as a QueueConnectionFactory (optional).

The XATopicConnectionFactory interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:  1.1 - 8 April 2002
Author:  Mark Hapner, Rich Burridge, Kate Stout
See Also:  QueueConnectionFactory, XAConnectionFactory
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createXAQueueConnection()</code></td>
<td>Creates an XA queue connection with the default user identity.</td>
</tr>
<tr>
<td><code>createXAQueueConnection(String userName, String password)</code></td>
<td>Creates an XA queue connection with the specified user identity.</td>
</tr>
</tbody>
</table>

### Methods inherited from interface `javax.jms.XAConnectionFactory`  
- `createXAConnection`, `createXAConnection`<br>  

### Methods inherited from interface `javax.jms.QueueConnectionFactory`  
- `createQueueConnection`, `createQueueConnection`<br>  

### Methods inherited from interface `javax.jms.ConnectionFactory`  
- `createConnection`, `createConnection`<br>  

### Method Detail

```java
public XAQueueConnection createXAQueueConnection() throws JMSException
    XA
return

throws JMSException: JMS XA
Throws JMSException: JMSException:

createXAQueueConnection

XAQueueConnection createXAQueueConnection() throws JMSException
```
Creates an XA queue connection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

**Returns:**
- a newly created XA queue connection

**Throws:**
- `JMSException` - if the JMS provider fails to create an XA queue connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

```java
public XAQueueConnection createXAQueueConnection(String userName, String password) throws JMSException, JMSSecurityException {
    return XAConnection.start(userName, password);
}
```

Creates an XA queue connection with the specified user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

**Parameters:**
- `userName` - the caller's user name
- `password` - the caller's password

**Returns:**
a newly created XA queue connection

**Throws:**

- `JMSException` - if the JMS provider fails to create an XA queue connection due to some internal error.
- `JMSSecurityException` - if client authentication fails due to an invalid user name or password.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface XAQueueSession

All Superinterfaces:
   Runnable, Session, XASession

public interface XAQueueSession
extends XASession

Implements: XASession

XAQueueSession QueueSession QueueReceiverQueueSender QueueBrowser

XAQueueSession JMS JMS XA

version 1.1 - February 2, 2002 en
See also javax.jms.XASession

An XAQueueSession provides a regular QueueSession, which can be used to create QueueReceiver, QueueSender, and QueueBrowser objects (optional).

The XAQueueSession interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:
   1.1 - February 2, 2002

Author:
   Mark Hapner, Rich Burridge, Kate Stout

See Also:
   XASession
# Field Summary

Fields inherited from interface javax.jms.Session

- AUTO_ACKNOWLEDGE
- CLIENT_ACKNOWLEDGE
- DUPS_OK_ACKNOWLEDGE
- SESSION_TRANSACTED

# Method Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueSession</td>
<td>getQueueSession()</td>
<td>Gets the queue session associated with this XAQueueSession.</td>
</tr>
</tbody>
</table>

# Method Detail

```java
public QueueSession getQueueSession() throws JMSException
```

- XAQueueSession
- return
- Throws
  - JMSException: [Exception message]
```
getQueueSession

QueueSession getQueueSession()
throws JMSException

Gets the queue session associated with this XAQueueSession.

**Returns:**
the queue session object

**Throws:**
JMSException - if an internal error occurs.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
The XAResource interface is a Java mapping of the industry standard XA interface based on the X/Open CAE Specification (Distributed Transaction Processing: The XA Specification).

The XA interface defines the contract between a Resource Manager and a Transaction Manager in a distributed transaction processing (DTP) environment. A JDBC driver or a JMS provider implements this interface to support the association between a global transaction and a database or message service connection.

The XAResource interface can be supported by any transactional resource that is intended to be used by application programs in an environment where transactions are controlled by an external transaction manager. An example of such a resource is a database management system. An application may access data through multiple database connections. Each database connection is enlisted with the transaction manager as a transactional resource. The transaction manager obtains an XAResource for each connection participating in a global transaction. The transaction manager uses the start method to associate the global transaction with the resource, and it uses the end method to disassociate
the transaction from the resource. The resource manager is responsible for associating the global transaction to all work performed on its data between the start and end method invocations.

At transaction commit time, the resource managers are informed by the transaction manager to prepare, commit, or rollback a transaction according to the two-phase commit protocol.

---

### Field Summary

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| **static int** | **TMENDRSCAN**  
Ends a recovery scan. |
| **static int** | **TMFAIL**  
Disassociates the caller and marks the transaction branch rollback-only. |
| **static int** | **TMJOIN**  
Caller is joining existing transaction branch. |
| **static int** | **TMNOFLAGS**  
Use TMNOFLAGS to indicate no flags value is selected. |
| **static int** | **TMONEPHASE**  
Caller is using one-phase optimization. |
| **static int** | **TMRESUME**  
Caller is resuming association with a suspended transaction branch. |
| **static int** | **TSTARTRSCAN**  
Starts a recovery scan. |
| **static int** | **TMSUCCESS**  
Disassociates caller from a transaction branch. |
| **static int** | **TMSUSPEND**  
Caller is suspending (not ending) its association with a transaction branch. |
| **static int** | **XA_OK**  
The transaction work has been prepared normally. |
| **static int** | **XA_RDONLY**  
The transaction branch has been read-only and has |
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void commit(\texttt{Xid} \texttt{xid}, boolean \texttt{onePhase})</strong></td>
<td>Commits the global transaction specified by \texttt{xid}.</td>
</tr>
<tr>
<td><strong>void end(\texttt{Xid} \texttt{xid}, int \texttt{flags})</strong></td>
<td>Ends the work performed on behalf of a transaction branch.</td>
</tr>
<tr>
<td><strong>void forget(\texttt{Xid} \texttt{xid})</strong></td>
<td>Tells the resource manager to forget about a heuristically completed transaction branch.</td>
</tr>
<tr>
<td><strong>int getTransactionTimeout()</strong></td>
<td>Obtains the current transaction timeout value set for this XAResource instance.</td>
</tr>
<tr>
<td><strong>boolean isSameRM(\texttt{XAResource} \texttt{xares})</strong></td>
<td>This method is called to determine if the resource manager instance represented by the target object is the same as the resource manager instance represented by the parameter \texttt{xares}.</td>
</tr>
<tr>
<td><strong>int prepare(\texttt{Xid} \texttt{xid})</strong></td>
<td>Ask the resource manager to prepare for a transaction commit of the transaction specified in \texttt{xid}.</td>
</tr>
<tr>
<td><strong>\texttt{Xid}[] recover(int flag)</strong></td>
<td>Obtains a list of prepared transaction branches from a resource manager.</td>
</tr>
<tr>
<td><strong>void rollback(\texttt{Xid} \texttt{xid})</strong></td>
<td>Informs the resource manager to roll back work done on behalf of a transaction branch.</td>
</tr>
<tr>
<td><strong>boolean setTransactionTimeout(int \texttt{seconds})</strong></td>
<td>Sets the current transaction timeout value for this XAResource instance.</td>
</tr>
<tr>
<td><strong>void start(\texttt{Xid} \texttt{xid}, int \texttt{flags})</strong></td>
<td>Starts work on behalf of a transaction branch specified in \texttt{xid}.</td>
</tr>
</tbody>
</table>
TMENDRSCAN

static final int TMENDRSCAN

   Ends a recovery scan.

   See Also:
   Constant Field Values

TMFAIL

static final int TMFAIL

   Disassociates the caller and marks the transaction branch rollback-only.

   See Also:
   Constant Field Values

TMJOIN

static final int TMJOIN

   Caller is joining existing transaction branch.

   See Also:
   Constant Field Values
TMNFLAGS

static final int TMNFLAGS

Use TMNFLAGS to indicate no flags value is selected.

See Also:
Constant Field Values

TMONEPHASE

static final int TMONEPHASE

Caller is using one-phase optimization.

See Also:
Constant Field Values

TMRESUME

static final int TMRESUME

Caller is resuming association with a suspended transaction branch.

See Also:
Constant Field Values

TMSTARTRSCAN
static final int TMSTARTRSCAN

Starts a recovery scan.

See Also:
  Constant Field Values

TMSUCCESS

static final int TMSUCCESS

Disassociates caller from a transaction branch.

See Also:
  Constant Field Values

TMSUSPEND

static final int TMSUSPEND

Caller is suspending (not ending) its association with a transaction branch.

See Also:
  Constant Field Values

XA_RDONLY

static final int XA_RDONLY

The transaction branch has been read-only and has been
XA_OK
static final int XA_OK

The transaction work has been prepared normally.

See Also:
Constant Field Values

Method Detail

public void commit(Xid xid, boolean onePhase) throws XAException

Throws
XAExceptions XA_HEURHAZ
XA_HEURCOMXA_HEURRBXA_HEURMIX
XAER_RMERRXAER_RMFAILXAER_NOTA
XAER_INVAL XAER_PROTO

commit

void commit(Xid xid,
            boolean onePhase)
        throws XAException
Commits the global transaction specified by xid.

**Parameters:**
- xid - A global transaction identifier
- onePhase - If true, the resource manager should use a one-phase commit protocol to commit the work done on behalf of xid.

**Throws:**
- XAException - An error has occurred. Possible XAExceptions are XA_HEURHAZ, XA_HEURCOM, XA_HEURRB, XA_HEURMIX, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

If the resource manager did not commit the transaction and the parameter onePhase is set to true, the resource manager may throw one of the XA_RB* exceptions. Upon return, the resource manager has rolled back the branch's work and has released all held resources.

```
public void end(Xid xid, int flags) throws XAException
{
    XA
    TMSUSPEND
    start
    TMRESUME
    TMFAIL
    TMSUCCESS
}
```

**Throws**
- XAException: XAException XAER_RMERR

```
        xid
    flags
         start
         TMSUCCESTMFAIL TMSUSPEND
    XAException: XAER_RMFAILXAER_NOTAXAER_INVALXAER_PROTO
```

```
end
```
void end(Xid xid,  
    int flags)  
throws XAException

Ends the work performed on behalf of a transaction branch. The resource manager disassociates the XA resource from the transaction branch specified and lets the transaction complete.

If TMSUSPEND is specified in the flags, the transaction branch is temporarily suspended in an incomplete state. The transaction context is in a suspended state and must be resumed via the start method with TMRESUME specified.

If TMFAIL is specified, the portion of work has failed. The resource manager may mark the transaction as rollback-only.

If TMSUCCESS is specified, the portion of work has completed successfully.

Parameters:
  xid - A global transaction identifier that is the same as the identifier used previously in the start method.
  flags - One of TMSUCCESS, TMFAIL, or TMSUSPEND.

Throws:
  XAException - An error has occurred. Possible XAException values are XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, XAERPROTO, or XA_RB*.

public void forget(Xid xid) throws XAException

xid

Throws XAException: XAER_RMERRXAER_RMFAIL  
XAER_NOTAXAER_INVAL XAER_PROTO

forget

void forget(Xid xid)  
throws XAException
Tells the resource manager to forget about a heuristically completed transaction branch.

**Parameters:**
- `xid` - A global transaction identifier.

**Throws:**
- `XAException` - An error has occurred. Possible exception values are XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAERPROTO.

```java
public int getTransactionTimeout() throws XAException
```

**getTransactionTimeout**

```java
public boolean isSameRM(XAResource xares) throws XAException
```
xares

xares XAResource
return RM true false
Throws XAException: XAER_RMERR XAER_RMFAIL

isSameRM

boolean isSameRM(XAResource xares)
throws XAException

This method is called to determine if the resource manager instance represented by the target object is the same as the resource manager instance represented by the parameter xares.

Parameters:
  xares - An XAResource object whose resource manager instance is to be compared with the resource manager instance of the target object.

Returns:
  true if it's the same RM instance; otherwise false.

Throws:
  XAException - An error has occurred. Possible exception values are XAER_RMERR and XAER_RMFAIL.

public int prepare(Xid xid) throws XAException

xid

throws XAException: XA_RB*XAER_RMERR

Throws XAER_RMFAIL XAER_NOTAXAER_INVAL XAER_PROTO

return XA_RDONLY XA_OK prepare XAException

prepare
int prepare(Xid xid)
    throws XAException

Ask the resource manager to prepare for a transaction commit of the transaction specified in xid.

Parameters:
    xid - A global transaction identifier.

Returns:
    A value indicating the resource manager's vote on the outcome of the transaction. The possible values are: XA_RDONLY or XA_OK. If the resource manager wants to roll back the transaction, it should do so by raising an appropriate XAException in the prepare method.

Throws:
    XAException - An error has occurred. Possible exception values are: XA_RB*, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

public Xid[] recover(int flag) throws XAException

flag:
    TMSTARTRSCANMENDRSCANMNOFLAGS
    MNOFLAGS

Throws:
    XAException: XAER_RMERR XAER_RMFAIL
    XAER_INVAL XAER_PROTO

return 0 XID XAException

recover

Xid[] recover(int flag)
    throws XAException

Obtains a list of prepared transaction branches from a resource manager. The transaction manager calls this method during recovery to obtain the list of transaction branches that are currently in prepared or heuristically completed states.
Parameters:
flag - One of TMSTARTRSCAN, TMENDRSCAN, TMNOFLAGS. TMNOFLAGS must be used when no other flags are set in the parameter.

Returns:
The resource manager returns zero or more XIDs of the transaction branches that are currently in a prepared or heuristically completed state. If an error occurs during the operation, the resource manager should throw the appropriate XAException.

Throws:
XAException - An error has occurred. Possible values are XAER_RMERR, XAER_RMFAIL, XAER_INVAL, and XAER_PROTO.

```java
public void rollback(Xid xid) throws XAException
```

**xid**

XAException: XAExceptions XA_HEURHAZ
XA_HEURCOMXAXA_HEURRBXAXA_HEURMIX
XAER_RMERRXAXAER_RMFAILXAXAER_NOTAXAER_INVAL
XAER_PROTO

Throws
XA_RB*

rollback

```java
void rollback(Xid xid)
```

throws XAException

Informs the resource manager to roll back work done on behalf of a transaction branch.

Parameters:

- **xid** - A global transaction identifier.

Throws:
XAException - An error has occurred. Possible XAExceptions are XA_HEURHAZ, XA_HEURCOM, XA_HEURRB, XA_HEURMIX, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

If the transaction branch is already marked rollback-only the resource manager may throw one of the XA_RB* exceptions. Upon return, the resource manager has rolled back the branch's work and has released all held resources.

### public boolean setTransactionTimeout(int seconds) throws XAException

<table>
<thead>
<tr>
<th>XAResource</th>
<th>setTransactionTimeout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>true  false false</td>
</tr>
</tbody>
</table>

*seconds*

*return*

*Throws* XAException: XAER_RMERR XAER_RMFAIL XAER_INVAL

### setTransactionTimeout

boolean setTransactionTimeout(int seconds) throws XAException

Sets the current transaction timeout value for this XAResource instance. Once set, this timeout value is effective until setTransactionTimeout is invoked again with a different value. To reset the timeout value to the default value used by the resource manager, set the value to zero. If the timeout operation is performed successfully, the method returns true; otherwise false. If a resource manager does not support explicitly setting the transaction timeout value, this method returns false.

**Parameters:**
seconds - The transaction timeout value in seconds.

**Returns:**

- *true* if the transaction timeout value is set successfully;
- *false* otherwise.

**Throws:**

- `XAException` - An error has occurred. Possible exception values are XAER_RMERR, XAER_RMFAIL, or XAER_INVAL.

```java
public void start(Xid xid, int flags) throws XAException

xid
TMJOIN TMRESUME
XAException

xid
flags
TMJOIN TMRESUME
XAException: XA_RB*XAER_RMERR

Throws
XAER_RMFAILXAER_DUPIDXAER_OUTSIDE
XAER_NOTAXAER_INVAL XAER_PROTO
```

**start**

```java
void start(Xid xid,
            int flags)
            throws XAException
```

Starts work on behalf of a transaction branch specified in `xid`. If TMJOIN is specified, the start applies to joining a transaction previously seen by the resource manager. If TMRESUME is specified, the start applies to resuming a suspended transaction specified in the parameter `xid`. If neither TMJOIN nor TMRESUME is specified and the transaction specified by `xid` has previously been seen by the resource manager, the resource manager throws the XAException exception with XAER_DUPIP error code.

**Parameters:**

- `xid` - A global transaction identifier to be associated with the resource.
flags - One of TMNOFLAGS, TMJOIN, or TMRESUME.

**Throws:**

XAException - An error has occurred. Possible exceptions are XA_RB*, XAER_RMERR, XAER_RMFAIL, XAER_DUPID, XAER_OUTSIDE, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS :
The XASession interface extends the capability of Session by adding access to a JMS provider's support for the Java Transaction API (JTA) (optional). This support takes the form of a javax.transaction.xa.XAResource object. The functionality of this object closely resembles that defined by the standard X/Open XA Resource interface.
An application server controls the transactional assignment of an XASession by obtaining its XAResource. It uses the XAResource to assign the session to a transaction, prepare and commit work on the transaction, and so on.

An XAResource provides some fairly sophisticated facilities for interleaving work on multiple transactions, recovering a list of transactions in progress, and so on. A JTA aware JMS provider must fully implement this functionality. This could be done by using the services of a database that supports XA, or a JMS provider may choose to implement this functionality from scratch.

A client of the application server is given what it thinks is a regular JMS Session. Behind the scenes, the application server controls the transaction management of the underlying XASession.

The XASession interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

**Version:**
1.1 February 2, 2002

**Author:**
Mark Hapner, Rich Burridge, Kate Stout

**See Also:**
[Session](#)

---

### Field Summary

Fields inherited from interface javax.jms.Session

- AUTO_ACKNOWLEDGE
- CLIENT_ACKNOWLEDGE
- DUPS_OK_ACKNOWLEDGE
- SESSION_TRANSACTED

---

### Method Summary
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>commit()</td>
<td>Throws a TransactionInProgressException, since it should not be called for an XASession object.</td>
</tr>
<tr>
<td>getSession()</td>
<td>Gets the session associated with this XASession.</td>
</tr>
<tr>
<td>getTransacted()</td>
<td>Indicates whether the session is in transacted mode.</td>
</tr>
<tr>
<td>getXAResource()</td>
<td>Returns an XA resource to the caller.</td>
</tr>
<tr>
<td>rollback()</td>
<td>Throws a TransactionInProgressException, since it should not be called for an XASession object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.Session

- close
- createBrowser
- createBytesMessage
- createConsumer
- createDurableSubscriber
- createMapMessage
- createMessage
- createObjectMessage
- createProducer
- createQueue
- createStreamMessage
- createTemporaryQueue
- createTemporaryTopic
- createTextMessage
- getAcknowledgeMode
- getMessageListener
- recover
- run
- setMessageListener
- unsubscribe

Method Detail

public Session getSession() throws JMSException

    XASession
    return

    Throws JMSException: since 1.1

getSession

Session getSession()
Gets the session associated with this XASession.

**Returns:**
the session object

**Throws:**
`JMSException` - if an internal error occurs.

**Since:**
1.1

```java
public XAResource getXAResource()
```

Returns an XA resource to the caller.

**Returns:**
an XA resource to the caller

---

```java
public boolean getTransacted() throws JMSException
```

**Returns:**
true

**Throws**
`JMSException`: JMS

---

**getTransacted**

```java
boolean getTransacted()
```
Indicates whether the session is in transacted mode.

Specified by:
   getTransacted in interface Session

Returns: true

Throws:
   JMSException - if the JMS provider fails to return the transaction mode due to some internal error.

---

**public void commit() throws JMSException**

TransactionInProgressException XASession

Throws TransactionInProgressException: XASession

**commit**

void commit()
   throws JMSException

Throws a TransactionInProgressException, since it should not be called for an XASession object.

Specified by:
   commit in interface Session

Throws:
   TransactionInProgressException - if the method is called on an XASession.
   JMSException - if the JMS provider fails to commit the transaction due to some internal error.
   TransactionRolledBackException - if the transaction is rolled back due to some internal error during commit.
   IllegalStateException - if the method is not called by a transacted session.

---

**public void rollback() throws JMSException**
rollback

void rollback() throws JMSException

Throws a TransactionInProgressException, since it should not be called for an XASession object.

Specified by:
rollback in interface Session

Throws:
TransactionInProgressException - if the method is called on an XASession.
JMSException - if the JMS provider fails to roll back the transaction due to some internal error.
IllegalStateException - if the method is not called by a transacted session.
The XATerminator interface is used for transaction completion and crash recovery flows.

**Version:**
1.0

**Author:**
Ram Jeyaraman

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>void</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>void</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>int</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Xid[]</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>void</strong></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
public void commit(Xid xid, boolean onePhase) throws XAException

Commits the global transaction specified by xid.

Parameters:
- xid: A global transaction identifier
- onePhase: If true, the resource manager should use a one-phase commit protocol to commit the work done on behalf of xid.

Throws:
- XAException: An error has occurred. Possible XAExceptions are XA_HEURHAZ, XA_HEURCOM, XA_HEURRB, XA_HEURMIX, XAER_RMERR, XAER_RMFIAL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

If the resource manager did not commit the transaction and the
parameter onePhase is set to true, the resource manager may throw one of the XA_RB* exceptions. Upon return, the resource manager has rolled back the branch’s work and has released all held resources.

**public void forget(Xid xid) throws XAException**

  *xid*

  **Throws**

  XAException: XAER_RMERR XAER_RMFAIL XAER_NOTA XAER_INVAL XAER_PROTO

**forget**

void forget(Xid xid)
  throws XAException

  Tells the resource manager to forget about a heuristically completed transaction branch.

  **Parameters:**

  xid - A global transaction identifier.

  **Throws:**

  XAException - An error has occurred. Possible exception values are XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

**public int prepare(Xid xid) throws XAException**

  *xid*

  **Throws**

  XAException: XA_RB* XAER_RMERR

  XAER_RMFAIL XAER_NOTA XAER_INVAL XAER_PROTO

  return javax.transaction.xa.XAResource prepare
    throws XAException
prepare

```java
int prepare(Xid xid)
    throws XAException
```

Ask the resource manager to prepare for a transaction commit of the transaction specified in xid.

**Parameters:**
- `xid` - A global transaction identifier.

**Returns:**
A value indicating the resource manager's vote on the outcome of the transaction. The possible values are: XA_RDONLY or XA_OK. These constants are defined in `javax.transaction.xa.XAResource` interface. If the resource manager wants to roll back the transaction, it should do so by raising an appropriate XAException in the prepare method.

**Throws:**
- `XAException` - An error has occurred. Possible exception values are: XA_RB*, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAER_INVAL, or XAER_PROTO.

---

public `Xid[]` recover(int flag) throws `XAException`

```java
flag TMSTARTRSCANMENDRSCANMNOFLAGS
     MNOFLAGS javax.transaction.xa.XAResource
Throws XAException: XAER_RMERR XAER_RMFAIL
             XAER_INVAL XAER_PROTO
return 0 XID XAException
```

recover

```java
Xid[] recover(int flag)
    throws XAException
```
Obtains a list of prepared transaction branches from a resource manager. The transaction manager calls this method during recovery to obtain the list of transaction branches that are currently in prepared or heuristically completed states.

**Parameters:**
flag - One of TMSTARTRSCAN, TMENDRSCAN, TMNOFLAGS. TMNOFLAGS must be used when no other flags are set in the parameter. These constants are defined in javax.transaction.xa.XAResource interface.

**Returns:**
The resource manager returns zero or more XIDs of the transaction branches that are currently in a prepared or heuristically completed state. If an error occurs during the operation, the resource manager should throw the appropriate XAException.

**Throws:**
- XAException - An error has occurred. Possible values are XAER_RMERR, XAER_RMFAIL, XAER_INVAL, and XAER_PROTO.

```java
public void rollback(Xid xid) throws XAException
```

### Throws
- XAException: XAExceptions XA_HEURHAZ
  - XA_HEURCOM
  - XA_HEURRB
  - XA_HEURMIX
  - XAER_RMERR
  - XAER_RMFAIL
  - XAER_NOTAX
  - XAER_INVAL
  - XAER_PROTO

### XA_RB*

```java
void rollback(Xid xid)
```

throws XAException
Informs the resource manager to roll back work done on behalf of a transaction branch.

**Parameters:**
- `xid` - A global transaction identifier.

**Throws:**
- `XAException` - An error has occurred. Possible XAExceptions are XA_HEURHAZ, XA_HEURCOM, XA_HEURRB, XA_HEURMIX, XAER_RMERR, XAER_RMFAIL, XAER_NOTA, XAERINVAL, or XAER_PROTO.

If the transaction branch is already marked rollback-only the resource manager may throw one of the XA_RB* exceptions. Upon return, the resource manager has rolled back the branch's work and has released all held resources.

---

**Overview**
**Package**
**Tree**
**Deprecated**
**Index**
**Help**

PREV CLASS| NEXT CLASS| SUMMARY: NESTED | FIELD | CONSTR | METHOD| FRAMES| NO FRAMES| DETAIL: FIELD | CONSTR | METHOD
|---|---|---|---|---|---|---|---|---|---|---|

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
An XATopicConnection provides the same create options as TopicConnection (optional). The Topic connections created are transactional.

The XATopicConnection interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:
1.1 - February 2, 2002

Author:
Mark Hapner, Rich Burridge, Kate Stout

See Also:
XAConnection
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createTopicSession</code></td>
<td>Creates an XATopicSession object.</td>
</tr>
<tr>
<td><code>createXATopicSession</code></td>
<td>Creates an XATopicSession object.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `javax.jms.XAConnection`
- `createSession`, `createXASession`

Methods inherited from interface `javax.jms.Connection`
- `close`, `createConnectionConsumer`, `getClientID`, `getExceptionListener`, `getMetaData`, `setClientID`, `setExceptionListener`, `start`, `stop`

Methods inherited from interface `javax.jms.TopicConnection`
- `createConnectionConsumer`, `createDurableConnectionConsumer`

Methods inherited from interface `javax.jms.Connection`
- `close`, `createConnectionConsumer`, `getClientID`, `getExceptionListener`, `getMetaData`, `setClientID`, `setExceptionListener`, `start`, `stop`

Method Detail

```java
public XATopicSession createXATopicSession() throws JMSException

    XATopicSession
    return

    Throws
    JMSException:     XATopicConnection     XA
```

createXATopicSession
**createXATopicSession()**

Creates an XATopicSession object.

**Returns:**
- a newly created XA topic session

**Throws:**
- **JMSException** - if the XATopicConnection object fails to create an XA topic session due to some internal error.

**public TopicSession createTopicSession(boolean transacted, int acknowledgeMode) throws JMSException**

<table>
<thead>
<tr>
<th>Return Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XATopicSession</td>
<td></td>
</tr>
<tr>
<td>transacted</td>
<td></td>
</tr>
<tr>
<td>acknowledgeMode</td>
<td></td>
</tr>
</tbody>
</table>

**createTopicSession**

**createTopicSession**(boolean transacted, int acknowledgeMode)

<table>
<thead>
<tr>
<th>Throws</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMSException</td>
<td></td>
</tr>
</tbody>
</table>

Creates an XATopicSession object.

**Specified by:**
createTopicSession in interface TopicConnection

**Parameters:**
- transacted - usage undefined
- acknowledgeMode - usage undefined

**Returns:**
a newly created XA topic session

**Throws:**
- **JMSException** - if the XATopicConnection object fails to create an XA topic session due to some internal error.
See Also:
Session.AUTO_ACKNOWLEDGE, Session.CLIENT_ACKNOWLEDGE, Session.DUPS_OK_ACKNOWLEDGE

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.jms Interface XATopicConnectionFactory

All Superinterfaces:
javax.jms.ConnectionFactory, javax.jms.TopicConnectionFactory, javax.jms.XAConnectionFactory

public interface XATopicConnectionFactory
    extends javax.jms.XAConnectionFactory, javax.jms.TopicConnectionFactory

Implements: javax.jms.XAConnectionFactory, javax.jms.TopicConnectionFactory

An XATopicConnectionFactory provides the same create options as a TopicConnectionFactory (optional).

The XATopicConnectionFactory interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than use these XA interfaces directly.

Version:
    1.1 February 2, 2002

Author:
    Mark Hapner, Rich Burridge, Kate Stout

See Also:
    javax.jms.TopicConnectionFactory, javax.jms.XAConnectionFactory
Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>createXATopicConnection()</td>
<td>Creates an XA topic connection with the default user identity.</td>
</tr>
<tr>
<td>createXATopicConnection(String userName, String password)</td>
<td>Creates an XA topic connection with the specified user identity.</td>
</tr>
</tbody>
</table>

Methods inherited from interface javax.jms.XAConnectionFactory
createXAConnection, createXAConnection

Methods inherited from interface javax.jms.TopicConnectionFactory
createTopicConnection, createTopicConnection

Methods inherited from interface javax.jms.ConnectionFactory
createConnection, createConnection

Method Detail

```java
public XATopicConnection createXATopicConnection() throws JMSException
```

```
return XAConnec

Throws JMSException: JMS XA
Throws JMSSecurityException:
```

createXATopicConnection

```java
XATopicConnection createXATopicConnection() throws JMSException
```
Creates an XA topic connection with the default user identity. The connection is created in stopped mode. No messages will be delivered until the Connection.start method is explicitly called.

Returns:
- a newly created XA topic connection

Throws:
- JMSException - if the JMS provider fails to create an XA topic connection due to some internal error.
- JMSSecurityException - if client authentication fails due to an invalid user name or password.

```java
public XATopicConnection createXATopicConnection(String userName, String password) throws JMSException
```

Parameters:
- `userName` - the caller's user name
- `password` - the caller's password

Returns:
a newly created XA topic connection

**Throws:**

*JMSException* - if the JMS provider fails to create an XA topic connection due to some internal error.

*JMSSecurityException* - if client authentication fails due to an invalid user name or password.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.jms Interface XATopicSession

All Superinterfaces:
   Runnable, Session, XASession

```
public interface XATopicSession
extends XASession

Implements: XASession
```

XATopicSession TopicSession TopicSubscriber TopicPublisher

XATopicSession JMS JMS XA

version 1.1 January 2, 2002
See also javax.jms.XASession, javax.jms.TopicSession

An XATopicSession provides a regular TopicSession which can be used to create TopicSubscriber and TopicPublisher objects (optional).

The XATopicSession interface is optional. JMS providers are not required to support this interface. This interface is for use by JMS providers to support transactional environments. Client programs are strongly encouraged to use the transactional support available in their environment, rather than using these XA interfaces directly.

Version:
   1.1 January 2, 2002

Author:
   Mark Hapner, Rich Burridge, Kate Stout

See Also:
   XASession, TopicSession
## Field Summary

### Fields inherited from interface javax.jms.Session

- AUTO_ACKNOWLEDGE
- CLIENT_ACKNOWLEDGE
- DUPS_OK_ACKNOWLEDGE
- SESSION_TRANSACTED

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>TopicSession getTopicSession()</code></td>
<td>Gets the topic session associated with this XATopicSession.</td>
</tr>
</tbody>
</table>

## Method Detail

```java
public TopicSession getTopicSession() throws JMSException
```

XATopicSession

```java
return
```

**Throws**

```java
JMSException:
```
getTopicSession

**TopicSession** getTopicSession()

**Throws: JMSException**

Gets the topic session associated with this XATopicSession.

**Returns:**

the topic session object

**Throws:**

JMSException - if an internal error occurs.
public interface Xid

Xid X/Open XID Java ID ID Xid

The Xid interface is a Java mapping of the X/Open transaction identifier XID structure. This interface specifies three accessor methods to retrieve a global transaction's format ID, global transaction ID, and branch qualifier. The Xid interface is used by the transaction manager and the resource managers. This interface is not visible to the application programs.

### Field Summary

| Static int | MAXBQUALSIZE | Maximum number of bytes returned by getBqual. |
| Static int | MAXGTRIDSIZE | Maximum number of bytes returned by getGtrid. |

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte[]</td>
<td>getBranchQualifier()</td>
</tr>
<tr>
<td>int</td>
<td>getFormatId()</td>
</tr>
<tr>
<td>byte[]</td>
<td>getGlobalTransactionId()</td>
</tr>
</tbody>
</table>
### Field Detail

**MAXGTRIDSIZE**

static final int `MAXGTRIDSIZE`

Maximum number of bytes returned by `getGtrid`.

**See Also:**

[Constant Field Values]

---

**MAXBQUALSIZE**

static final int `MAXBQUALSIZE`

Maximum number of bytes returned by `getBqual`.

**See Also:**

[Constant Field Values]

### Method Detail

**public int getFormatId()**

```
XID
    return O OSI CCR
```

**getFormatId**

```
int getFormatId()
    Obtain the format identifier part of the XID.
```
Returns:
Format identifier. O means the OSI CCR format.

```java
public byte[] getGlobalTransactionId()
XID
    return

getGlobalTransactionId

byte[] getGlobalTransactionId()

    Obtain the global transaction identifier part of XID as an array of bytes.

    Returns:
    Global transaction identifier.

public byte[] getBranchQualifier()
XID
    return

getBranchQualifier

byte[] getBranchQualifier()

    Obtain the transaction branch identifier part of XID as an array of bytes.

    Returns:
    Global transaction identifier.
```
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: NESTED</td>
<td>ENUM CONSTANTS</td>
</tr>
</tbody>
</table>
javax.xml.bind.annotation  

**Enum XmlAccessOrder**

**java.lang.Object**  
| java.lang.Enum<XmlAccessOrder> | javax.xml.bind.annotation.XmlAccessOrder |

**All Implemented Interfaces:**  
Serializable, Comparable<XmlAccessOrder>

```java
public enum XmlAccessOrder
extends Enum<XmlAccessOrder>
```

**Extends:** Enum<E>

XmlAccessorOrder  
JAXB  
since  
version  
See also  
javax.xml.bind.annotation.XmlAccessorOrder

Used by XmlAccessorOrder to control the ordering of properties and fields in a JAXB bound class.

**Since:**  
JAXB2.0

**Version:**  
$Revision: 1.1 $

**Author:**  
Sekhar Vajjhala, Sun Microsystems, Inc.

**See Also:**  
XmlAccessorOrder

---

**Enum Constant Summary**

| ALPHABETICAL |
The ordering of fields and properties in a class is in alphabetical order as determined by the method java.lang.String.compareTo(String anotherString).

**UNDEFINED**

The ordering of fields and properties in a class is undefined.

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static XmlAccessOrder.valueOf(String name)</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td>static XmlAccessOrder[] values()</td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Enum**

- clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf

**Methods inherited from class java.lang.Object**

- finalize, getClass, notify, notifyAll, wait, wait, wait

**Enum Constant Detail**

**UNDEFINED**

public static final XmlAccessOrder UNDEFINED

The ordering of fields and properties in a class is undefined.
The ordering of fields and properties in a class is in alphabetical order as determined by the method java.lang.String.compareTo(String anotherString).

### Method Detail

**final public static XmlAccessOrder[] values()**

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(XmlAccessOrder c : XmlAccessOrder.values())
    System.out.println(c);
```

**Returns:**
- an array containing the constants of this enum type, in the order they're declared

**public static XmlAccessOrder valueOf(String name)**

Returns the enum constant of this type with the specified name. The string must match *exactly* an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not...
permitted.)

**Parameters:**
- `name` - the name of the enum constant to be returned.

**Returns:**
- the enum constant with the specified name

**Throws:**
- `IllegalArgumentException` - if this enum type has no constant with the specified name
Annotation Type
XmlAccessorOrder

@Inherited
@Retention(value=RUNTIME)
@Target(value={PACKAGE, TYPE})
public @interface XmlAccessorOrder

Implements: Annotation
@Inherited
@Retention(value=RUNTIME)
@Target(value={PACKAGE, TYPE})

@XmlAccessorOrder
•
•

javax.xml.bind.package javadoc ""

@XmlAccessorOrder
•
•
•

@XmlAccessorOrder

@XmlAccessorType(XmlAccessOrder.UNDEFINED)

@XmlAccessorOrder  @XmlAccessorOrder
Controls the ordering of fields and properties in a class.

Usage

@XmlAccessorType(XmlAccessOrder.UNDEFINED)

 XmlAccessorType XmlRootElement XmlAccessorType XmlSchema XmlSchemaType XmlSchemaTypes
 XmlJavaTypeAdapter XmlJavaTypeAdapter

since JAXB2.0
version $Revision: 1.11 $
See also javax.xml.bind.annotation.XmlAccessorType

@XmlAccessorType(XmlAccessOrder.UNDEFINED)

package a top level class


The annotation @XmlAccessorType on a package applies to all classes in a package. The following inheritance semantics apply:

• If there is a @XmlAccessorType on a class, then it is used.
• Otherwise, if a @XmlAccessorType exists on one of its super classes, then it is inherited.
• Otherwise, the @XmlAccessorType on a package is inherited.

Defaulting Rules:

By default, if @XmlAccessorType on a package is absent, then the following package level annotation is assumed.

@XmlElement(XmlAccessOrder.UNDEFINED)
By default, if @XmlAccessorOrder on a class is absent and none of super classes is annotated with XmlAccessorOrder, then the following default on the class is assumed:

@XmlAccessorType(XmlAccessOrder.UNDEFINED)

This annotation can be used with the following annotations: XmlType, XmlRootElement, XmlAccessorType, XmlSchema, XmlSchemaType, XmlSchemaTypes, XmlJavaTypeAdapter. It can also be used with the following annotations at the package level: XmlJavaTypeAdapter.

Since: JAXB2.0
Version: $Revision: 1.11 $
Author: Sekhar Vajjhala, Sun Microsystems, Inc.
See Also: XmlAccessOrder
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  Annotation Type

XmlAccessorType

--

@Inherited
@Retention(value=RUNTIME)
@Target(value={PACKAGE,TYPE})
public @interface XmlAccessorType

**Implements:** Annotation
@Inherited
@Retention(value=RUNTIME)
@Target(value={PACKAGE,TYPE})

Javabean

@XmlAccessorType

•

javax.xml.bind.package javadoc ""

@XmlAccessorType

• @XmlAccessorType
• @XmlAccessorType
•  @XmlAccessorOrder

@XmlAccessorType

@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)
@XmlAccessorType

@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)

```
@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)

Xm1Type
Xm1RootElement
Xm1Schema
Xm1SchemaType
Xm1SchemaTypes

Xm1JavaTypeAdapter

Xm1JavaTypeAdapter

since

JAXB2.0

version

$Revision: 1.9 $

See also

javax.xml.bind.annotation.XmlAccessorType
```

Controls whether fields or Javabean properties are serialized by default.

**Usage**

@XmlAccessorType annotation can be used with the following program elements:

- package
- a top level class


This annotation provides control over the default serialization of properties and fields in a class.

The annotation @XmlAccessorType on a package applies to all classes in the package. The following inheritance semantics apply:

- If there is a @XmlAccessorType on a class, then it is used.
- Otherwise, if a @XmlAccessorType exists on one of its super classes, then it is inherited.
- Otherwise, the @XmlAccessorType on a package is inherited.

**Defaulting Rules:**

By default, if @XmlAccessorType on a package is absent, then the
following package level annotation is assumed.

```java
@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)
```

By default, if `@XmlAccessorType` on a class is absent, and none of its super classes is annotated with `@XmlAccessorType`, then the following default on the class is assumed:

```java
@XmlAccessorType(XmlAccessType.PUBLIC_MEMBER)
```

This annotation can be used with the following annotations: `XmlType`, `XmlRootElement`, `XmlAccessorType`, `XmlSchema`, `XmlSchemaType`, `XmlSchemaTypes`, `XmlJavaTypeAdapter`. It can also be used with the following annotations at the package level: `XmlJavaTypeAdapter`.

Since: JAXB2.0

Version: $Revision: 1.9$

Author: Sekhar Vajjhala, Sun Microsystems, Inc.

See Also: `XmlAccessType`
public abstract XmlAccessType value

Specifies whether fields or properties are serialized.

See Also:
    XmlAccessType

Default:
    PUBLIC_MEMBER
javax.xml.bind.annotation  **Enum XmlAccessType**

java.lang.Object
  └java.lang.Enum<XmlAccessType>
    └javax.xml.bind.annotation.XmlAccessType

**All Implemented Interfaces:**
  Serializable, Comparable<XmlAccessType>

```java
public enum XmlAccessType
  extends Enum<XmlAccessType>

Extends: Enum<E>
```

XmlAccessorType
  since JAXB2.0
  version $Revision: 1.1$
  See also javax.xml.bind.annotation.XmlAccessorType

Used by XmlAccessorType to control serialization of fields or properties.

Since:
  JAXB2.0

Version:
  $Revision: 1.1$

Author:
  Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:
  XmlAccessorType

---

**Enum Constant Summary**

<table>
<thead>
<tr>
<th>FIELD</th>
</tr>
</thead>
<tbody>
<tr>
<td>FIELD</td>
</tr>
</tbody>
</table>

Every non static, non transient field in a JAXB-bound class will be
automatically bound to XML, unless annotated by `XmlTransient`.

<table>
<thead>
<tr>
<th>NONE</th>
<th>None of the fields or properties is bound to XML unless they are specifically annotated with some of the JAXB annotations.</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROPERTY</td>
<td>Every getter/setter pair in a JAXB-bound class will be automatically bound to XML, unless annotated by <code>XmlTransient</code>.</td>
</tr>
<tr>
<td>PUBLIC_MEMBER</td>
<td>Every public getter/setter pair and every public field will be automatically bound to XML, unless annotated by <code>XmlTransient</code>.</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>static <code>XmlAccessType</code></th>
<th><code>valueOf(String name)</code></th>
<th>Returns the enum constant of this type with the specified name.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static <code>XmlAccessType[]</code></td>
<td><code>values()</code></td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
</tbody>
</table>

### Methods inherited from class `java.lang.Enum`  
`clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf`

### Methods inherited from class `java.lang.Object`  
`finalize, getClass, notify, notifyAll, wait, wait, wait`

### Enum Constant Detail

**PROPERTY**

```java
public static final XmlAccessType PROPERTY
```
Every getter/setter pair in a JAXB-bound class will be automatically bound to XML, unless annotated by `XmlTransient`. Fields are bound to XML only when they are explicitly annotated by some of the JAXB annotations.

---

**FIELD**

```java
public static final XmlAccessType FIELD
```

Every non static, non transient field in a JAXB-bound class will be automatically bound to XML, unless annotated by `XmlTransient`. Getter/setter pairs are bound to XML only when they are explicitly annotated by some of the JAXB annotations.

---

**PUBLIC_MEMBER**

```java
public static final XmlAccessType PUBLIC_MEMBER
```

Every public getter/setter pair and every public field will be automatically bound to XML, unless annotated by `XmlTransient`. Fields or getter/setter pairs that are private, protected, or defaulted to package-only access are bound to XML only when they are explicitly annotated by the appropriate JAXB annotations.

---

**NONE**

```java
public static final XmlAccessType NONE
```

None of the fields or properties is bound to XML unless they are specifically annotated with some of the JAXB annotations.
Method Detail

final public static XmlAccessType[] values()

values

public static final XmlAccessType[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

```java
for(XmlAccessType c : XmlAccessType.values())
    System.out.println(c);
```

**Returns:**

an array containing the constants of this enum type, in the order they're declared

---

public static XmlAccessType valueOf(String name)

valueOf

public static XmlAccessType valueOf(String name)

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

**Parameters:**

name - the name of the enum constant to be returned.

**Returns:**

the enum constant with the specified name

**Throws:**
IllegalArgumentException - if this enum type has no constant with the specified name

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation.adapters Class
XmlAdapter<ValueType,BoundType>

java.lang.Object
  ↓ javax.xml.bind.annotation.adapters.XmlAdapter<ValueType,BoundType>

Type Parameters:
  BoundType - The type that JAXB doesn't know how to handle. An
  adapter is written to allow this type to be used as an in-memory
  representation through the ValueType.
  ValueType - The type that JAXB knows how to handle out of the box.

Direct Known Subclasses:
  CollapsedStringAdapter, HexBinaryAdapter,
  NormalizedStringAdapter

public abstract class XmlAdapter<ValueType,BoundType>
extends Object

Extended by: CollapsedStringAdapter, HexBinaryAdapter,
NormalizedStringAdapter

Java

Java XML HashMap JavaBean XML Java
Java XML Java xs:DateTime
XmlGregorianCalendar xs:DateTime
MyXmlGregorianCalendar XML

bound value value bound JAXB

- XmlAdapter.marshall(...) JAXB XmlAdapter.marshall(..)
  bound value value XML
- XmlAdapter.unmarshall(...) JAXB XML value
XmlAdapter.unmarshal(..) value bound

- 
  - XmlJavaTypeAdapter

HashMap

@XmlAdapter  @XmlJavaTypeAdapter  HashMap

1  HashMap XML

<hashmap>
<entry key="id123">this is a value</entry>
<entry key="id312">this is another value</entry>
...
</hashmap>

2  XML

<xs:complexType name="myHashMapType">
  <xs:sequence>
    <xs:element name="entry" type="myHashMapEntryType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>

<xs:complexType name="myHashMapEntryType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="key" type="xs:int"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

3  value

    public class MyHashMapType {
        List<MyHashMapEntryType> entry;
    }
public class MyHashMapEntryType {
    @XmlAttribute
    public Integer key;

    @XmlValue
    public String value;
}

4 value (MyHashMapType) bound HashMap

public final class MyHashMapAdapter extends XmlAdapter<HashMap, MyHashMapType> { ... } }

5

public class Foo {
    @XmlJavaTypeAdapter(MyHashMapAdapter.class)
    HashMap hashmap;
    ...
}

<xs:complexType name="Foo">
<xs:sequence>
<xs:element name="hashmap" type="myHashMapType"
</xs:sequence>
</xs:complexType>

BoundType JAXB

ValueType JAXB

since JAXB 2.0

See also javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter

Adapts a Java type for custom marshaling.

Usage:

Some Java types do not map naturally to a XML representation, for
example HashMap or other non JavaBean classes. Conversely, a XML representation may map to a Java type but an application may choose to access the XML representation using another Java type. For example, the schema to Java binding rules bind xs:DateTime by default to XmlGregorianCalendar. But an application may desire to bind xs:DateTime to a custom type, MyXmlGregorianCalendar, for example. In both cases, there is a mismatch between bound type, used by an application to access XML content and the value type, that is mapped to an XML representation.

This abstract class defines methods for adapting a bound type to a value type or vice versa. The methods are invoked by the JAXB binding framework during marshaling and unmarshalling:

- **XmlAdapter.marshal(...):** During marshalling, JAXB binding framework invokes XmlAdapter.marshal(..) to adapt a bound type to value type, which is then marshaled to XML representation.
- **XmlAdapter.unmarshal(...):** During unmarshalling, JAXB binding framework first unmarshals XML representation to a value type and then invokes XmlAdapter.unmarshal(..) to adapt the value type to a bound type.

Writing an adapter therefore involves the following steps:

- Write an adapter that implements this abstract class.
- Install the adapter using the annotation `@XmlJavaTypeAdapter`

**Example:** Customized mapping of HashMap

The following example illustrates the use of `@XmlAdapter` and `@XmlJavaTypeAdapter` to customize the mapping of a HashMap.

**Step 1:** Determine the desired XML representation for HashMap.

```xml
<hashmap>
  <entry key="id123">this is a value</entry>
  <entry key="id312">this is another value</entry>
  ...
</hashmap>
```
Step 2: Determine the schema definition that the desired XML representation shown above should follow.

```xml
<xs:complexType name="myHashMapType">
  <xs:sequence>
    <xs:element name="entry" type="myHashMapEntryType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
<xs:complexType name="myHashMapEntryType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="key" type="xs:int"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Step 3: Write value types that can generate the above schema definition.

```java
public class MyHashMapType {
    List<MyHashMapEntryType> entry;
}

public class MyHashMapEntryType {
    @XmlAttribute
    public Integer key;

    @XmlValue
    public String value;
}
```

Step 4: Write the adapter that adapts the value type, MyHashMapType to a bound type, HashMap, used by the application.

```java
public final class MyHashMapAdapter extends XmlAdapter<HashMap, MyHashMapType> {
    ...
}
```

Step 5: Use the adapter.

```java
public class Foo {
```
@XmlJavaTypeAdapter(MyHashMapAdapter.class)
HashMap hashmap;
...
}

The above code fragment will map to the following schema:

```xml
<xs:complexType name="Foo">
  <xs:sequence>
    <xs:element name="hashmap" type="myHashMapType"/>
  </xs:sequence>
</xs:complexType>
```

Since:
JAXB 2.0

Author:
- Sekhar Vajjhala, Sun Microsystems Inc.
- Kohsuke Kawaguchi, Sun Microsystems Inc.

See Also:
XmlJavaTypeAdapter

---

### Constructor Summary

| protected | XmlAdapter() | Do-nothing constructor for the derived classes. |

### Method Summary

| abstract ValueType | marshal(BoundType v) | Convert a bound type to a value type. |
| abstract BoundType  | unmarshal(ValueType v) | Convert a value type to a bound type. |

Methods inherited from class java.lang.Object:
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait
Constructor Detail

protected XmlAdapter()

XmlAdapter

protected XmlAdapter()

Do-nothing constructor for the derived classes.

Method Detail

unmarshal

public abstract BoundType unmarshal(ValueType v)

throws Exception

Convert a value type to a bound type.

Parameters:
  v - The value to be converted. Can be null.

Throws:
  Exception - if there's an error during the conversion. The caller is responsible for reporting the error to the user through ValidationEventHandler.

marshal

public abstract ValueType marshal(BoundType v)

throws Exception
Convert a bound type to a value type.

**Parameters:**

- `v` - The value to be converted. Can be null.

**Throws:**

- `Exception` - if there's an error during the conversion. The caller is responsible for reporting the error to the user through `ValidationEventHandler`.

---

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
### javax.xml.bind.annotation Annotation Type XmlAnyAttribute

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

public @interface XmlAnyAttribute

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean

@XmlAnyAttribute

- JavaBean
- static transient

Maps a JavaBean property to a map of wildcard attributes.

**Usage**

The @XmlAnyAttribute annotation can be used with the following program elements:
JavaBean property
non static, non transient field


The usage is subject to the following constraints:

- At most one field or property in a class can be annotated with @XmlAnyAttribute.
- The type of the property or the field must java.util.Map

While processing attributes to be unmarshalled into a value class, each attribute that is not statically associated with another JavaBean property, viaXmlAttribute, is entered into the wildcard attribute map represented by Map<QName, Object>. The attribute QName is the map's key. The key's value is the String value of the attribute.

Since: JAXB2.0

Author: Kohsuke Kawaguchi, Sun Microsystems, Inc.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: REQUIRED | OPTIONAL FRAMES NO FRAMES DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  Annotation Type XmlAnyElement

@interface XmlAnyElement

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlAnyElement

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean  XML / JAXB

xml JAXB "catch-all" JavaBean JavaBean
JavaBean  @XmlElement  @XmlElementRef  xml "catch-all"
@XmlAnyElement
public Element[] others;

// Collection of Element or JAXB elements.
@XmlAnyElement(lax="true")
public Object[] others;

@XmlAnyElement
private List<Element> nodes;

@XmlAnyElement
private Element node;
XmlElement XmlAttribute XmlValue XmlElements XmlID XmlIDREF

XmlAnyElement JavaBean
XmlJavaTypeAdapter  DOM DOM  XML

XmlMixed

// List of java.lang.String or DOM nodes.
@XmlElement @XmlMixed
List<Object> others;
Java

class Foo {
    int a;
    int b;
    @XmlAnyElement
    List<Element> any;
}

Java

<?xml version="1.0" encoding="UTF-8"?>
<foo xmlns:e="extra">
    <a>1</a>
    <e:other> // this will be bound to DOM, because unmarshalling
        <b>3</b> // this will be bound to DOM, because the a
    <c>5</c>
</foo>

Java

<xs:complexType name="bar">
    <xs:complexContent>
        <xs:extension base="foo">
            <xs:sequence>
                <xs:element name="c" type="xs:int"/>
                <xs:element name="d" type="xs:int"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>
```java
class Bar extends Foo {
    int c;
    // Foo.getAny() also represents wildcard content for type definition
}
```
**Java**

```java
&lt;xs:complexType name="foo">
    &lt;xs:choice maxOccurs="unbounded" minoccurs="0">
        &lt;xs:element name="a" type="xs:int"/>
        &lt;/xs:element&gt;&lt;xs:element name="b" type="xs:int"/>
        &lt;/xs:element&gt;&lt;xs:any namespace="##other" processContents="lax"/>
    &lt;/xs:choice&gt;
&lt;/xs:complexType&gt;

class Foo {
    @XmlAnyElement(lax="true")
    @XmlElementRefs(
        @XmlElementRef(name="a", type=JAXBElement.class)
        @XmlElementRef(name="b", type=JAXBElement.class)
    )
    List&lt;Object&gt; others;
}

@XmlRegistry
class ObjectFactory {
    ...
    @XmlElementDecl(name = "a", namespace = ",", scope = Foo.class)
    JAXBElement&lt;Integer&gt; createFooA( Integer i ) { ... }
    @XmlElementDecl(name = "b", namespace = ",", scope = Foo.class)
    JAXBElement&lt;Integer&gt; createFooB( Integer i ) { ... }
}

&lt;foo xmlns:e="extra"&gt;
    &lt;a&gt;1&lt;/a&gt;  // this will unmarshal to a &lt;a href="#"&gt;1&lt;/a&gt;
    &lt;/e:other&gt; // this will unmarshal to a DOM &lt;a href="#"&gt;1&lt;/a&gt;
    &lt;b&gt;3&lt;/b&gt;  // this will unmarshal to a &lt;b href="#"&gt;3&lt;/b&gt;
    &lt;/e:other&gt;&lt;/foo&gt;
```
W3C XML "lax"

lax "lax" Java

```java
@XmlRootElement
class Foo {
    @XmlAnyElement(lax=true)
    public Object[] others;
}
```

```xml
<foo>
    <unknown/>
</unknown></foo><foo>
</foo>
```

Foo foo = unmarshal();
// 1 for 'unknown', another for 'foo'
assert foo.others.length==2;
// 'unknown' unmarshals to a DOM element
assert foo.others[0] instanceof Element;
// because of lax=true, the 'foo' element eagerly
// unmarshals to a Foo object.
assert foo.others[1] instanceof Foo;

since JAXB2.0

Maps a JavaBean property to XML infoset representation and/or JAXB element.

This annotation serves as a "catch-all" property while unmarshalling xml content into a instance of a JAXB annotated class. It typically annotates a multi-valued JavaBean property, but it can occur on single value JavaBean property. During unmarshalling, each xml element that does not match a static @XmlElement or @XmlElementRef annotation for the other JavaBean properties on the class, is added to this "catch-all" property.
Usages:

```java
@XmlElement
public Element[] others;

// Collection of Element or JAXB elements.
@XmlElement(lax="true")
public Object[] others;

@XmlElement
private List<Element> nodes;

@XmlElement
private Element node;
```
Restriction usage constraints

This annotation is mutually exclusive with `XmlElement`, `XmlAttribute`, `XmlValue`, `XmlElements`, `XmlID`, and `XmlIDREF`.

There can be only one `XmlAnyElement` annotated JavaBean property in a class and its super classes.
Relationship to other annotations

This annotation can be used with `XmlJavaTypeAdapter`, so that users can map their own data structure to DOM, which in turn can be composed into XML.

This annotation can be used with `XmlMixed` like this:

```java
// List of java.lang.String or DOM nodes.
@XmlElement @XmlMixed
List<Object> others;
```
Schema To Java example

The following schema would produce the following Java class:

```xml
&lt;xs:complextype name="foo"&gt;
    &lt;xs:sequence&gt;
        &lt;xs:element name="a" type="xs:int"/&gt;
        &lt;xs:element name="b" type="xs:int"&gt;
            &lt;xsi:namespace prefix="#other" processContents="lax" minOccurs="0" maxOccurs="unbounded" /&gt;
        &lt;/xs:element&gt;
    &lt;/xs:sequence&gt;
&lt;/xs:complextype&gt;

class Foo {
    int a;
    int b;
    @XmlAnyElement
    List&lt;Element&gt; any;
}
```

It can unmarshal instances like

```xml
&lt;foo xmlns:e="extra"&gt;
    &lt;a&gt;1&lt;/a&gt;
    &lt;e:other&gt; // this will be bound to DOM, because unmarshalling is orderless
    &lt;b&gt;3&lt;/b&gt;
    &lt;/e:other&gt;
    &lt;/e:other&gt;
    &lt;e:other&gt;
    &lt;c&gt;5&lt;/c&gt; // this will be bound to DOM, because the annotation doesn't remember namespaces.
    &lt;/e:other&gt;
&lt;/foo&gt;
```

The following schema would produce the following Java class:

```xml
&lt;xs:complextype name="bar"&gt;
    &lt;xs:complexContent&gt;
        &lt;xs:extension base="foo"&gt;
            &lt;xs:sequence&gt;
                &lt;xs:element name="c" type="xs:int"/&gt;
                &lt;xsi:namespace prefix="#other" processContents="lax" minOccurs="0" maxOccurs="unbounded" /&gt;
            &lt;/xs:sequence&gt;
        &lt;/xs:extension&gt;
    &lt;/xs:complexContent&gt;
&lt;/xs:complextype&gt;
```
class Bar extends Foo {
    int c;
    // Foo.getAny() also represents wildcard content for type definit
}

It can unmarshal instances like

&lt;bar xmlns:e="extra"&gt;
    &lt;a&gt;1&lt;/a&gt;
    &lt;/e:other&gt;  // this will be bound to DOM, because unmarshal
    &lt;b&gt;3&lt;/b&gt;  
    &lt;/e:other&gt;&lt;/e:other&gt;  // this now goes to Bar.c
    &lt;c&gt;5&lt;/c&gt;  // this will go to Foo.any
&lt;/e:other&gt;&lt;/bar&gt;
Using `XmlAnyElement` with `XmlElementRef`

The `XmlAnyElement` annotation can be used with `XmlElementRefs` to designate additional elements that can participate in the content tree.

The following schema would produce the following Java class:

```xml
<xs:complexType name="foo">
  <xs:choice maxOccurs="unbounded" minOccurs="0">  
    <xs:element name="a" type="xs:int"></xs:element>
    <xs:element name="b" type="xs:int"></xs:element>
    <xs:any namespace="##other" processContents="lax"></xs:any>
  </xs:choice>
</xs:complexType>

class Foo {
  @XmlAnyElement(lax="true")
  @XmlElementRefs({
    @XmlElementRef(name="a", type=JAXBElement.class)
    @XmlElementRef(name="b", type=JAXBElement.class)
  })
  List<Object> others;
}

@XmlRegistry
class ObjectFactory {
...
  @XmlElementDecl(name = "a", namespace = ",", scope = Foo.class)
  JAXBElement<Integer> createFooA(Integer i) { ... }

  @XmlElementDecl(name = "b", namespace = ",", scope = Foo.class)
  JAXBElement<Integer> createFooB(Integer i) { ... }
}

It can unmarshal instances like

```xml
&lt;foo xmlns:e="extra"&gt;
  &lt;a&gt;1&lt;/a&gt;  // this will unmarshal to a &lt;a href="
  &lt;e:other&gt; // this will unmarshal to a DOM &lt;a href="http
  &lt;b&gt;3&lt;/b&gt;  // this will unmarshal to a &lt;a href="
  &lt;/e:other&gt;&lt;/foo&gt;
```
W3C XML Schema "lax" wildcard emulation

The lax element of the annotation enables the emulation of the "lax" wildcard semantics. For example, when the Java source code is annotated like this:

```java
@XmlRootElement
class Foo {
    @XmlAnyElement(lax=true)
    public Object[] others;
}
```

then the following document will unmarshal like this:

```xml
<foo>
    <unknown/>
    <unknown/>
</foo>
<foo>
</foo>
```

Foo foo = unmarshal();
// 1 for 'unknown', another for 'foo'
assert foo.others.length==2;
// 'unknown' unmarshals to a DOM element
assert foo.others[0] instanceof Element;
// because of lax=true, the 'foo' element eagerly
// unmarshals to a Foo object.
assert foo.others[1] instanceof Foo;

Since:
   JAXB2.0
Author:
   Kohsuke Kawaguchi

---

Optional Element Summary

<table>
<thead>
<tr>
<th>boolean</th>
<th>lax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controls the unmarshaller behavior when it sees elements known to the current JAXBContext.</td>
<td></td>
</tr>
</tbody>
</table>
abstract public boolean lax()

false

false DOM DOM

ture

true JAXBContext XmlAnyElement XmlRootElement

XmlElementDecl unmarshaller JAXB DOM xsi:type xsi:type JAXB JAXBElem entraller

JAXBElem elem

DOM JAXB

W3C XML "lax"

lax

public abstract boolean lax

Controls the unmarshaller behavior when it sees elements known to the current JAXBContext.

When false

If false, all the elements that match the property will be unmarshalled to DOM, and the property will only contain DOM elements.

When true
If true, when an element matches a property marked with 
XmlAnyElement is known to JAXBContext (for example, there's a class 
with XmlRootElement that has the same tag name, or there's 
XmlElementDecl that has the same tag name), the unmarshaller will 
eagerly unmarshal this element to the JAXB object, instead of 
unmarshalling it to DOM. Additionally, if the element is unknown but 
it has a known xsi:type, the unmarshaller eagerly unmarshals the 
element to a JAXBElement, with the unknown element name and the 
JAXBElement value is set to an instance of the JAXB mapping of the 
known xsi:type.

As a result, after the unmarshalling, the property can become 
heterogeneous; it can have both DOM nodes and some JAXB 
objects at the same time.

This can be used to emulate the "lax" wildcard semantics of the W3C 
XML Schema.

**Default:**
false

---

abstract public Class<T> value()

**DomHandler** XML  DOM

value

public abstract Class<? extends DomHandler> value

Specifies the DomHandler which is responsible for actually converting 
XML from/to a DOM-like data structure.

**Default:**
javax.xml.bind.annotation.W3CDomHandler.class
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  Annotation Type
XmlAttachmentRef

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})
public @interface XmlAttachmentRef

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})

/ XML  MIME  URI  MIME out-of-line
DataHandler
since JAXB2.0

Marks a field/property that its XML form is a uri reference to mime
The mime content is optimally stored out-of-line as an attachment.

A field/property must always map to the DataHandler class.
Usage

```java
@XmlElement
class Foo {
    @XmlAttribute
    @XmlAttachmentRef
    DataHandler data;

    @XmlAttachmentRef
    @XmlElement
    DataHandler body;
}
```

The above code maps to the following XML:

```xml
<xs:element name="foo" xmlns:ref="http://ws-i.org/profiles/basic">
    <xs:complexType>
        <xs:sequence>
        </xs:sequence></xs:complexType></xs:element><xs:element name="body" type="ref:swaRef" minOccurs="0">
    <xs:attribute name="data" type="ref:swaRef" use="optional">
    </xs:attribute></xs:element>
```

The above binding supports WS-I AP 1.0 [WS-I Attachments Profile Version 1.0](https://www.ws-i.org). Since:

JAXB 2.0
Author:

Kohsuke Kawaguchi

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is su

PS:
javax.xml.bind.annotation  Annotation Type XmlAttribute

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlAttribute

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean XML

@XmlAttribute

- JavaBean
- static final XML fixed

javax.xml.bind.package javadoc ""

- // Examples
  @XmlAttribute List<Integer> items; //legal
  @XmlAttribute List<Bar> foo; // illegal if Bar does not map

- // Examples
  @XmlAttribute int foo; // legal
  @XmlAttribute Foo foo; // illegal if Foo does not map to a
1 JavaBean XML

// Example: Code fragment
public class USPrice {
    @XmlAttribute
    public java.math.BigDecimal getPrice() {...};
    public void setPrice(java.math.BigDecimal ) {...};
}

<!-- Example: XML Schema fragment -->
<xs:complexType name="USPrice">
    <xs:sequence/>
    <xs:attribute name="price" type="xs:decimal"/>
</xs:complexType>

2 JavaBean XML

@ XmlType

3 JavaBean XML

// Example: Code fragment
class Foo {
    ...
    @XmlAttribute List<Integer> items;
}

<!-- Example: XML Schema fragment -->
<xs:complexType name="foo">
    ...
    <xs:attribute name="items">
        <xs:simpleType>
            <xs:list itemType="xs:int"/>
        </xs:simpleType>
    </xs:complexType>
</xs:complexType>
See also [javax.xml.bind.annotation.XmlType](https://docs.oracle.com/javadocs/api/java/beans/property/javax/xml/bind/annotation/XmlType.html)

Maps a JavaBean property to a XML attribute.

**Usage**

The `@XmlAttribute` annotation can be used with the following program elements:

- JavaBean property
- field

A static final field is mapped to a XML fixed attribute.


The usage is subject to the following constraints:

- If type of the field or the property is a collection type, then the collection item type must be mapped to schema simple type.

  // Examples
  @XmlAttribute List<Integer> items; //legal
  @XmlAttribute List<Bar> foo; // illegal if Bar does not map

- If the type of the field or the property is a non collection type, then the type of the property or field must map to a simple schema type.

  // Examples
  @XmlAttribute int foo; // legal
  @XmlAttribute Foo foo; // illegal if Foo does not map to a

- This annotation can be used with the following annotations: `XmlID`, `XmlIDREF`, `XmlList`, `XmlSchemaType`, `XmlValue`, `XmlAttachmentRef`, `XmlMimeType`, `XmlInlineBinaryData`, `XmlJavaTypeAdapter`.

**Example 1:** Map a JavaBean property to an XML attribute.
//Example: Code fragment
public class USPrice {
    @XmlAttribute
    public java.math.BigDecimal getPrice() {...};
    public void setPrice(java.math.BigDecimal ) {...};
}
<!-- Example: XML Schema fragment -->
<xs:complexType name="USPrice">
    <xs:sequence>
        <xs:attribute name="price" type="xs:decimal"/>
    </xs:sequence>
</xs:complexType>

Example 2: Map a JavaBean property to an XML attribute with anonymous type.

See Example 7 in @XmlType.

Example 3: Map a JavaBean collection property to an XML attribute.

    // Example: Code fragment
    class Foo {
        ...
        @XmlAttribute List<Integer> items;
    }

    <!-- Example: XML Schema fragment -->
    <xs:complexType name="foo">
        ...
        <xs:attribute name="items">
            <xs:simpleType>
                <xs:list itemType="xs:int"/>
            </xs:simpleType>
        </xs:attribute>
    </xs:complexType>

Since:
    JAXB2.0
Version:
    $Revision: 1.14 $
Author:
    Sekhar Vajjhala, Sun Microsystems, Inc.
See Also:
**Optional Element Summary**

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>Name of the XML Schema attribute.</td>
</tr>
<tr>
<td>String</td>
<td>namespace</td>
<td>Specifies the XML target namespace of the XML Schema attribute.</td>
</tr>
<tr>
<td>boolean</td>
<td>required</td>
<td>Specifies if the XML Schema attribute is optional or required.</td>
</tr>
</tbody>
</table>

**abstract public String name()**

Name of the XML Schema attribute. By default, the XML Schema attribute name is derived from the JavaBean property name.

**Default:**
"##default"

**abstract public boolean required()**

Name of the XML Schema attribute. By default, the XML Schema attribute name is derived from the JavaBean property name.
public abstract boolean required

Specifies if the XML Schema attribute is optional or required. If true, then the JavaBean property is mapped to a XML Schema attribute that is required. Otherwise it is mapped to a XML Schema attribute that is optional.

Default:
false

abstract public String namespace()
XML XML

namespace

public abstract String namespace

Specifies the XML target namespace of the XML Schema attribute.

Default:
"##default"

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.bind.annotation  Annotation Type XmlElement

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public@interface XmlElement

Implements: Annotation
Inner classes: XmlElement.DEFAULT
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean XML

@XmlElement

- JavaBean
- 
- XmlElements

- XmlIDXmlIDREFXmlListXmlSchemaTypeXmlValueXmlAttachmentRefXmlMime
  XmlJavaTypeAdapter
- JavaBean collection (indexed)
  @XmlElement.type() collection DEFAULT.class

@XmlElement JavaBean XML XML

1  public static final

    //Example: Code fragment
    public class USPrice {
        @XmlElement(name="itemprice")
        public java.math.BigDecimal price;
    }

    <!-- Example: Local XML Schema element -->
2 nillable

//Example: Code fragment
public class USPrice {
    @XmlElement(nillable=true)
    public java.math.BigDecimal price;
}

<!-- Example: Local XML Schema element -->
<x:s:complexType name="USPrice">
    <xs:sequence>
        <xs:element name="price" type="xs:decimal" nillable="true" />
    </sequence>
</xs:complexType>

3 nillable_required

//Example: Code fragment
public class USPrice {
    @XmlElement(nillable=true, required=true)
    public java.math.BigDecimal price;
}

<!-- Example: Local XML Schema element -->
<x:s:complexType name="USPrice">
    <xs:sequence>
        <xs:element name="price" type="xs:decimal" nillable="true" />
    </sequence>
</xs:complexType>

4 JavaBean XML

@  

since  JAXB2.0

version  $Revision: 1.19 $
Maps a JavaBean property to a XML element derived from property name.

Usage

@XmlElement annotation can be used with the following program elements:

- a JavaBean property
- non static, non transient field
- within XmlElements

The usage is subject to the following constraints:

- This annotation can be used with following annotations: XmlID, XmlIDREF, XmlList, XmlSchemaType, XmlValue, XmlAttachmentRef, XmlMimeType, XmlInlineBinaryData, XmlElementWrapper, XmlJavaTypeAdapter
- if the type of JavaBean property is a collection type of array, an indexed property, or a parameterized list, and this annotation is used with XmlElements then, @XmlElement.type() must be DEFAULT.class since the collection item type is already known.

A JavaBean property, when annotated with @XmlElement annotation is mapped to a local element in the XML Schema complex type to which the containing class is mapped.

**Example 1:** Map a public non static non final field to local element

```java
//Example: Code fragment
class USPrice {
    @XmlElement(name="itemprice")
    public java.math.BigDecimal price;
}

<!-- Example: Local XML Schema element -->
<xs:complexType name="USPrice"/>
<xs:sequence>
    <xs:element name="itemprice" type="xs:decimal" minOccurs="0"
</sequence>
```
Example 2: Map a field to a nillable element.

```
//Example: Code fragment
public class USPrice {
    @XmlElement(nillable=true)
    public java.math.BigDecimal price;
}
```

<!-- Example: Local XML Schema element -->
```
<xs:complexType name="USPrice">
    <xs:sequence>
        <xs:element name="price" type="xs:decimal" nillable="true"/>
    </sequence>
</xs:complexType>
```

Example 3: Map a field to a nillable, required element.

```
//Example: Code fragment
public class USPrice {
    @XmlElement(nillable=true, required=true)
    public java.math.BigDecimal price;
}
```

<!-- Example: Local XML Schema element -->
```
<xs:complexType name="USPrice">
    <xs:sequence>
        <xs:element name="price" type="xs:decimal" nillable="true"/>
    </sequence>
</xs:complexType>
```

Example 4: Map a JavaBean property to an XML element with anonymous type.

See Example 6 in @XmlType.

Since: JAXB2.0

Version: $Revision: 1.19$

Author: Sekhar Vajjhala, Sun Microsystems, Inc.
### Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>Default Value</th>
<th>Default value of this element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>Name of the XML Schema element.</td>
</tr>
<tr>
<td>String</td>
<td>namespace</td>
<td>XML target namespace of the XML Schema element.</td>
</tr>
<tr>
<td>boolean</td>
<td>nillable</td>
<td>Customize the element declaration to be nillable.</td>
</tr>
<tr>
<td>boolean</td>
<td>required</td>
<td>Customize the element declaration to be required.</td>
</tr>
<tr>
<td>Class</td>
<td>type</td>
<td>The Java class being referenced.</td>
</tr>
</tbody>
</table>

abstract public String name()
XML

"##default" JavaBean

**name**

public abstract String name

    Name of the XML Schema element.

    If the value is "##default", then element name is derived from the JavaBean property name.

**Default:**

"##default"
abstract public boolean nillable()
nillable()
	nillable()  true  JavaBean  XML  nillable

nillable

public abstract boolean nillable

    Customize the element declaration to be nillable.
    
    If nillable() is true, then the JavaBean property is mapped to a XML Schema nillable element declaration.

    Default:
    false

abstract public boolean required()

required()  true  JavaBean  minOccurs="1"  XML
maxOccurs  "1"  maxOccurs  "unbounded"

required()  false  JavaBean  minOccurs="0"
XML  maxOccurs  "1"  maxOccurs
"unbounded"

required

public abstract boolean required

    Customize the element declaration to be required.
If required() is true, then JavaBean property is mapped to an XML schema element declaration with minOccurs="1". maxOccurs is "1" for a single valued property and "unbounded" for a multivalued property.

If required() is false, then the JavaBean property is mapped to XML Schema element declaration with minOccurs="0". maxOccurs is "1" for a single valued property and "unbounded" for a multivalued property.

Default: false

abstract public String namespace()
XML  XML

"##default"

1. XmlSchema  elementFormDefault  QUALIFIED
2. ""

namespace

public abstract String namespace

XML target namespace of the XML Schema element.

If the value is "##default", then the namespace is determined as follows:

1. If the enclosing package has XmlSchema annotation, and its elementFormDefault is QUALIFIED, then the namespace of the enclosing class.
2. Otherwise "" (which produces unqualified element in the default namespace.)
Abstract public String defaultValue()

" null 'no default value'

defaultValue

Public abstract String defaultValue

Default value of this element.

The " value specified as a default of this annotation element is used as a poor-man's substitute for null to allow implementations to recognize the 'no default value' state.

Default:

"\u0000"

Abstract public Class<T> type()

Java

type

Public abstract Class type

The Java class being referenced.

Default:

javax.xml.bind.annotation.XmlElement.DEFAULT.class
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
Class XmlElement.DEFAULT

public static final class XmlElement.DEFAULT extends Object

Contained within: XmlElement

type()

Used in XmlElement.type() to signal that the type be inferred from the signature of the property.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlElement.DEFAULT()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object</td>
</tr>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>

Constructor Detail
public XmlElement.DEFAULT()

XmlElement.DEFAULT

public XmlElement.DEFAULT()
javax.xml.bind.annotation Annotation Type XmlElementDecl

@Retention(value=RUNTIME)
@Target(value=METHOD)
public @interface XmlElementDecl

**Implements**: Annotation
**Inner classes**: XmlElementDecl.GLOBAL

@Retention(value=RUNTIME)
@Target(value=METHOD)

XML

XML JAXBEElement Java ObjectFactory

@XmlElementDecl

- @XmlElementDecl
- Object

1

// Example: code fragment
@XmlRegistry
class ObjectFactory {
    @XmlElementDecl(name="foo")
    JAXBElement<String> createFoo(String s) { ... }
}

&lt;!-- XML input --&gt;
&lt;foo&gt;string&lt;/foo&gt;

// Example: code fragment corresponding to XML input
JAXBElement&lt;String&gt; o =
(JAXBElement&lt;String&gt;&lt;/string&gt;&lt;/string&gt;)unmarshaller.unmarshal(aboveDocument);
// print JAXBElement instance to show values
System.out.println(o.getName()); // prints "{}foo"
System.out.println(o.getValue()); // prints "string"
System.out.println(o.getValue().getClass()); // prints "java.la
&lt;!-- Example: XML schema definition --&gt;
&lt;xs:element name="foo" type="xs:string"&gt;
&lt;/xs:element&gt;&lt;/string&gt;

javadoc

&lt;!-- Example: XML schema definition --&gt;
&lt;xs:schema&gt;
&lt;xs:complexType name="pea">
 &lt;xs:choice maxOccurs="unbounded">
    &lt;xs:element name="foo" type="xs:string"&gt;
    &lt;/xs:element&gt;
    &lt;xs:element name="bar" type="xs:string"&gt;
    &lt;/xs:element&gt;
 &lt;/xs:choice&gt;
 &lt;/xs:complexType&gt;
 &lt;xs:element name="foo" type="xs:int"&gt;
 &lt;/xs:element&gt;&lt;/xs:schema&gt;

// Example: expected default binding
class Pea {
    @XmlElementRefs({
        @XmlElementRef(name="foo",type=JAXBElement.class)
        @XmlElementRef(name="bar",type=JAXBElement.class)
    })
    List<JAXBElement<String>> fooOrBar;
}

@XmlRegistry
class ObjectFactory {
    @XmlElementDecl(scope=Pea.class,name="foo")
    JAXBElement createPeaFoo(String s);

    @XmlElementDecl(scope=Pea.class,name="bar")
    JAXBElement createPeaBar(String s);

    @XmlElementDecl(name="foo")
    JAXBElement createFoo(Integer i);
}

createFoo createPeaFoo "foo" XML
Maps a factory method to a XML element.

Usage

The annotation creates a mapping between an XML schema element declaration and a *element factory method* that returns a JAXBElemnt instance representing the element declaration. Typically, the element factory method is generated (and annotated) from a schema into the ObjectFactory class in a Java package that represents the binding of the element declaration's target namespace. Thus, while the annotation syntax allows `@XmlElementDecl` to be used on any method, semantically its use is restricted to annotation of element factory method. The usage is subject to the following constraints:

- The class containing the element factory method annotated with `@XmlElementDecl` must be marked with `XmlRegistry`.
- The element factory method must take one parameter assignable to `Object`.

**Example 1: Annotation on a factory method**

```java
// Example: code fragment
@XmlRegistry
class ObjectFactory {
    @XmlElementDecl(name="foo")
    JAXBElement<String> createFoo(String s) { ... }
}

&lt;!-- XML input --&gt;
    &lt;foo&gt;string&lt;/foo&gt;

// Example: code fragment corresponding to XML input
JAXBElement&lt;String&gt; o =
    (JAXBElement&lt;/string&gt;&lt;string&gt;)unmarshaller.unmarshal(aboveDocument);
// print JAXBElement instance to show values
System.out.println(o.getName()); // prints "{}foo"
System.out.println(o.getValue()); // prints "string"
```
System.out.println(o.getValue().getClass()); // prints "java.lang.String"

<!-- Example: XML schema definition --&gt;
&lt;xs:element name="foo" type="xs:string"&gt;
&lt;/xs:element&gt;&lt;/string&gt;

**Example 2: Element declaration with non local scope**

The following example illustrates the use of scope annotation parameter in binding of element declaration in schema derived code.

The following example may be replaced in a future revision of this javadoc.

```xml
&lt;xs:element name="foo" type="xs:int"&gt;
&lt;/xs:element&gt;

&lt;xs:element name="foo" type="xs:string"&gt;
&lt;/xs:element&gt;

&lt;xs:complexType name="pea">
  &lt;xs:choice maxOccurs="unbounded">
    &lt;xs:element name="foo" type="xs:string"&gt;
    &lt;/xs:element&gt;
    &lt;xs:element name="bar" type="xs:string"&gt;
    &lt;/xs:element&gt;
  &lt;/xs:choice&gt;
&lt;/xs:complexType&gt;

// Example: expected default binding
class Pea {
  @XmlElementRefs(
    @XmlElementRef(name="foo",type=JAXBElement.class)
    @XmlElementRef(name="bar",type=JAXBElement.class)
  )
  List<JAXBElement<String>> fooOrBar;
}

@XmlRegistry
class ObjectFactory {
  @XmlElementDecl(scope=Pea.class,name="foo")
  JAXBElement createPeaFoo(String s);

  @XmlElementDecl(scope=Pea.class,name="bar")
  JAXBElement createPeaBar(String s);

  @XmlElementDecl(name="foo")
  JAXBElement createFoo(Integer i);
}
Without scope createFoo and createPeaFoo would become ambiguous since both of them map to an XML schema element with the same local name "foo".

Since:
JAXB 2.0
See Also:
XmlRegistry

---

### Required Element Summary

| String name | local name of the XML element. |

### Optional Element Summary

| String defaultValue | Default value of this element. |
| String namespace | namespace name of the XML element. |
| Class scope | scope of the mapping. |
| String substitutionHeadName | XML local name of a substitution group's head element. |
| String substitutionHeadNamespace | namespace name of a substitution group's head XML element. |

### Element Detail

abstract public String name()
XML
See also

name

public abstract String name

local name of the XML element.

**Note to reviewers:** There is no default name; since the annotation is on a factory method, it is not clear that the method name can be derived from the factory method name.

See Also:

namespace()

abstract public Class<T> scope()

**XmlElementDecl.GLOBAL**

scope

public abstract Class scope

scope of the mapping.

If this is not XmlElementDecl.GLOBAL, then this element declaration mapping is only active within the specified class.

**Default:**

javax.xml.bind.annotation.XmlElementDecl.GLOBAL.class
abstract public String namespace()

"##default"

See also name()

namespace

public abstract String namespace

namespace name of the XML element.

If the value is "##default", then the value is the namespace name for the package of the class containing this factory method.

See Also:

name()

Default:

"##default"

-------------------------------

abstract public String substitutionHeadNamespace()

XML

substitutionHeadName() XML

substitutionHeadName() "" "##default"

substitutionHeadName() ""

substitutionHeadName() "" "##default"

XmlRegistry

substitutionHeadName() "" "##default"

See also substitutionHeadName()
substitutionHeadNamespace

public abstract String substitutionHeadNamespace

namespace name of a substitution group's head XML element.

This specifies the namespace name of the XML element whose local name is specified by substitutionHeadName().

If substitutionHeadName() is "", then this value can only be "##default". But the value is ignored since this element is not part of substitution group when the value of substitutionHeadName() is "".

If substitutionHeadName() is not "" and the value is "##default", then the namespace name is the namespace name to which the package of the containing class, marked with XmlRegistry, is mapped.

If substitutionHeadName() is not "" and the value is not "##default", then the value is the namespace name.

See Also:
substitutionHeadName()

Default:
"##default"

abstract public String substitutionHeadName()

XML

See also
substitutionHeadNamespace()
public abstract String substitutionHeadName

XML local name of a substitution group's head element.

If the value is "", then this element is not part of any substitution group.

See Also:
  substitutionHeadNamespace()

Default:
  ""

abstract public String defaultValue()

" null 'no default value'

defaultValue

public abstract String defaultValue

Default value of this element.

The " value specified as a default of this annotation element is used as a poor-man's substitute for null to allow implementations to recognize the 'no default value' state.

Default:
  "\u0000"
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  Class XmlElementDecl.GLOBAL

java.lang.Object
   ↓ javax.xml.bind.annotation.XmlElementDecl.GLOBAL

Enclosing class:
   XmlElementDecl

public static final class XmlElementDecl.GLOBAL

extends Object

Contained within: XmlElementDecl

scope()

Used in XmlElementDecl.scope() to signal that the declaration is in the global scope.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlElementDecl.GLOBAL()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object</td>
</tr>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>

Constructor Detail
public XmlElementDecl.GLOBAL()

XmlElementDecl.GLOBAL

class public XmlElementDecl.GLOBAL()

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.xml.bind.annotation Annotation Type XmlElementRef

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlElementRef

**Implements:** Annotation  
**Inner classes:** `XmlElementRef.DEFAULT`

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean XML

@XmlElementRef JavaBean  XmlElementRefs

XML JavaBean JavaBean  XmlElementRefs
JavaBean XML JavaBean

XML

XML XML XML

public void setTerm(JAXBElement<? extends Operator?>);
public JAXBElement<? extends Operator?> getTerm();

XmlElementDecl XML  JAXBElemment @XmlElementRef
JAXBElemment JavaBean XML

- collection collection
  
  `javax.xml.bind.JAXBElement` @XmlElementRef}.name()
  @XmlElementRef.name() @XmlRegistry
@XmlElementDecl  ObjectFactory
  ○ @XmlElementDecl.name()  @XmlElementRef.name()
  ○ @XmlElementDecl.namespace()
        @XmlElementRef.namespace()
• collection  collection

javax.xml.bind.JAXBElement  XmlRootElement
  XmlElementWrapper  XmlJavaTypeAdapter

javax.xml.bind.package  javadoc ""

1  Ant

Java  Ant  Ant  Ant  XML  @XmlRootElement

@XmlRootElement(name="target")
class Target {
    // The presence of @XmlElementRef indicates that the XML
    // element name will be derived from the @XmlRootElement
    // annotation on the type (for e.g. "jar" for JarTask).
    @XmlElementRef
    List<Task> tasks;
}

abstract class Task {
}

@XmlRootElement(name="jar")
class JarTask extends Task {
    ...
}

@XmlRootElement(name="javac")
class JavacTask extends Task {
    ...
}

<!-- XML Schema fragment -->
<xs:element name="target" type="Target">
<xs:complexType name="Target">
    <xs:sequence>
        <xs:choice maxOccurs="unbounded">
            <xs:element ref="jar">
                <xs:element ref="javac">
                    </xs:choice>
        </xs:sequence>
</xs:complexType>
Target target = new Target();
target.tasks.add(new JarTask());
target.tasks.add(new JavacTask());
marshal(target);

XML

    &lt;target&gt;
    &lt;jar&gt;
    ....
    &lt;/jar&gt;
    &lt;javac&gt;
    ....
    &lt;/javac&gt;
    &lt;/target&gt;

2XML

XML  ObjectFactory

    @XmlElement
    class Math {
        // The value of #type() is
        // JAXBElemen CLASS, which indicates the XML
        // element name ObjectFactory - in general a class marked
        // with @XmlRegistry. (See ObjectFactory below)
        //
        // The #name() is "operator", a pointer to a
        // factory method annotated with a
        // XmlElementDec1 with the name "operator". Since
        // "operator" is the head of a substitution group that
        // contains elements "add" and "sub" elements, "operator"
        // element can be substituted in an instance document by
        // elements "add" or "sub". At runtime, JAXBEmelent
        // instance contains the element name that has been
        // substituted in the XML document.
        //
        @XmlElementRef(type=JAXBElemen CLASS, name="operator")
        JAXBElemen &lt;? extends Operator&gt; term;
Math m = new Math();
m.term = new ObjectFactory().createAdd(new Operator());
marshal(m);

XML

<math>
<add>...</add>
</math>

since JAXB2.0

See also javax.xml.bind.annotation.XmlElementRefs

Maps a JavaBean property to a XML element derived from property's type.

Usage

@XmlElementRef annotation can be used with a JavaBean property or from within XmlElementRefs

This annotation dynamically associates an XML element name with the JavaBean property. When a JavaBean property is annotated with
XmlElement, the XML element name is statically derived from the JavaBean property name. However, when this annotation is used, the XML element name is derived from the instance of the type of the JavaBean property at runtime.

**XML Schema substitution group support**

XML Schema allows a XML document author to use XML element names that were not statically specified in the content model of a schema using substitution groups. Schema derived code provides support for substitution groups using an *element property*, (section 5.5.5, "Element Property" of JAXB 2.0 specification). An element property method signature is of the form:

```java
public void setTerm(JAXBElement<? extends Operator?>);
public JAXBElement<? extends Operator?> getTerm();
```

An element factory method annotated with `XmlElementDecl` is used to create a `JAXBElement` instance, containing an XML element name. The presence of `@XmlElementRef` annotation on an element property indicates that the element name from `JAXBElement` instance be used instead of deriving an XML element name from the JavaBean property name.

The usage is subject to the following constraints:

- If the collection item type (for collection property) or property type (for single valued property) is `JAXBElement`, then `@XmlElementRef`.name() and `@XmlElementRef.namespace()` must point to an element factory method with an `@XmlElementDecl` annotation in a class annotated with `@XmlRegistry` (usually ObjectFactory class generated by the schema compiler):
  - `@XmlElementDecl.name()` must equal `@XmlElementRef.name()`
  - `@XmlElementDecl.namespace()` must equal `@XmlElementRef.namespace()`.
- If the collection item type (for collection property) or property type (for single valued property) is not `JAXBElement`, then the type...
referenced by the property or field must be annotated with
\texttt{XmlRootElement}.

- This annotation can be used with the following annotations:
  \texttt{XmlElementWrapper, XmlJavaTypeAdapter}.

See "Package Specification" in \texttt{javax.xml.bind.package javadoc} for additional common information.

\textbf{Example 1: Ant Task Example}

The following Java class hierarchy models an Ant build script. An Ant task corresponds to a class in the class hierarchy. The XML element name of an Ant task is indicated by the \texttt{@XmlRootElement} annotation on its corresponding class.

```java
@XmlElement(name="target")
class Target {
    // The presence of @XmlElementRef indicates that the XML
    // element name will be derived from the @XmlRootElement
    // annotation on the type (for e.g. "jar" for JarTask).
    @XmlElementRef
    List<Task> tasks;
}

abstract class Task {
}

@XmlElement(name="jar")
class JarTask extends Task {
    ...
}

@XmlElement(name="javac")
class JavacTask extends Task {
    ...
}

<xs:element name="target" type="Target">
    <xs:complexType name="Target">
        <xs:sequence>
            <xs:choice maxOccurs="unbounded">
                <xs:element ref="jar">
                <xs:element ref="javac">
            </xs:choice>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```
Thus the following code fragment:

```java
Target target = new Target();
target.tasks.add(new JarTask());
target.tasks.add(new JavacTask());
marshal(target);
```

will produce the following XML output:

```xml
&lt;target&gt;
  &lt;jar&gt;
    ....
  &lt;/jar&gt;
  &lt;javac&gt;
    ....
  &lt;/javac&gt;
&lt;/target&gt;
```

It is not an error to have a class that extends `Task` that doesn't have `XmlRootElement`. But they can't show up in an XML instance (because they don't have XML element names).

**Example 2: XML Schema Substitution group support**

The following example shows the annotations for XML Schema substitution groups. The annotations and the ObjectFactory are derived from the schema.

```java
@XmlElement
class Math {
    // The value of `type()` is
    // JAXBElement.class, which indicates the XML
    // element name ObjectFactory - in general a class marked
    // with @XmlRegistry. (See ObjectFactory below)
    //
    // The `name()` is "operator", a pointer to a
    // factory method annotated with a
    // `XmlElementDecl` with the name "operator". Since
    // "operator" is the head of a substitution group that
    // contains elements "add" and "sub" elements, "operator"
```
// element can be substituted in an instance document by
// elements "add" or "sub". At runtime, JAXBElement
// instance contains the element name that has been
// substituted in the XML document.

@XmlElementRef(type=JAXBElement.class,name="operator")
JAXBElement<? extends Operator> term;

@XmlRegistry
class ObjectFactory {
    @XmlElementDecl(name="operator")
    JAXBElement<Operator> createOperator(Operator o) {...}
    @XmlElementDecl(name="add",substitutionHeadName="operator")
    JAXBElement<Operator> createAdd(Operator o) {...}
    @XmlElementDecl(name="sub",substitutionHeadName="operator")
    JAXBElement<Operator> createSub(Operator o) {...}
}

class Operator {
    ...
}

Thus, the following code fragment

Math m = new Math();
m.term = new ObjectFactory().createAdd(new Operator());
marshal(m);

will produce the following XML output:

<math>
    <add>...</add>
</math>

Since:
JAXB2.0
Author:
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Sekhar Vajjhala, Sun Microsystems, Inc.
See Also:
XmlElementRefs
Optional Element Summary

<table>
<thead>
<tr>
<th>String name</th>
<th>This parameter and name() are used to determine the XML element for the JavaBean property.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String namespace</td>
<td></td>
</tr>
<tr>
<td>Class type</td>
<td>The Java type being referenced.</td>
</tr>
</tbody>
</table>

abstract public Class<T> type()
Java

DEFAULT.class JavaBean

type

public abstract Class type

The Java type being referenced.

If the value is DEFAULT.class, the type is inferred from the the type of the JavaBean property.

Default:
javax.xml.bind.annotation.XmlElementRef.DEFAULT.class

abstract public String namespace()

#name() JavaBean XML

type() JAXBElement.class namespace() name() XmlElementDecl XML XML XmlElementDecl
XmlElementDecl
namespace

public abstract String namespace

This parameter and name() are used to determine the XML element for the JavaBean property.

If type() is JAXBElement.class, then namespace() and name() point to a factory method with XmlElementDecl. The XML element name is the element name from the factory method's XmlElementDecl annotation or if an element from its substitution group (of which it is a head element) has been substituted in the XML document, then the element name is from the XmlElementDecl on the substituted element.

If type() is not JAXBElement.class, then the XML element name is the XML element name statically associated with the type using the annotation XmlRootElement on the type. If the type is not annotated with an XmlElementDecl, then it is an error.

If type() is not JAXBElement.class, then this value must be "".

Default: ""

abstract public String name()

See also namespace()
public abstract String name

See Also:
namespace()

Default:
"##default"
javax.xml.bind.annotation  Class XmlElementRef.DEFAULT

java.lang.Object
  javax.xml.bind.annotation.XmlElementRef.DEFAULT

Enclosing class:
    XmlElementRef

public static final class XmlElementRef.DEFAULT
  extends Object

Contained within: XmlElementRef

    type()

Used in XmlElementRef.type() to signal that the type be inferred from the signature of the property.

---

Constructor Summary

<table>
<thead>
<tr>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlElementRef.DEFAULT()</td>
</tr>
</tbody>
</table>

---

Method Summary

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

---

Constructor Detail
public XmlElementRef.DEFAULT()

XmlElementRef.DEFAULT

public XmlElementRef.DEFAULT()
javax.xml.bind.annotation  Annotation Type XmlElementRefs

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public interface XmlElementRefs

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

XmlElement JAXBElement

XmlElement @xsi:type

XmlJavaTypeAdapter XmlElementWrapper

since JAXB2.0

See javax.xml.bind.annotation.XmlElementWrapper,
also javax.xml.bind.annotation.XmlElementRef

Marks a property that refers to classes with XmlElement or JAXBElement.

Compared to an element property (property with XmlElement annotation),
a reference property has a different substitution semantics. When a sub-class
is assigned to a property, an element property produces the same
tag name with @xsi:type, whereas a reference property produces a
different tag name (the tag name that's on the the sub-class.)

This annotation can be used with the following annotations:
XmlJavaTypeAdapter, XmlElementWrapper.

**Since:**
JAXB2.0

**Author:**
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Sekhar Vajjhala, Sun Microsystems, Inc.

**See Also:**
Required Element Summary

| XmlElementRef[] | value |

Element Detail

abstract public XmlElementRef[] value()

value

public abstract XmlElementRef[] value

Overview Package Tree Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation Annotation Type XmlElements

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlElements

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

@ XmlElement	@ Xmlelements
(XmlElements{{ @XmlElement(...),@XmlElement(...) }})

@XMLElements

- JavaBean
- static transient

JavaBean

- @ XMLIDREF	@ XMLElementWrapper
- @XMLIDREF JavaBean @XmlElement.type()
  @XmliD JavaBean

javax.xml.bind.package javadoc ""

1

// Mapped code fragment
public class Foo {
    @XmlElements(
```java
public List items;

<!-- XML Representation for a List of {1,2.5} -->
... 1
  2.5
...

<xs:complexType name="Foo">
  <xs:sequence>
    <xs:choice minOccurs="0" maxOccurs="unbounded">
      <xs:element name="A" type="xs:int"/>
      <xs:element name="B" type="xs:float"/>
    </xs:choice>
  </xs:sequence>
</xs:complexType>

// Mapped code fragment
public class Foo {
  @XmlElementWrapper(name="bar")
  @XmlElements(
    @XmlElement(name="A", type=Integer.class),
    @XmlElement(name="B", type=Float.class)
  )
  public List items;
}

<xs:complexType name="Foo">
  <xs:sequence>
    <xs:element name="bar">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="A" type="xs:int"/>
          <xs:element name="B" type="xs:float"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```
class Foo {
    @XmlJavaTypeAdapter(QtoPAdapter.class)
    @XmlElements(
        @XmlElement(name="A",type=PX.class),
        @XmlElement(name="B",type=PY.class)
    )
    Q bar;
}

@XmlType abstract class P {...}
@XmlType(name="PX") class PX extends P {...}
@XmlType(name="PY") class PY extends P {...}

<!-- XML Schema fragment -->
<xs:complexType name="Foo">
    <xs:sequence>
        <xs:element name="bar">
            <xs:complexType>
                <xs:choice minOccurs="0" maxOccurs="unbounded">
                    <xs:element name="A" type="PX"/>
                    <xs:element name="B" type="PY"/>
                </xs:choice>
            </xs:complexType>
        </xs:element>
    </xs:sequence>
</xs:complexType>

since JAXB2.0

javax.xml.bind.annotation.XmlElement,
See javax.xml.bind.annotation.XmlElementRef,
also javax.xml.bind.annotation.XmlElementRefs,
javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter

A container for multiple @XmlElement annotations. Multiple annotations of
the same type are not allowed on a program element. This annotation
therefore serves as a container annotation for multiple @XmlElements as
follows:
The `@XmlElements` annotation can be used with the following program elements:

- a JavaBean property
- non static, non transient field

This annotation is intended for annotation a JavaBean collection property (e.g. List).

**Usage**

The usage is subject to the following constraints:

- This annotation can be used with the following annotations: `@XmlIDREF, @XmlElementWrapper`.
- If `@XmlIDREF` is also specified on the JavaBean property, then each `@XmlElement.type()` must contain a JavaBean property annotated with `@XmlID`.


---

**Example 1:** Map to a list of elements

```java
// Mapped code fragment
public class Foo {
    @XmlElements(
        @XmlElement(name="A", type=Integer.class),
        @XmlElement(name="B", type=Float.class)
    )
    public List items;
}
```

XML output is not wrapped using another element -->

```
... 1 2.5 ...
```
Example 2: Map to a list of elements wrapped with another element

```java
// Mapped code fragment
public class Foo {
    @XmlElementWrapper(name="bar")
    @XmlElements(
        @XmlElement(name="A", type=Integer.class),
        @XmlElement(name="B", type=Float.class)
    )
    public List items;
}
```

Example 3: Change element name based on type using an adapter.

class Foo {
    @XmlJavaTypeAdapter(QtoPAdapter.class)
    @XmlElements({
        @XmlElement(name="A", type=PX.class),
        @XmlElement(name="B", type=PY.class)
    })
}
Q bar;
}

@XmlType abstract class P {...}
@XmlType(name="PX") class PX extends P {...}
@XmlType(name="PY") class PY extends P {...}

<xs:complexType name="Foo">
  <xs:sequence>
    <xs:element name="bar">
      <xs:complexType>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
          <xs:element name="A" type="PX"/>
          <xs:element name="B" type="PY"/>
        </xs:choice>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

Since: JAXB2.0

Author:
  - Kohsuke Kawaguchi, Sun Microsystems, Inc.
  - Sekhar Vajjhala, Sun Microsystems, Inc.

See Also: 
  XElement, XmlElementRef, XmlElementRefs, XmlJavaTypeAdapter

Required Element Summary

<table>
<thead>
<tr>
<th>XElement[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection of @XmlElement annotations</td>
<td></td>
</tr>
</tbody>
</table>

Element Detail

abstract public XElement[] value()
@XmlElement
public abstract XmlElement[] value

Collection of @XmlElement annotations
javax.xml.bind.annotation  Annotation Type
XmlElementWrapper

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlElementWrapper

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

```
XML
//Example:code fragment
int[] names;

// XML Serialization Form 1 (Unwrapped collection)
<names> ...</names>
<names> ...</names>

// XML Serialization Form 2 (Wrapped collection)
<wrapperElement>
<names> value-of-item </names>
<names> value-of-item </names>
....
</wrapperElement>
```

nillable XML null

@XmlElementWrapper

- JavaBean
- static transient
Generates a wrapper element around XML representation. This is primarily intended to be used to produce a wrapper XML element around collections. The annotation therefore supports two forms of serialization shown below.

```java
//Example: code fragment
int[] names;

// XML Serialization Form 1 (Unwrapped collection)
<names> ... </names>
<names> ... </names>

// XML Serialization Form 2 (Wrapped collection )
<wrapperElement>
    <names> value-of-item </names>
    <names> value-of-item </names>
    ....
</wrapperElement>
```

The two serialized XML forms allow a null collection to be represented either by absence or presence of an element with a nillable attribute.

**Usage**

The `@XmlElementWrapper` annotation can be used with the following program elements:

- JavaBean property
- non static, non transient field
The usage is subject to the following constraints:

- The property must be a collection property
- This annotation can be used with the following annotations:
  - `XmlElement`, `XmlElements`, `XmlElementRef`, `XmlElementRefs`, `XmlJavaTypeAdapter`


Since:
- JAXB2.0

Author:
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:
- `XmlElement`, `XmlElements`, `XmlElementRef`, `XmlElementRefs`

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>String</td>
<td>name</td>
<td>Name of the XML wrapper element.</td>
</tr>
<tr>
<td>String</td>
<td>namespace</td>
<td>XML target namespace of the XML wrapper element.</td>
</tr>
<tr>
<td>boolean</td>
<td>nillable</td>
<td>If true, the absence of the collection is represented by using xsi:nil='true'.</td>
</tr>
</tbody>
</table>

abstract public String name()

XML XML JavaBean

name

public abstract String name
Name of the XML wrapper element. By default, the XML wrapper element name is derived from the JavaBean property name.

Default:
"##default"

---

abstract public String namespace()

XML namespace

"##default"

1. XmlSchema elementFormDefault QUALIFIED
2. ""

---

namespace

public abstract String namespace

XML target namespace of the XML wrapper element.

If the value is "##default", then the namespace is determined as follows:

1. If the enclosing package has XmlSchema annotation, and its elementFormDefault is QUALIFIED, then the namespace of the enclosing class.
2. Otherwise "" (which produces unqualified element in the default namespace.

Default:
"##default"

---

abstract public boolean nillable()

true xsi:nil='true'
nillable

public abstract boolean nillable

If true, the absence of the collection is represented by using xsi:nil='true'. Otherwise, it is represented by the absence of the element.

Default:
false
javax.xml.bind.annotation Annotation Type XmlEnum

@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface XmlEnum

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value=TYPE)

Enum XML

XmlEnumValue XML

@XmlEnum

@Retention(value=RUNTIME)
@Target(value=TYPE)

Enum XML

@XmlEnumValue

since JAXB2.0

Maps an enum type Enum to XML representation.

This annotation, together with XmlEnumValue provides a mapping of enum type to XML representation.
Usage

The @XmlEnum annotation can be used with the following program elements:

- enum type

The usage is subject to the following constraints:

- This annotation can be used the following other annotations: XmlType, XmlRootElement

See "Package Specification" in javax.xml.bind.package javadoc for additional common information

An enum type is mapped to a schema simple type with enumeration facets. The schema type is derived from the Java type to which @XmlEnum.value(). Each enum constant @XmlEnumValue must have a valid lexical representation for the type @XmlEnum.value().

Examples: See examples in XmlEnumValue

Since: JAXB2.0

---

Optional Element Summary

<table>
<thead>
<tr>
<th>Class&lt;?</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Java type that is mapped to a XML simple type.</td>
</tr>
</tbody>
</table>

abstract public Class<T> value()
XML  Java

value

public abstract Class<?> value
Java type that is mapped to a XML simple type.

**Default:**

java.lang.String.class

---

PS:
javax.xml.bind.annotation  Annotation Type XmlEnumValue

@Retention(value=RUNTIME)
@Target(value=FIELD)
public @interface XmlEnumValue

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=FIELD)

Enum  XML

@XmlEnumValue

javax.xml.bind.package javadoc ""

XmlEnum  XML

documentation facet  @XmlEnum.value()  Java
@XmlEnumValue  @XmlEnum.value()

name()  XML

1  ->  enumeration facet

//Example: Code fragment
@XmlEnum(String.class)
public enum Card { CLUBS, DIAMONDS, HEARTS, SPADES }

<!-- Example: XML Schema fragment -->
<xs:simpleType name="Card">
  <xs:restriction base="xs:string"/>
  <xs:enumeration value="CLUBS"/>
  <xs:enumeration value="DIAMONDS"/>
  <xs:enumeration value="HEARTS"/>
  <xs:enumeration value="SPADES"/>
</xs:simpleType>
2  -> enumeration facet

//Example: code fragment
@XmlElement
@XmlElementType
public enum Coin {
   @XmlElementValue("1") PENNY(1),
   @XmlElementValue("5") NICKEL(5),
   @XmlElementValue("10") DIME(10),
   @XmlElementValue("25") QUARTER(25) }

<!-- Example: XML Schema fragment -->
<xs:simpleType name="Coin">
   <xs:restriction base="xs:int">
      <xs:enumeration value="1"/>
      <xs:enumeration value="5"/>
      <xs:enumeration value="10"/>
      <xs:enumeration value="25"/>
   </xs:restriction>
</xs:simpleType>

3  -> enumeration facet

//Code fragment
@XmlElement
@XmlElementType
public enum Code {
   @XmlElementValue("1") ONE,
   @XmlElementValue("2") TWO;
}

<!-- Example: XML Schema fragment -->
<xs:simpleType name="Code">
   <xs:restriction base="xs:int">
      <xs:enumeration value="1"/>
      <xs:enumeration value="2"/>
   </xs:restriction>
</xs:simpleType>

since JAXB 2.0

Maps an enum constant in Enum type to XML representation.
Usage

The `@XmlEnumValue` annotation can be used with the following program elements:

- enum constant


This annotation, together with `XmlEnum` provides a mapping of enum type to XML representation.

An enum type is mapped to a schema simple type with enumeration facets. The schema type is derived from the Java type specified in `@XmlEnum.value()`. Each enum constant `@XmlEnumValue` must have a valid lexical representation for the type `@XmlEnum.value()`.

In the absence of this annotation, `Enum.name()` is used as the XML representation.

**Example 1:** Map enum constant name -> enumeration facet

```java
//Example: Code fragment
@XmlEnum(String.class)
public enum Card { CLUBS, DIAMONDS, HEARTS, SPADES }

<!-- Example: XML Schema fragment -->
<x:simpleType name="Card">
    <x:restriction base="xs:string"/>
    <x:enumeration value="CLUBS"/>
    <x:enumeration value="DIAMONDS"/>
    <x:enumeration value="HEARTS"/>
    <x:enumeration value="SPADES"/>
</xs:simpleType>
```

**Example 2:** Map enum constant name(value) -> enumeration facet

```java
//Example: code fragment
@XmlType
@XmlEnum(Integer.class)
public enum Coin {
    @XmlEnumValue("1") PENNY(1),
```
Example 3: Map enum constant name -> enumeration facet

```java
//Code fragment
@XmlType
@XmlEnum(Integer.class)
public enum Code {
    @XmlEnumValue("1") ONE,
    @XmlEnumValue("2") TWO;
}

<!-- Example: XML Schema fragment -->
<xs:simpleType name="Coin">
    <xs:restriction base="xs:int">
        <xs:enumeration value="1"/>
        <xs:enumeration value="5"/>
        <xs:enumeration value="10"/>
        <xs:enumeration value="25"/>
    </xs:restriction>
</xs:simpleType>
```

Since:
JAXB 2.0

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
abstract public String value()

value

public abstract String value
javax.xml.stream.events **Interface XMLEvent**

**All Superinterfaces:**

XMLStreamConstants

**All Known Subinterfaces:**

Attribute, Characters, Comment, DTD, EndDocument,EndElement, EntityDeclaration, EntityReference, Namespace, NotationDeclaration, ProcessingInstruction, StartDocument, StartElement

```java
public interface XMLEvent
extends XMLStreamConstants

Implements: XMLStreamConstants
Implemented by: Attribute, Characters, Comment, DTD, EndDocument, Element, EntityDeclaration, EntityReference, Namespace, NotationDeclaration, ProcessingInstruction, StartDocument, StartElement
```

**XML 1.0 InfoSet value version**

1.0

<table>
<thead>
<tr>
<th>See also</th>
</tr>
</thead>
</table>

This is the base event interface for handling markup events. Events are value objects that are used to communicate the XML 1.0 InfoSet to the Application. Events may be cached and referenced after the parse has
Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants
ATTRIBUTE, CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT

Method Summary

<table>
<thead>
<tr>
<th></th>
<th>asCharacters()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Characters</td>
<td>Returns this event as Characters, may result in a class cast exception if this event is not Characters.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>asEndElement()</th>
</tr>
</thead>
<tbody>
<tr>
<td>EndElement</td>
<td>Returns this event as an end element event, may result in a class cast exception if this event is not a end element.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>asStartElement()</th>
</tr>
</thead>
<tbody>
<tr>
<td>StartElement</td>
<td>Returns this event as a start element event, may result in a class cast exception if this event is not a start element.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>getEventType()</th>
</tr>
</thead>
<tbody>
<tr>
<td>int</td>
<td>Returns an integer code for this event.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>getLocation()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location</td>
<td>Return the location of this event.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>getSchemaType()</strong></td>
<td>This method is provided for implementations to provide optional type information about the associated event.</td>
</tr>
<tr>
<td><strong>isAttribute()</strong></td>
<td>A utility function to check if this event is an Attribute.</td>
</tr>
<tr>
<td><strong>isCharacters()</strong></td>
<td>A utility function to check if this event is Characters.</td>
</tr>
<tr>
<td><strong>isEndDocument()</strong></td>
<td>A utility function to check if this event is an EndDocument.</td>
</tr>
<tr>
<td><strong>isEndElement()</strong></td>
<td>A utility function to check if this event is a EndElement.</td>
</tr>
<tr>
<td><strong>isEntityReference()</strong></td>
<td>A utility function to check if this event is an EntityReference.</td>
</tr>
<tr>
<td><strong>isNamespace()</strong></td>
<td>A utility function to check if this event is a Namespace.</td>
</tr>
<tr>
<td><strong>isProcessingInstruction()</strong></td>
<td>A utility function to check if this event is a ProcessingInstruction.</td>
</tr>
<tr>
<td><strong>isStartDocument()</strong></td>
<td>A utility function to check if this event is a StartDocument.</td>
</tr>
<tr>
<td><strong>isStartElement()</strong></td>
<td>A utility function to check if this event is a StartElement.</td>
</tr>
<tr>
<td><strong>writeAsEncodedUnicode()</strong></td>
<td>This method will write the XMLEvent as per the XML 1.0 specification as Unicode characters.</td>
</tr>
</tbody>
</table>

## Method Detail

**public int getEventType()**
getEventType

int getEventType()

Returns an integer code for this event.

See Also:
XMLStreamConstants.START_ELEMENT,
XMLStreamConstants.END_ELEMENT,
XMLStreamConstants.CHARACTERS,
XMLStreamConstantsATTRIBUTE,
XMLStreamConstants.NAMESPACE,
XMLStreamConstants.PROCESSING_INSTRUCTION,
XMLStreamConstants.COMMENT,
XMLStreamConstants.START_DOCUMENT,
XMLStreamConstants.END_DOCUMENT,
XMLStreamConstants.DTD

public Location getLocation()

Location

See also
javax.xml.stream.Location

getAddress

Location getSocketAddress()

Return the location of this event. The Location returned from this method is non-volatile and will retain its information.

See Also:
Location
public boolean isStartElement()

See also javax.xml.stream.events.StartElement

isStartElement

boolean isStartElement()

A utility function to check if this event is a StartElement.

See Also: StartElement

public boolean isAttribute()

See also javax.xml.stream.events.Attribute

isAttribute

boolean isAttribute()

A utility function to check if this event is an Attribute.

See Also: Attribute

public boolean isNamespace()

See also javax.xml.stream.events.Namespace

isNamespace

boolean isNamespace()
A utility function to check if this event is a Namespace.

See Also:
Namespace

```java
public boolean isEndElement()
EndElement
See also javax.xml.stream.events.EndElement
```

isEndElement

```java
boolean isEndElement()
```

A utility function to check if this event is a EndElement.

See Also:
EndElement

```java
public boolean isEntityReference()
EntityReference
See also javax.xml.stream.events.EntityReference
```

isEntityReference

```java
boolean isEntityReference()
```

A utility function to check if this event is an EntityReference.

See Also:
EntityReference

```java
public boolean isProcessingInstruction()
ProcessingInstruction
```
See also [javax.xml.stream.events.ProcessingInstruction](https://docs.oracle.com/javaee/8/javase-ref-guide/xml/stream-events.html#JAVJAA-1E3612471)

### isProcessingInstruction

```java
boolean isProcessingInstruction()
```

A utility function to check if this event is a ProcessingInstruction.

**See Also:** [ProcessingInstruction](https://docs.oracle.com/javaee/8/javase-ref-guide/xml/stream-events.html#JAVJAA-1E3612471)

---

### public boolean isCharacters()

**Characters**

See also [javax.xml.stream.events.Characters](https://docs.oracle.com/javaee/8/javase-ref-guide/xml/stream-events.html#JAVJAA-1E3612471)

### isCharacters

```java
boolean isCharacters()
```

A utility function to check if this event is Characters.

**See Also:** [Characters](https://docs.oracle.com/javaee/8/javase-ref-guide/xml/stream-events.html#JAVJAA-1E3612471)

---

### public boolean isStartDocument()

**StartDocument**

See also [javax.xml.stream.events.StartDocument](https://docs.oracle.com/javaee/8/javase-ref-guide/xml/stream-events.html#JAVJAA-1E3612471)

### isStartDocument

```java
boolean isStartDocument()
```

A utility function to check if this event is a StartDocument.
public boolean isEndDocument()

EndDocument

See also javax.xml.stream.events.EndDocument

isEndDocument

boolean isEndDocument()

A utility function to check if this event is an EndDocument.

See Also: EndDocument

public StartElement asStartElement()

asStartElement

StartElement asStartElement()

Returns this event as a start element event, may result in a class cast exception if this event is not a start element.

publicEndElement asEndElement()

asEndElement

EndElement asEndElement()
Returns this event as an end element event, may result in a class cast exception if this event is not a end element.

```java
public Characters asCharacters()

Characters

asCharacters

Characters asCharacters()

Returns this event as Characters, may result in a class cast exception if this event is not Characters.

```

```java
public javax.xml.namespace.QName getSchemaType()

null

getSchemaType

QName getSchemaType()

This method is provided for implementations to provide optional type information about the associated event. It is optional and will return null if no information is available.

```

```java
public void writeAsEncodedUnicode(java.io.Writer writer)

throws XMLStreamException

XML 1.0 XMLEvent Unicode Event Event

writer

Throws XMLStreamException:

```
writeAsEncodedUnicode

void writeAsEncodedUnicode(Writer writer)
    throws XMLStreamException

This method will write the XMLEvent as per the XML 1.0 specification as Unicode characters. No indentation or whitespace should be outputted. Any user defined event type SHALL have this method called when being written to on an output stream. Built in Event types MUST implement this method, but implementations MAY choose not call these methods for optimizations reasons when writing out built in Events to an output stream. The output generated MUST be equivalent in terms of the infoset expressed.

Parameters:
    writer - The writer that will output the data

Throws:
    XMLStreamException - if there is a fatal error writing the event

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
public interface XMLEventAllocator

This interface defines a class that allows a user to register a way to allocate events given an XMLStreamReader. An implementation is not required to use the XMLEventFactory implementation but this is recommended. The XMLEventAllocator can be set on an XMLInputFactory using the property "javax.xml.stream.allocator"

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLEvent allocate(XMLStreamReader reader)</td>
<td>This method allocates an event given the current state of the XMLStreamReader.</td>
</tr>
<tr>
<td>void allocate(XMLStreamReader reader, XMLEventConsumer consumer)</td>
<td>This method allocates an event or set of events given the current state of the XMLStreamReader and adds the event or set of events to the consumer that was</td>
</tr>
</tbody>
</table>
This method creates an instance of the XMLEventAllocator.

Method Detail

```java
public XMLEventAllocator newInstance()
XMLEventAllocator XMLInputFactory reader
```

newInstance

```java
XMLEventAllocator newInstance()
```

This method creates an instance of the XMLEventAllocator. This allows the XMLInputFactory to allocate a new instance per reader.

```java
public XMLEvent allocate(XMLStreamReader reader)
throws XMLStreamException
XMLStreamReader XMLEventAllocator reader
null XMLStreamReader
```

allocate

```java
XMLEvent allocate(XMLStreamReader reader)
throws XMLStreamException
```

This method allocates an event given the current state of the XMLStreamReader. If this XMLEventAllocator does not have a one-
to-one mapping between reader states and events this method will return null. This method must not modify the state of the XMLStreamReader.

**Parameters:**
- `reader` - The XMLStreamReader to allocate from

**Returns:**
- the event corresponding to the current reader state

**Throws:**
- [XMLStreamException](#)

```java
public void allocate(XMLStreamReader reader, XMLEventConsumer consumer) throws XMLStreamException
```

This method allocates an event or set of events given the current state of the XMLStreamReader and adds the event or set of events to the consumer that was passed in. This method can be used to expand or contract reader states into event states. This method may modify the state of the XMLStreamReader.

**Parameters:**
- `reader` - The XMLStreamReader to allocate from
- `consumer` - The XMLEventConsumer to add to.

**Throws:**
- [XMLStreamException](#)
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream.util  Interface XMLEventConsumer

All Known Subinterfaces:
   XMLEventWriter

public interface XMLEventConsumer

Implemented by: XMLEventWriter

add

   version 1.0

This interface defines an event consumer interface. The contract of the consumer is to accept the event. This interface can be used to mark an object as able to receive events. Add may be called several times in immediate succession so a consumer must be able to cache events it hasn't processed yet.

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>void add(XMLEvent event)</td>
</tr>
<tr>
<td>This method adds an event to the consumer.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public void add(XMLEvent event) throws XMLStreamException</td>
</tr>
</tbody>
</table>
This method adds an event to the consumer. Calling this method invalidates the event parameter. The client application should discard all references to this event upon calling add. The behavior of an application that continues to use such references is undefined.

**Parameters:**
- event - the event to add, may not be null

**Throws:**
- XMLStreamException
| Overview | Package | Tree | Deprecated | Index | Help |

| PREV CLASS | NEXT CLASS | FRAME | NO FRAMES | SUMMARY: NESTED | FIELD | CONSTR | METHOD |
| FRAMES | DETAIL: FIELD | CONSTR | METHOD | |

---
public abstract class XMLEventFactory
extends Object

XMLEvent version 1.0

    javax.xml.stream.events.StartElement,
    javax.xml.stream.events.EndElement,
    javax.xml.stream.events.ProcessingInstruction,
    javax.xml.stream.events.Comment,
    javax.xml.stream.events.Characters,
    javax.xml.stream.events.StartDocument,
    javax.xml.stream.events.EndDocument,
    javax.xml.stream.events.DTD

This interface defines a utility class for creating instances of XMLEvents

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.
See Also: StartElement, EndElement, ProcessingInstruction, Comment,
         Characters, StartDocument, EndDocument, DTD
<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>Attribute createAttribute(QName name, String value)</td>
<td>Create a new Attribute</td>
</tr>
<tr>
<td>Abstract</td>
<td>Attribute createAttribute(String localName, String value)</td>
<td>Create a new Attribute</td>
</tr>
<tr>
<td>Abstract</td>
<td>Attribute createAttribute(String prefix, String namespaceURI, String localName, String value)</td>
<td>Create a new Attribute</td>
</tr>
<tr>
<td>Abstract</td>
<td>Characters createCData(String content)</td>
<td>Create a Characters event with the CData flag set to true</td>
</tr>
<tr>
<td>Abstract</td>
<td>Characters createCharacters(String content)</td>
<td>Create a Characters event, this method does not check if the content is all whitespace.</td>
</tr>
<tr>
<td>Abstract</td>
<td>Comment createComment(String text)</td>
<td>Create a comment</td>
</tr>
<tr>
<td>Abstract</td>
<td>DTD createDTD(String dtd)</td>
<td>Create a document type definition event This string contains the entire document type declaration that matches the doctypedcl in the XML 1.0 specification</td>
</tr>
<tr>
<td>Abstract</td>
<td>EndDocument createEndDocument()</td>
<td>Creates a new instance of an EndDocument event</td>
</tr>
<tr>
<td>Abstract</td>
<td>EndElement createEndElement(QName name, Iterator namespaces)</td>
<td>Create a new EndElement</td>
</tr>
<tr>
<td>Abstract</td>
<td>EndElement createEndElement(String prefix, String namespaceUri, String localName)</td>
<td>Create a new EndElement</td>
</tr>
<tr>
<td>Abstract</td>
<td>EndElement createEndElement(String prefix, String namespaceUri, String localName, Iterator namespaces)</td>
<td>Create a new EndElement</td>
</tr>
<tr>
<td>Abstract</td>
<td>EndElement createEntityReference(String name,</td>
<td>Create a new EndElement</td>
</tr>
<tr>
<td>abstract EntityReference</td>
<td>EntityDeclaration declaration)</td>
<td></td>
</tr>
<tr>
<td>--------------------------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a new instance of a EntityReference event</td>
<td></td>
</tr>
<tr>
<td>abstract Characters</td>
<td>createIgnorableSpace(String content)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create an ignorable space</td>
<td></td>
</tr>
<tr>
<td>abstract Namespace</td>
<td>createNamespace(String namespaceURI)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a new default Namespace</td>
<td></td>
</tr>
<tr>
<td>abstract Namespace</td>
<td>createNamespace(String prefix, String namespaceUri)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a new Namespace</td>
<td></td>
</tr>
<tr>
<td>abstract ProcessingInstruction</td>
<td>createProcessingInstruction(String target, String data)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a processing instruction</td>
<td></td>
</tr>
<tr>
<td>abstract Characters</td>
<td>createSpace(String content)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a Characters event with the isSpace flag set to true</td>
<td></td>
</tr>
<tr>
<td>abstract StartDocument</td>
<td>createStartDocument()</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a new instance of a StartDocument event</td>
<td></td>
</tr>
<tr>
<td>abstract StartDocument</td>
<td>createStartDocument(String encoding)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a new instance of a StartDocument event</td>
<td></td>
</tr>
<tr>
<td>abstract StartDocument</td>
<td>createStartDocument(String encoding, String version)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a new instance of a StartDocument event</td>
<td></td>
</tr>
<tr>
<td>abstract StartDocument</td>
<td>createStartDocument(String encoding, String version, boolean standalone)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Creates a new instance of a StartDocument event</td>
<td></td>
</tr>
<tr>
<td>abstract StartElement</td>
<td>createStartElement(QName name, Iterator attributes, Iterator namespaces)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a new StartElement.</td>
<td></td>
</tr>
<tr>
<td>abstract StartElement</td>
<td>createStartElement(String prefix, String namespaceUri, String localName)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Create a new StartElement.</td>
<td></td>
</tr>
<tr>
<td>abstract StartElement</td>
<td>createStartElement(String prefix,</td>
<td></td>
</tr>
</tbody>
</table>
abstract StartElement(String namespaceUri, String localName, Iterator attributes, Iterator namespaces)

Create a new StartElement.

abstract StartElement createStartElement(String prefix, String namespaceUri, String localName, Iterator attributes, Iterator namespaces, NamespaceContext context)

Create a new StartElement.

static XMLEventFactory newInstance()

Create a new instance of the factory

static XMLEventFactory newInstance(String factoryId, ClassLoader classLoader)

Create a new instance of the factory

abstract void setLocation(Location location)

This method allows setting of the Location on each event that is created by this factory.

Methods inherited from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait

Constructor Detail

protected XMLEventFactory()

XMLEventFactory

protected XMLEventFactory()

Method Detail

public static XMLEventFactory newInstance() throws FactoryConfigurationError
Throws: FactoryConfigurationException

newInstance

public static XMLEventFactory newInstance() throws FactoryConfigurationException

Create a new instance of the factory

Throws:
FactoryConfigurationException - if an instance of this factory cannot be loaded

newInstance

public static XMLEventFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationException

factoryId
classLoader
return

Throws: FactoryConfigurationException

newInstance

public static XMLEventFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationException

Parameters:
factoryId - Name of the factory to find, same as a property name
classLoader - classLoader to use
Returns:
   the factory implementation

Throws:
   FactoryConfigurationError - if an instance of this factory cannot be loaded

abstract public void setLocation(Location location)
   Location location null

   location

setLocation

public abstract void setLocation(Location location)

   This method allows setting of the Location on each event that is created by this factory. The values are copied by value into the events created by this factory. To reset the location information set the location to null.

   Parameters:
   location - the location to set on each event created

abstract public Attribute createAttribute(String prefix, String namespaceURI, String localName, String value)

   Attribute

   prefix null
   namespaceURI null
   localName XML localName null
   value unll
   return Attribute

createAttribute
public abstract Attribute createAttribute(String prefix,
String namespaceURI,
String localName,
String value)

Create a new Attribute

Parameters:
prefix - the prefix of this attribute, may not be null
namespaceURI - the attribute value is set to this value, may not be null
localName - the local name of the XML name of the attribute,
localName cannot be null
value - the attribute value to set, may not be null

Returns:
the Attribute with specified values

abstract public Attribute createAttribute(String localName,
String value)

Attribute

localName  XML localName  null
value      unll
return     Attribute

createAttribute

public abstract Attribute createAttribute(String localName,
String value)

Create a new Attribute

Parameters:
localName - the local name of the XML name of the attribute,
localName cannot be null
value - the attribute value to set, may not be null

Returns:
the Attribute with specified values
abstract public Attribute createAttribute(javax.xml.namespace.QName name, String value)

Attribute

<table>
<thead>
<tr>
<th>name</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>null</td>
</tr>
<tr>
<td>return</td>
<td>Attribute</td>
</tr>
</tbody>
</table>

createAttribute

public abstract Attribute createAttribute(QName name, String value)

Create a new Attribute

Parameters:
- name - the qualified name of the attribute, may not be null
- value - the attribute value to set, may not be null

Returns: the Attribute with specified values

abstract public Namespace createNamespace(String namespaceURI)

Namespace

<table>
<thead>
<tr>
<th>namespaceURI</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>return</td>
<td>Namespace</td>
</tr>
</tbody>
</table>

createNamespace

public abstract Namespace createNamespace(String namespaceURI)

Create a new default Namespace
abstract public Namespace createNamespace(String prefix, String namespaceUri)

Create a new Namespace

Parameters:
  prefix - the prefix of this namespace, may not be null
  namespaceUri - the attribute value is set to this value, may not be null

Returns:
  the Namespace with the specified values

abstract public StartElement createStartElement(javax.xml.namespace.QName name, java.util.Iterator<E> attributes, java.util.Iterator<E> namespaces)

StartElement Namespace Iterator StartElement Attribute Iterator StartElement

Parameters:
  name - null
  attributes - StartElement Attribute null
namespaces

return

createStartElement

public abstract **StartElement** createStartElement(**QName** name,
**Iterator** attributes,
**Iterator** namespaces)

Create a new StartElement. Namespaces can be added to this StartElement by passing in an Iterator that walks over a set of Namespace interfaces. Attributes can be added to this StartElement by passing an iterator that walks over a set of Attribute interfaces.

**Parameters:**
- **name** - the qualified name of the attribute, may not be null
- **attributes** - an optional unordered set of objects that implement Attribute to add to the new StartElement, may be null
- **namespaces** - an optional unordered set of objects that implement Namespace to add to the new StartElement, may be null

**Returns:**
- an instance of the requested StartElement

---

abstract public **StartElement** createStartElement(**String** prefix,
**String** namespaceUri,
**String** localName)

**StartElement** **NamespaceContext**

**NamespaceContext**
- **namespaceUri**
- **localName**
- **prefix**
- **return**

createStartElement

public abstract **StartElement** createStartElement(**String** prefix,
Create a new StartElement. This defaults the NamespaceContext to an empty NamespaceContext. Querying this event for its namespaces or attributes will result in an empty iterator being returned.

**Parameters:**
- `namespaceUri` - the uri of the QName of the new StartElement
- `localName` - the local name of the QName of the new StartElement
- `prefix` - the prefix of the QName of the new StartElement

**Returns:**
an instance of the requested StartElement

```java
abstract public StartElement createStartElement(String prefix, String namespaceUri, String localName, java.util.Iterator<E> attributes, java.util.Iterator<E> namespaces)
```

### CreateStartElement

```java
public abstract StartElement createStartElement(String prefix, String namespaceUri, String localName, Iterator attributes, Iterator namespaces)
```
Create a new StartElement. Namespaces can be added to this StartElement by passing in an Iterator that walks over a set of Namespace interfaces. Attributes can be added to this StartElement by passing an iterator that walks over a set of Attribute interfaces.

Parameters:
- namespaceUri - the uri of the QName of the new StartElement
- localName - the local name of the QName of the new StartElement
- prefix - the prefix of the QName of the new StartElement
- attributes - an unordered set of objects that implement Attribute to add to the new StartElement
- namespaces - an unordered set of objects that implement Namespace to add to the new StartElement

Returns:
an instance of the requested StartElement

abstract public StartElement createStartElement(String prefix, String namespaceUri, String localName, java.util.Iterator<E> attributes, java.util.Iterator<E> namespaces, javax.xml.namespace.NamespaceContext context)

StartElement Namespace Iterator StartElement
StartElement Attribute Iterator StartElement
namespaceUri StartElement QName URI
localName StartElement QName
prefix StartElemente QName
attributes StartElement Attribute null
namespaces StartElement Namespace null
context startElement
return StartElement

createStartElement

public abstract StartElement createStartElement(String prefix, String namespaceUri, String localName, java.util.Iterator<E> attributes, java.util.Iterator<E> namespaces, javax.xml.namespace.NamespaceContext context)
Create a new StartElement. Namespaces can be added to this StartElement by passing in an Iterator that walks over a set of Namespace interfaces. Attributes can be added to this StartElement by passing an iterator that walks over a set of Attribute interfaces.

Parameters:
- `namespaceUri` - the uri of the QName of the new StartElement
- `localName` - the local name of the QName of the new StartElement
- `prefix` - the prefix of the QName of the new StartElement
- `attributes` - an unordered set of objects that implement Attribute to add to the new StartElement, may be null
- `namespaces` - an unordered set of objects that implement Namespace to add to the new StartElement, may be null
- `context` - the namespace context of this element

Returns:
- an instance of the requested StartElement

abstract public **EndElement**
createEndElement( javax.xml.namespace.QName name, java.util.Iterator<E> namespaces)

**EndElement**
- `name`
- `namespaces`
- `return`

createEndElement

public abstract **EndElement** createEndElement( QName name, Iterator namespaces)
Create a new EndElement

**Parameters:**
- name - the qualified name of the EndElement
- namespaces - an optional unordered set of objects that implement Namespace that have gone out of scope, may be null

**Returns:**
an instance of the requested EndElement

```java
abstract public EndElement createEndElement(String prefix, String namespaceUri, String localName)
```

## createEndElement

Create a new EndElement

**Parameters:**
- namespaceUri - the uri of the QName of the new StartElement
- localName - the local name of the QName of the new StartElement
- prefix - the prefix of the QName of the new StartElement

**Returns:**
an instance of the requested EndElement

```java
abstract public EndElement createEndElement(String prefix, String namespaceUri, String localName, String qName)
```
java.util.Iterator<E> namespaces)

EndElement

<table>
<thead>
<tr>
<th>namespaceUri</th>
<th>StartElement QName</th>
<th>URI</th>
</tr>
</thead>
<tbody>
<tr>
<td>localName</td>
<td>StartElement QName</td>
<td></td>
</tr>
<tr>
<td>prefix</td>
<td>StartElement QName</td>
<td></td>
</tr>
<tr>
<td>namespaces</td>
<td>Namespace null</td>
<td></td>
</tr>
<tr>
<td>return</td>
<td>EndElement</td>
<td></td>
</tr>
</tbody>
</table>

createEndElement

public abstract EndElement createEndElement(String prefix,
                                          String namespaceUri,
                                          String localName,
                                          Iterator namespaces)

Create a new EndElement

Parameters:
- namespaceUri - the uri of the QName of the new StartElement
- localName - the local name of the QName of the new StartElement
- prefix - the prefix of the QName of the new StartElement
- namespaces - an unordered set of objects that implement Namespace that have gone out of scope, may be null

Returns:
- an instance of the requested EndElement

abstract public Characters createCharacters(String content)

Character #createSpace(String)

Creates a new EndElement

Parameters:
- namespaceUri - the uri of the QName of the new StartElement
- localName - the local name of the QName of the new StartElement
- prefix - the prefix of the QName of the new StartElement
- namespaces - an unordered set of objects that implement Namespace that have gone out of scope, may be null

Returns:
- an instance of the requested EndElement

abstract public Characters createCharacters(String content)

Character #createSpace(String)
public abstract Characters createCharacters(String content)

Create a Characters event, this method does not check if the content is all whitespace. To create a space event use #createSpace(String)

Parameters:
  content - the string to create
Returns:
a Characters event

abstract public Characters createCData(String content)
CData true Character

createCData

public abstract Characters createCData(String content)

Create a Characters event with the CData flag set to true

Parameters:
  content - the string to create
Returns:
a Characters event

abstract public Characters createSpace(String content)
isSpace true Character

createSpace

public abstract Characters createSpace(String content)
Create a Characters event with the isSpace flag set to true

**Parameters:**
- content - the content of the space to create

**Returns:**
a Characters event

---

abstract public Characters createIgnorableSpace(String content)

```
calendar
    content
    return

Character
```

**createIgnorableSpace**

public abstract Characters createIgnorableSpace(String content)

Create an ignorable space

**Parameters:**
- content - the space to create

**Returns:**
a Characters event

---

abstract public StartDocument createStartDocument()

**StartDocument**

```
    return

StartDocument
```

**createStartDocument**

public abstract StartDocument createStartDocument()

Creates a new instance of a StartDocument event
abstract public StartDocument createStartDocument(String encoding, String version, boolean standalone)
StartDocument
  encoding
  version
  standalone
  return
return StartDocument

createStartDocument

public abstract StartDocument createStartDocument(String encoding, String version, boolean standalone)

Creates a new instance of a StartDocument event

Parameters:
  encoding - the encoding style
  version - the XML version
  standalone - the status of standalone may be set to "true" or "false"

Returns:
a StartDocument event

abstract public StartDocument createStartDocument(String encoding, String version)
StartDocument
  encoding
  version
  return
return StartDocument
createStartDocument

public abstract StartDocument createStartDocument(String encoding, String version)

Creates a new instance of a StartDocument event

Parameters:
   encoding - the encoding style
   version - the XML version

Returns:
   a StartDocument event

----------------------------------------

abstract public StartDocument createStartDocument(String encoding)
StartDocument
   encoding
   return
   StartDocument

createStartDocument

public abstract StartDocument createStartDocument(String encoding)

Creates a new instance of a StartDocument event

Parameters:
   encoding - the encoding style

Returns:
   a StartDocument event

----------------------------------------

abstract public EndDocument createEndDocument()
EndDocument
   return
   EndDocument
createEndDocument

public abstract EndDocument createEndDocument()

    Creates a new instance of an EndDocument event

    Returns:
    an EndDocument event

abstract public EntityReference createEntityReference(String name, EntityDeclaration declaration)

    EntityReference
        name
        declaration
        return EntityReference

createEntityReference

public abstract EntityReference createEntityReference(String name, EntityDeclaration declaration)

    Creates a new instance of a EntityReference event

    Parameters:
    name - The name of the reference
    declaration - the declaration for the event

    Returns:
    an EntityReference event

abstract public Comment createComment(String text)

text Comment
createComment

public abstract Comment createComment(String text)

Create a comment

Parameters:

text - The text of the comment a Comment event

abstract public ProcessingInstruction
createProcessingInstruction(String target, String data)

target
data
return ProcessingInstruction

createProcessingInstruction

public abstract ProcessingInstruction createProcessingInstruction(String target, String data)

Create a processing instruction

Parameters:

target - The target of the processing instruction
data - The text of the processing instruction

Returns:
a ProcessingInstruction event

abstract public DTD createDTD(String dtd)

XML 1.0 doctypedecl
dtd
return DTD
createDTD

public abstract DTD createDTD(String dtd)

Create a document type definition event This string contains the entire document type declaration that matches the doctypedcl in the XML 1.0 specification

Parameters:

dtd - the text of the document type definition

Returns:

a DTD event
public interface XMLEventReader

extends Iterator

Implements: java.util.Iterator<E>
Implemented by: EventReaderDelegate

This is the top level interface for parsing XML Events. It provides the ability to peek at the next event and returns configuration information through the property interface.

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>close()</td>
<td>Frees any resources associated with this Reader.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getElementText()</code></td>
<td>Reads the content of a text-only element.</td>
</tr>
<tr>
<td><code>getProperty(String name)</code></td>
<td>Get the value of a feature/property from the underlying implementation</td>
</tr>
<tr>
<td><code>hasNext()</code></td>
<td>Check if there are more events.</td>
</tr>
<tr>
<td><code>nextEvent()</code></td>
<td>Get the next XMLEvent</td>
</tr>
<tr>
<td><code>nextTag()</code></td>
<td>Skips any insignificant space events until a START_ELEMENT or END_ELEMENT is reached.</td>
</tr>
<tr>
<td><code>peek()</code></td>
<td>Check the next XMLEvent without reading it from the stream.</td>
</tr>
</tbody>
</table>

Methods inherited from interface `java.util.Iterator`:
- `next`, `remove`

### Method Detail

**public XMLEvent nextEvent() throws XMLStreamException**

**XMLEvent**

- Throws `XMLStreamException`: XML
- Throws `NoSuchElementException`: 
- See also [javax.xml.stream.events.XMLEvent](#)

**nextEvent**

**XMLEvent nextEvent()**

- throws `XMLStreamException`

Get the next XMLEvent
Throws:

XMLStreamException - if there is an error with the underlying XML.
NoSuchElementException - iteration has no more elements.

See Also:
XMLEvent

public boolean hasNext()
true false
    return reader true false

hasNext

boolean hasNext()

Check if there are more events. Returns true if there are more events and false otherwise.

Specified by:
hasNext in interface Iterator

Returns:
true if the event reader has more events, false otherwise

public XMLEvent peek() throws XMLStreamException
XMLEvent EOF  XMLEvent null peek()
next()

Throws
XMLStreamException:
See also
javax.xml.stream.events.XMLEvent

peek

XMLEvent peek()
    throws XMLStreamException
Check the next XMLEvent without reading it from the stream. Returns null if the stream is at EOF or has no more XMLEvents. A call to peek() will be equal to the next return of next().

Throws:

XMLStreamException

See Also:

XMLEvent

```java
public String getElementText() throws XMLStreamException
```

START_ELEMENT END_ELEMENT

Throws XMLStreamException: START_ELEMENT

getElementText

```java
String getElementText() throws XMLStreamException
```

Reads the content of a text-only element. Precondition: the current event is START_ELEMENT. Postcondition: The current event is the corresponding END_ELEMENT.

Throws:

XMLStreamException - if the current event is not a START_ELEMENT or if a non text element is encountered

```java
public XMLEvent nextTag() throws XMLStreamException
```

START_ELEMENT END_ELEMENT

DTD

Throws XMLStreamException:
**nextTag()**

Skips any insignificant space events until a START_ELEMENT or END_ELEMENT is reached. If anything other than space characters are encountered, an exception is thrown. This method should be used when processing element-only content because the parser is not able to recognize ignorable whitespace if the DTD is missing or not interpreted.

**Throws:**

- **XMLStreamException** - if anything other than space characters are encountered

---

**getProperty(String name) throws IllegalArgumentException**

Get the value of a feature/property from the underlying implementation

**Parameters:**

- **name** - The name of the property

**Returns:**

- The value of the property

**Throws:**

- **IllegalArgumentException** - if the property is not supported
public void close() throws XMLStreamException

Reader

Throws XMLStreamException:

close

void close()
throws XMLStreamException

Frees any resources associated with this Reader. This method does not close the underlying input source.

Throws:

XMLStreamException - if there are errors freeing associated resources

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
**javax.xml.stream** Interface **XMLEventWriter**

**All Superinterfaces:**

[XMLEventConsumer](#)

---

**public interface XMLEventWriter**

extends [XMLEventConsumer](#)

**Implements:** [XMLEventConsumer](#)

**XML version**

1.0

[javax.xml.stream.XMLEventReader](#),

[javax.xml.stream.events.XMLEvent](#),

[javax.xml.stream.events.Characters](#),

[javax.xml.stream.events.ProcessingInstruction](#),

[javax.xml.stream.events.StartElement](#),

[javax.xml.stream.events.EndElement](#)

---

This is the top level interface for writing XML documents. Instances of this interface are not required to validate the form of the XML.

**Version:**

1.0

**Author:**

Copyright (c) 2003 by BEA Systems. All Rights Reserved.

**See Also:**

[XMLEventReader](#), [XMLEvent](#), [Characters](#), [ProcessingInstruction](#), [StartElement](#), [EndElement](#)

---

**Method Summary**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>add</strong>(XMLEvent event)</td>
<td>Add an event to the output stream Adding a</td>
</tr>
</tbody>
</table>
START_ELEMENT will open a new namespace scope that will be closed when the corresponding END_ELEMENT is written.

```java
void add(XMLEventReader reader)
    Adds an entire stream to an output stream, calls next() on the inputStream argument until hasNext() returns false This should be treated as a convenience method that will perform the following loop over all the events in an event reader and call add on each event.
```

```java
void close()
    Frees any resources associated with this stream
```

```java
void flush()
    Writes any cached events to the underlying output mechanism
```

```java
namespaceContext getNamespaceContext()
    Returns the current namespace context.
```

```java
String getPrefix(String uri)
    Gets the prefix the uri is bound to
```

```java
void setDefaultNamespace(String uri)
    Binds a URI to the default namespace This URI is bound in the scope of the current START_ELEMENT / END_ELEMENT pair.
```

```java
void setNamespaceContext(NamespaceContext context)
    Sets the current namespace context for prefix and uri bindings.
```

```java
void setPrefix(String prefix, String uri)
    Sets the prefix the uri is bound to.
```

### Method Detail

public void flush() throws [XMLStreamException](#)

**Throws**

XMLStreamException:
flush

void flush() throws XMLStreamException

Writes any cached events to the underlying output mechanism

Throws:
   XMLStreamException

public void close() throws XMLStreamException

    throws XMLStreamException:

close

void close() throws XMLStreamException

    Frees any resources associated with this stream

    XMLStreamException:

public void add(XMLEvent event) throws XMLStreamException

START_ELEMENT   END_ELEMENT

<table>
<thead>
<tr>
<th>writer</th>
</tr>
</thead>
<tbody>
<tr>
<td>START_ELEMENT</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
### EventWriter

- **QName**
  - name

- **URI**
  - writer

- **QName**
  - writer

- **EventWriter**

- **ATTRIBUTE**
  - QName name
  - String value
  - QName type

- **NAMESPACE**
  - String prefix
  - String namespace
  - URI
  - boolean
  - isDefault
  - NamespaceDeclaration

- **PROCESSING_INSTRUCTION**
  - String target
  - String data

- **COMMENT**
  - String comment

- **START_DOCUMENT**
  - String encoding
  - boolean standalone
  - String version

- **END_DOCUMENT**

- **DTD**
  - String DocumentTypeDefinition

---

<table>
<thead>
<tr>
<th>event</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Throws</strong></td>
</tr>
</tbody>
</table>

### add

```java
void add(XMLEvent event)
    throws XMLStreamException
```

Add an event to the output stream. Adding a START_ELEMENT will open a new namespace scope that will be closed when the corresponding END_ELEMENT is written.

### Required and optional fields for events added to the writer

<table>
<thead>
<tr>
<th>Required fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>name</td>
</tr>
<tr>
<td>namespace</td>
</tr>
<tr>
<td>version</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Optional fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>standalone</td>
</tr>
<tr>
<td>encoding</td>
</tr>
<tr>
<td>Event Type</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td>START_ELEMENT</td>
</tr>
<tr>
<td>END_ELEMENT</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
</tr>
<tr>
<td>NAMESPACE</td>
</tr>
</tbody>
</table>
public void add(XMLEventReader reader) throws XMLStreamException
hasNext() false inputStream next()
reader add
   reader
   Throws XMLStreamException:

add

void add(XMLEventReader reader)
   throws XMLStreamException

   Adds an entire stream to an output stream, calls next() on the
   inputStream argument until hasNext() returns false. This should be
   treated as a convenience method that will perform the following loop
   over all the events in an event reader and call add on each event.

   Parameters:
   reader - the event stream to add to the output

   Throws:
   XMLStreamException

public String getPrefix(String uri) throws XMLStreamException

   URI

   uri
   Throws URI
   XMLStreamException:

getPrefix

String getPrefix(String uri)
   throws XMLStreamException

   Gets the prefix the uri is bound to

   Parameters:
   uri - the uri to look up

   Throws:
   XMLStreamException
public void setPrefix(String prefix, String uri) throws XMLStreamException

Sets the prefix the uri is bound to. This prefix is bound in the scope of the current START_ELEMENT / END_ELEMENT pair. If this method is called before a START_ELEMENT has been written the prefix is bound in the root scope.

Parameters:
prefix - the prefix to bind to the uri
uri - the uri to bind to the prefix

Throws: XMLStreamException

public void setDefaultNamespace(String uri) throws XMLStreamException

Sets the default namespace for the current scope. If this method is called before a START_ELEMENT has been written the default namespace is bound in the root scope.

Parameters:
uri - the uri to bind to the default namespace

Throws: XMLStreamException
void setDefaultCloseOperation(String uri)
    throws XMLStreamException

Binds a URI to the default namespace. This URI is bound in the scope of the current START_ELEMENT / END_ELEMENT pair. If this method is called before a START_ELEMENT has been written the uri is bound in the root scope.

Parameters:
    uri - the uri to bind to the default namespace

Throws:
    XMLStreamException

public void setNamespaceContext(javax.xml.namespace.NamespaceContext context) throws XMLStreamException

Sets the current namespace context for prefix and uri bindings. This context becomes the root namespace context for writing and will replace the current root namespace context. Subsequent calls to setPrefix and setDefaultCloseOperation will bind namespaces using the context passed to the method as the root context for resolving namespaces.

Parameters:
    context - the namespace context to use for this writer

Throws:
    XMLStreamException
public javax.xml.namespace.NamespaceContext
getNamespaceContext() {
    return getNamespaceContext();
}

getNamespaceContext

NamespaceContext getNamespaceContext() {

    Returns the current namespace context.

    Returns: the current namespace context

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation  Annotation Type XmlID

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlID

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean XML ID

XML

@XmlID
- JavaBean
- static transient

javax.xml.bind.package javadoc ""

- @XmlID
- JavaBean java.lang.String
- @XmlID @XmlElement @XmlAttribute

JavaBean xs:ID

// Example: code fragment
public class Customer {
    @XmlAttribute
    @XmlID
    public String getCustomerID();
    public void setCustomerID(String id);
    .... other properties not shown
}
Maps a JavaBean property to XML ID.

To preserve referential integrity of an object graph across XML serialization followed by a XML deserialization, requires an object reference to be marshalled by reference or containment appropriately. Annotations @XmlID and @XmlIDREF together allow a customized mapping of a JavaBean property's type by containment or reference.

Usage

The @XmlID annotation can be used with the following program elements:

- a JavaBean property
- non static, non transient field


The usage is subject to the following constraints:

- At most one field or property in a class can be annotated with @XmlID.
- The JavaBean property's type must be java.lang.String.
- The only other mapping annotations that can be used with @XmlID are: @XmlElement and @XmlAttribute.
Example: Map a JavaBean property's type to xs:ID

    // Example: code fragment
    public class Customer {
        @XmlAttribute
        @XmlID
        public String getCustomerID();
        public void setCustomerID(String id);
        .... other properties not shown
    }

    <!-- Example: XML Schema fragment -->
    <xs:complexType name="Customer">
        <xs:complexContent>
            <xs:sequence>
            ....
             </xs:sequence>
            <xs:attribute name="customerID" type="xs:ID"/>
        </xs:complexContent>
    </xs:complexType>

Since:
JAXB2.0
Version:
$Revision: 1.5 $
Author:
Sekhar Vajjhala, Sun Microsystems, Inc.
See Also:
XmlIDREF

Overview  Package  Tree  Deprecated  Index  Help
PREV CLASS  NEXT CLASS
SUMMARY: REQUIRED | OPTIONAL
FRAMES  NO FRAMES
DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.annotation  Annotation Type XmlIDREF

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlIDREF

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean  XML IDREF

XML

@XmlIDREF

- JavaBean
- static transient

javax.xml.bind.package javadoc ""

- collection
- single valued

collection java.lang.Object

 collection java.lang.Object

- element
- attribute

java.util.List

JavaBean  xs:IDREF

//EXAMPLE: Code fragment
public class Shipping {
    @XmlIDREF public Customer getCustomer();
    public void setCustomer(Customer customer);
    ....
<!-- Example: XML Schema fragment -->
<xs:complexType name="Shipping">
  <xs:complexContent>
    <xs:sequence>
      <xs:element name="customer" type="xs:IDREF"/>
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>

// By default, Customer maps to complex type xs:Customer
public class Customer {

  // map JavaBean property type to xs:ID
  @XmlID public String getCustomerID();
  public void setCustomerID(String id);

  // .... other properties not shown
}

// By default, Invoice maps to a complex type xs:Invoice
public class Invoice {

  // map by reference
  @XmlIDREF public Customer getCustomer();
  public void setCustomer(Customer customer);

  // .... other properties not shown here
}

// By default, Shipping maps to complex type xs:Shipping
public class Shipping {

  // map by reference
  @XmlIDREF public Customer getCustomer();
  public void setCustomer(Customer customer);
}

// at least one class must reference Customer by containment;
// Customer instances won't be marshalled.
@XmlElement(name="CustomerData")
public class CustomerData {
// map reference to Customer by containment by default.
public Customer getCustomer();

// maps reference to Shipping by containment by default.
public Shipping getShipping();

// maps reference to Invoice by containment by default.
public Invoice getInvoice();
}

<!-- XML Schema mapping for above code fragment -->
<xs:complexType name="Invoice">
  <xs:complexContent>
    <xs:sequence>
      <xs:element name="customer" type="xs:IDREF"/>
      ....
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="Shipping">
  <xs:complexContent>
    <xs:sequence>
      <xs:element name="customer" type="xs:IDREF"/>
      ....
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="Customer">
  <xs:complexContent>
    <xs:sequence>
      ....
    </xs:sequence>
    <xs:attribute name="CustomerID" type="xs:ID"/>
  </xs:complexContent>
</xs:complexType>

<xs:complexType name="CustomerData">
  <xs:complexContent>
    <xs:sequence>
      <xs:element name="customer" type="xs:Customer"/>
      <xs:element name="shipping" type="xs:Shipping"/>
      <xs:element name="invoice" type="xs:Invoice"/>
    </xs:sequence>
  </xs:complexContent>
</xs:complexType>

<xs:element name="customerData" type="xs:CustomerData"/>
<customerData>
    <customer customerID="Alice">
        ....
    </customer>

    <shipping customer="Alice">
        ....
    </shipping>

    <invoice customer="Alice">
        ....
    </invoice>
</customerData>

3  IDREF

    // Code fragment
    public class Shipping {
        @XmlIDREF
        @XmlElement(name="Alice")
            public List customers;
    }

</xs:complexType name="Shipping">
</xs:sequence>
</xs:complexType>

4  IDREF

    // Code fragment
    public class Shipping {
        @XmlIDREF
        @XmlElements(
            @XmlElement(name="Alice", type="Customer.class")
            @XmlElement(name="John", type="InternationalCustomer.c"
        public List customers;
    }

</!-- XML Schema fragment -->
Maps a JavaBean property to XML IDREF.

To preserve referential integrity of an object graph across XML serialization followed by a XML deserialization, requires an object reference to be marshalled by reference or containment appropriately. Annotations @XmlID and @XmlIDREF together allow a customized mapping of a JavaBean property's type by containment or reference.

Usage

The @XmlIDREF annotation can be used with the following program elements:

- a JavaBean property
- non static, non transient field


The usage is subject to the following constraints:

- If the type of the field or property is a collection type, then the collection item type must contain a property or field annotated with @XmlID.
- If the field or property is single valued, then the type of the property or field must contain a property or field annotated with @XmlID.
Note: If the collection item type or the type of the property (for non collection type) is java.lang.Object, then the instance must contain a property/field annotated with @XmlID attribute.

- This annotation can be used with the following annotations: XmlElement, XmlAttribute, XmlList, and XmlElements.

Example: Map a JavaBean property to xs:IDREF (i.e. by reference rather than by containment)

```java
//EXAMPLE: Code fragment
public class Shipping {
    @XmlIDREF public Customer getCustomer();
    public void setCustomer(Customer customer);
    ....
}

<!-- Example: XML Schema fragment -->
<xs:complexType name="Shipping">
    <xs:complexContent>
        <xs:sequence>
            <xs:element name="customer" type="xs:IDREF"/>
            ....
        </xs:sequence>
    </xs:complexContent>
</xs:complexType>
```

Example 2: The following is a complete example of containment versus reference.

```java
// By default, Customer maps to complex type xs:Customer
public class Customer {
    // map JavaBean property type to xs:ID
    @XmlID public String getCustomerID();
    public void setCustomerID(String id);
    // .... other properties not shown
}

// By default, Invoice maps to a complex type xs:Invoice
public class Invoice {
```
// map by reference
@XmlIDREF public Customer getCustomer();
public void setCustomer(Customer customer);

// .... other properties not shown here
}

// By default, Shipping maps to complex type xs:Shipping
public class Shipping {

    // map by reference
    @XmlIDREF public Customer getCustomer();
    public void setCustomer(Customer customer);
}

// at least one class must reference Customer by containment;
// Customer instances won't be marshalled.
@XmlElement(name="CustomerData")
public class CustomerData {

    // map reference to Customer by containment by default.
    public Customer getCustomer();

    // maps reference to Shipping by containment by default.
    public Shipping getShipping();

    // maps reference to Invoice by containment by default.
    public Invoice getInvoice();
}

<!-- XML Schema mapping for above code fragment -->

<xs:complexType name="Invoice">
    <xs:complexContent>
        <xs:sequence>
            <xs:element name="customer" type="xs:IDREF"/>
            ....
        </xs:sequence>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="Shipping">
    <xs:complexContent>
        <xs:sequence>
            <xs:element name="customer" type="xs:IDREF"/>
            ....
        </xs:sequence>
    </xs:complexContent>
</xs:complexType>

<xs:complexType name="Customer"/>
Example 3: Mapping List to repeating element of type IDREF

    // Code fragment
    public class Shipping {
        @XmlIDREF
        @XmlElement(name="Alice")
        public List customers;
    }

    <!-- XML schema fragment -->
    <xs:complexType name="Shipping">
        <xs:sequence>
Example 4: Mapping a List to a list of elements of type IDREF.

//Code fragment
public class Shipping {
    @XmlIDREF
    @XmlElements(
        @XmlElement(name="Alice", type="Customer.class")
        @XmlElement(name="John", type="InternationalCustomer.class")
    public List customers;
}

<!-- XML Schema fragment -->
<xs:complexType name="Shipping">
    <xs:sequence>
        <xs:choice minOccurs="0" maxOccurs="unbounded">
            <xs:element name="Alice" type="xs:IDREF"/>
            <xs:element name="John" type="xs:IDREF"/>
        </xs:choice>
    </xs:sequence>
</xs:complexType>

Since:
   JAXB2.0

Version:
   $Revision: 1.12 $

Author:
   Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:
   XmlID
to license terms.

PS:
javax.xml.bind.annotation Annotation Type
XmlInlineBinaryData

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, TYPE})
public @interface XmlInlineBinaryData

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, TYPE})

XML base64-encoded XOP

XOP isXOPPackage() base64-encoded ImageSource
byte[] XOP JAXB JAXB
addMtomAttachment(DataHandler, String, String)
addMtomAttachment(byte[], int, int, String, String, String)
since JAXB2.0

Disable consideration of XOP encoding for datatypes that are bound to base64-encoded binary data in XML.

When XOP encoding is enabled as described in AttachmentMarshaller.isXOPPackage(), this annotation disables datatypes such as Image or Source or byte[] that are bound to base64-encoded binary from being considered for XOP encoding. If a JAXB property is annotated with this annotation or if the JAXB property's base type is annotated with this annotation, neither AttachmentMarshaller.addMtomAttachment(DataHandler, String, String) nor AttachmentMarshaller.addMtomAttachment(byte[], int, int, String, String, String) is ever called for the property. The binary data will always be inlined.

Since: JAXB2.0
Author:
Joseph Fialli

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
public abstract class **XMLInputFactory**

extends **Object**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.stream.isValidating</td>
<td>/ DTD</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isNamespaceAware</td>
<td>/ XML 1.0</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isCoalescing</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isReplacingEntityReferences</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isSupportingExternalEntities</td>
<td></td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.supportDTD</td>
<td></td>
<td>DTD</td>
</tr>
<tr>
<td>javax.xml.stream.reporter</td>
<td>/ XMLReporter</td>
<td>javax.x</td>
</tr>
<tr>
<td>javax.xml.stream.resolver</td>
<td>/ XMLResolver</td>
<td>javax.x</td>
</tr>
<tr>
<td>javax.xml.streamallocator</td>
<td>/ XMLEventAllocator</td>
<td>javax.x</td>
</tr>
</tbody>
</table>

See also

javax.xml.stream.util.XMLEventAllocator

Defines an abstract implementation of a factory for getting streams. The following table defines the standard properties of this specification. Each property varies in the level of support required by each implementation. The level of support required is described in the 'Required' column.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Behavior</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.stream.isValidating</td>
<td>Turns on/off implementation specific DTD validation</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isNamespaceAware</td>
<td>Turns on/off namespace processing for XML 1.0 support</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isCoalescing</td>
<td>Requires the processor to coalesce adjacent character data</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isReplacingEntityReferences</td>
<td>replace internal entity references with their replacement text and report them as characters</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.isSupportingExternalEntities</td>
<td>Resolve external parsed entities</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.supportDTD</td>
<td>Use this property to request processors that do not support DTDs</td>
<td>Boolean</td>
</tr>
<tr>
<td>javax.xml.stream.reporter</td>
<td>sets/gets the impl of the XMLReporter</td>
<td>javax.x</td>
</tr>
</tbody>
</table>
javax.xml.stream.resolver

<table>
<thead>
<tr>
<th>Sets/gets the impl of the XMLResolver interface</th>
</tr>
</thead>
</table>

javax.xml.stream.allocator

<table>
<thead>
<tr>
<th>Sets/gets the impl of the XMLEventAllocator interface</th>
</tr>
</thead>
</table>

**Version:**

1.0

**Author:**

Copyright (c) 2003 by BEA Systems. All Rights Reserved.

**See Also:**

XMLOutputFactory, XMLEventReader, XMLStreamReader, EventFilter, XMLReporter, XMLResolver, XMLEventAllocator

### Field Summary

<table>
<thead>
<tr>
<th>static String</th>
<th>ALLOCATOR</th>
<th>The property used to set/get the implementation of the allocator</th>
</tr>
</thead>
<tbody>
<tr>
<td>static String</td>
<td>IS_COALESCING</td>
<td>The property that requires the parser to coalesce adjacent character data sections</td>
</tr>
<tr>
<td>static String</td>
<td>IS_NAMESPACE_AWARE</td>
<td>The property used to turn on/off namespace support, this is to support XML 1.0 documents, only the true setting must be supported</td>
</tr>
<tr>
<td>static String</td>
<td>IS_REPLACING_ENTITY_REFERENCES</td>
<td>Requires the parser to replace internal entity references with their replacement text and report them as characters</td>
</tr>
<tr>
<td>static String</td>
<td>IS_SUPPORTING_EXTERNAL_ENTITIES</td>
<td>The property that requires the parser to resolve external parsed entities</td>
</tr>
</tbody>
</table>
### Constructor Summary

| protected | XMLInputFactory() |

### Method Summary

<table>
<thead>
<tr>
<th>abstract</th>
<th>XMLEventReader</th>
<th>createFilteredReader(XMLEventReader reader, EventFilter filter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>abstract</td>
<td>XMLStreamReader</td>
<td>createFilteredReader(XMLStreamReader reader, StreamFilter filter)</td>
</tr>
<tr>
<td>abstract</td>
<td>XMLEventReader</td>
<td>createXMLEventReader(InputStream stream)</td>
</tr>
<tr>
<td>abstract</td>
<td>XMLEventReader</td>
<td>createXMLEventReader(InputStream stream, String encoding)</td>
</tr>
<tr>
<td>abstract</td>
<td>XMLEventReader</td>
<td>createXMLEventReader(Reader reader)</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventReader(Source source)</code></td>
<td>Create a new XMLEventReader from a JAXP source.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventReader(String systemId, InputStream stream)</code></td>
<td>Create a new XMLEventReader from a java.io.InputStream</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventReader(String systemId, Reader reader)</code></td>
<td>Create a new XMLEventReader from a reader</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventReader(XMLStreamReader reader)</code></td>
<td>Create a new XMLEventReader from an XMLStreamReader.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(InputStream stream)</code></td>
<td>Create a new XMLStreamReader from a java.io.InputStream</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(InputStream stream, String encoding)</code></td>
<td>Create a new XMLStreamReader from a java.io.InputStream with a specific encoding</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(Reader reader)</code></td>
<td>Create a new XMLStreamReader from a reader</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(Source source)</code></td>
<td>Create a new XMLStreamReader from a JAXP source.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(String systemId, InputStream stream)</code></td>
<td>Create a new XMLStreamReader from a java.io.InputStream</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamReader(String systemId, Reader reader)</code></td>
<td>Create a new XMLStreamReader from a reader</td>
<td></td>
</tr>
<tr>
<td><code>getEventAllocator()</code></td>
<td>Gets the allocator used by streams created with this factory</td>
<td></td>
</tr>
</tbody>
</table>
abstract Object **get**Property**(String name)**  
Get the value of a feature/property from the underlying implementation

abstract XMLReporter **getXMLReporter**( )  
The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

abstract XMLResolver **getXMLResolver**( )  
The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

abstract boolean **isPropertySupported**(String name)  
Query the set of properties that this factory supports.

static XMLInputFactory **newInstance**( )  
Create a new instance of the factory.

static XMLInputFactory **newInstance**(String factoryId, ClassLoader classLoader)  
Create a new instance of the factory

abstract void **setEventAllocator**(XMLEventAllocator allocator)  
Set a user defined event allocator for events

abstract void **setProperty**(String name, Object value)  
Allows the user to set specific feature/property on the underlying implementation.

abstract void **setXMLReporter**(XMLReporter reporter)  
The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

abstract void **setXMLResolver**(XMLResolver resolver)  
The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

**Methods inherited from class java.lang.Object**

*clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait*
IS_NAMESPACE_AWARE

public static final String IS_NAMESPACE_AWARE

The property used to turn on/off namespace support, this is to support XML 1.0 documents, only the true setting must be supported

See Also:
Constant Field Values

IS_VALIDATING

public static final String IS_VALIDATING

The property used to turn on/off implementation specific validation

See Also:
Constant Field Values

IS_COALESCING

public static final String IS_COALESCING

The property that requires the parser to coalesce adjacent character data sections

See Also:
### Constant Field Values

**IS_REPLACING_ENTITY_REFERENCES**

```java
public static final String IS_REPLACING_ENTITY_REFERENCES
```

Requires the parser to replace internal entity references with their replacement text and report them as characters.

**See Also:**
- [Constant Field Values](#)

---

**IS_SUPPORTING_EXTERNAL_ENTITIES**

```java
public static final String IS_SUPPORTING_EXTERNAL_ENTITIES
```

The property that requires the parser to resolve external parsed entities.

**See Also:**
- [Constant Field Values](#)

---

**SUPPORT_DTD**

```java
public static final String SUPPORT_DTD
```

The property that requires the parser to support DTDs.

**See Also:**
- [Constant Field Values](#)
REPORTER

public static final String REPORTER

    The property used to set/get the implementation of the XMLReporter interface

    See Also:
    Constant Field Values

RESOLVER

public static final String RESOLVER

    The property used to set/get the implementation of the XMLResolver

    See Also:
    Constant Field Values

ALLOCATOR

public static final String ALLOCATOR

    The property used to set/get the implementation of the allocator

    See Also:
    Constant Field Values

Constructor Detail

protected XMLInputFactory()
public static XMLInputFactory newInstance() throws FactoryConfigurationError

Create a new instance of the factory. This static method creates a new factory instance. This method uses the following ordered lookup procedure to determine the XMLInputFactory implementation class to load: Use the javax.xml.stream.XMLInputFactory system property. Use the properties file "lib/stax.properties" in the JRE directory. This configuration file is in standard java.util.Properties format and contains the fully qualified name of the implementation class with the key being the system property defined above. Use the Services API (as detailed in the JAR specification), if available, to determine the classname. The Services API will look for a classname in the file META-INF/services(javax.xml.stream.XMLInputFactory) in jars available to the runtime. Platform default XMLInputFactory instance.
Once an application has obtained a reference to a XMLInputFactory it can use the factory to configure and obtain stream instances.

Throws:

- `FactoryConfigurationError` - if an instance of this factory cannot be loaded

```java
public static XMLInputFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationError

factoryId
classLoader
return
```

Throws `FactoryConfigurationError`:

```java
newInstance
public static XMLInputFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationError

factoryId - Name of the factory to find, same as a property name
classLoader - classLoader to use

Returns:
the factory implementation

Throws:
- `FactoryConfigurationError` - if an instance of this factory cannot be loaded

abstract public XMLStreamReader
```
createXMLStreamReader(java.io.Reader reader) throws XMLStreamException

reader  XMLStreamReader

Create a new XMLStreamReader from a reader

Parameters:
reader - the XML data to read from

Throws:
XMLStreamException

createXMLStreamReader(javax.xml.transform.Source source) throws XMLStreamException

JAXP  XMLStreamReader

source  XMLEventReader

Create a new XMLStreamReader from a JAXP source. This method is optional.

Parameters:
source - the source to read from

Throws:
UnsupportedOperationException - if this method is not supported by this XMLInputFactory
XMLStreamException

abstract public XMLStreamReader createXMLStreamReader(java.io.InputStream stream)
throws XMLStreamException

java.io.InputStream  XMLStreamReader
stream

Throws
XMLStreamException:

createXMLStreamReader

public abstract XMLStreamReader createXMLStreamReader(InputStream stream)
throws XMLStreamException

Create a new XMLStreamReader from a java.io.InputStream

Parameters:
stream - the InputStream to read from

Throws:
XMLStreamException

abstract public XMLStreamReader createXMLStreamReader(java.io.InputStream stream, String encoding) throws XMLStreamException

java.io.InputStream  XMLStreamReader
stream  encoding

Throws
XMLStreamException:
createXMLStreamReader

```java
public abstract XMLStreamReader createXMLStreamReader(InputStream stream,
String encoding)
throws XMLStreamException
```

Create a new XMLStreamReader from a java.io.InputStream

**Parameters:**
- `stream` - the InputStream to read from
- `encoding` - the character encoding of the stream

**Throws:**
- XMLStreamException

---

abstract public XMLStreamReader
createXMLStreamReader(String systemId,
java.io.InputStream stream) throws XMLStreamException
java.io.InputStream XMLStreamReader

```java
    systemId ID
    stream XMLEventReader InputStream
```

createXMLStreamReader

```java
public abstract XMLStreamReader createXMLStreamReader(String systemId,
InputStream stream)
throws XMLStreamException
```

Create a new XMLStreamReader from a java.io.InputStream

**Parameters:**
- `systemId` - the system ID of the stream
- `stream` - the InputStream to read from

**Throws:**
- XMLStreamException

---

abstract public XMLStreamReader
createXMLStreamReader(String systemId, java.io.Reader reader) throws XMLStreamException
java.io.InputStream XMLStreamReader

Create a new XMLStreamReader from a java.io.InputStream

Parameters:
- systemId - the system ID of the stream
- reader - the InputStream to read from

Throws:
XMLStreamException

abstract public XMLEventReader createXMLEventReader(java.io.Reader reader) throws XMLStreamException

Create a new XMLEventReader from a reader

Parameters:
abstract public XMLEventReader createXMLEventReader(String systemId, java.io.Reader reader) throws XMLStreamException
reader XMLEventReader

Create a new XMLEventReader from a reader

Parameters:
  systemId - the system ID of the input
  reader - the XML data to read from

Throws:
  XMLStreamException

abstract public XMLEventReader createXMLEventReader(XMLStreamReader reader) throws XMLStreamException

Create a new XMLEventReader from a reader

Parameters:
  reader - the XML data to read from

Throws:
  XMLStreamException
createXMLEventReader

```java
public abstract XMLEventReader createXMLEventReader(XMLStreamReader reader) throws XMLStreamException

Create a new XMLEventReader from an XMLStreamReader. After being used to construct the XMLEventReader instance returned from this method the XMLStreamReader must not be used.

Parameters:
  reader - the XMLStreamReader to read from (may not be modified)

Returns:
  a new XMLEventReader

Throws:
  XMLStreamException
```

abstract public XMLEventReader createXMLEventReader(javax.xml.transform.Source source) throws XMLStreamException

JAXP XMLEventReader
```java
  source XMLEventReader
  Throws UnsupportedOperationException: XMLInputFactory
```

createXMLEventReader

```java
public abstract XMLEventReader createXMLEventReader(Source source) throws XMLStreamException

Create a new XMLEventReader from a JAXP source. Support of this method is optional.

Parameters:
  source - the source to read from

Throws:
UnsupportedOperationException - if this method is not supported by this XMLInputFactory

XMLStreamException

abstract public XMLEventReader createXMLEventReader(java.io.InputStream stream) throws XMLStreamException

java.io.InputStream XMLEventReader

Create a new XMLEventReader from a java.io.InputStream

Parameters:
stream - the InputStream to read from

Throws:
XMLStreamException

abstract public XMLEventReader createXMLEventReader(java.io.InputStream stream, String encoding) throws XMLStreamException

java.io.InputStream XMLEventReader

Create a new XMLEventReader from a java.io.InputStream

Parameters:
stream - the InputStream to read from
encoding - the encoding to use

Throws:
XMLStreamException

createXMLEventReader
public abstract XMLEventReader createXMLEventReader(InputStream stream, String encoding) throws XMLStreamException

Create a new XMLEventReader from a java.io.InputStream

Parameters:
- stream - the InputStream to read from
- encoding - the character encoding of the stream

Throws:
- XMLStreamException

abstract public XMLEventReader createXMLEventReader(String systemId, java.io.InputStream stream) throws XMLStreamException
java.io.InputStream XMLEventReader

Create a new XMLEventReader from a java.io.InputStream

Parameters:
- systemId - the system ID of the stream
- stream - the InputStream to read from

Throws:
- XMLStreamException

abstract public XMLStreamReader createFilteredReader(XMLStreamReader reader,
StreamFilter filter) throws XMLStreamException
reader reader

Throws

createFilteredReader

public abstract XMLStreamReader createFilteredReader(StreamFilter filter)
throws XMLStreamException:

Create a filtered reader that wraps the filter around the reader

Parameters:
reader - the reader to filter
filter - the filter to apply to the reader

Throws:
XMLStreamException

abstract public XMLEventReader createFilteredReader(XMLEventReader reader, EventFilter filter)
throws XMLStreamException
reader reader

Throws

createFilteredReader

public abstract XMLEventReader createFilteredReader(XMLEventReader reader, EventFilter filter)
throws XMLStreamException:

Create a filtered event reader that wraps the filter around the event

---
Parameters:
reader - the event reader to wrap
filter - the filter to apply to the event reader

Throws:
XMLStreamException

abstract public XMLResolver getXMLResolver()
XMLStreamReader  XMLEventReader

getXMLResolver

public abstract XMLResolver getXMLResolver()

The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

abstract public void setXMLResolver(XMLResolver resolver)
XMLStreamReader  XMLEventReader
resolver

setXMLResolver

public abstract void setXMLResolver(XMLResolver resolver)

The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

Parameters:
resolver - the resolver to use to resolve references
abstract public XMLReporter getXMLReporter()
XMLStreamReader  XMLEventReader

getXMLReporter

public abstract XMLReporter getXMLReporter()

The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

abstract public void setXMLReporter(XMLReporter reporter)
XMLStreamReader  XMLEventReader

setXMLReporter

public abstract void setXMLReporter(XMLReporter reporter)

The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

Parameters:

reporter - the resolver to use to report non fatal errors

abstract public void setProperty(String name, Object value)
throws IllegalArgumentException
/ IllegalArgumentException

name null
value

Throws IllegalArgumentException:

IllegalArgumentOutOfRangeException:
setProperty

```java
public abstract void setProperty(String name, Object value)
    throws IllegalArgumentException
```

Allows the user to set specific feature/property on the underlying implementation. The underlying implementation is not required to support every setting of every property in the specification and may use IllegalArgumentException to signal that an unsupported property may not be set with the specified value.

**Parameters:**
- `name` - The name of the property (may not be null)
- `value` - The value of the property

**Throws:**
- `IllegalArgumentException` - if the property is not supported

---

abstract public Object getProperty(String name) throws IllegalArgumentException

```java
Object getProperty(String name)
```

Get the value of a feature/property from the underlying implementation

**Parameters:**
- `name` - The name of the property (may not be null)

**Returns:**
- The value of the property

---

getProperty

```java
public abstract Object getProperty(String name)
    throws IllegalArgumentException
```

Get the value of a feature/property from the underlying implementation

**Parameters:**
- `name` - The name of the property (may not be null)

**Returns:**
- The value of the property
abstract public boolean isPropertySupported(String name)

    name null
    return true false

isPropertySupported

public abstract boolean isPropertySupported(String name)

    Query the set of properties that this factory supports.

    Parameters:
        name - The name of the property (may not be null)

    Returns:
        true if the property is supported and false otherwise

abstract public void setEventAllocator(XMLEventAllocator allocator)

    allocator

setEventAllocator

public abstract void setEventAllocator(XMLEventAllocator allocator)

    Set a user defined event allocator for events

    Parameters:
        allocator - the user defined allocator
abstract public XMLEventAllocator getEventAllocator()
javax.xml.bind.annotation.adapters  Annotation Type

XmlJavaTypeAdapter

@Retention(value=RUNTIME)
@Target(value={PACKAGE, FIELD, METHOD, TYPE, PARAMETER})
public @interface XmlJavaTypeAdapter

Implements: Annotation
Inner classes: XmlJavaTypeAdapter.DEFAULT

@Retention(value=RUNTIME)
@Target(value={PACKAGE, FIELD, METHOD, TYPE, PARAMETER})

 XmlAdapter

@XmlJavaTypeAdapter

- JavaBean
- XmlAccessorType
- XmlSchema
- XmlSchemaType
- XmlSchemaTypes

XmlElement
XmlAttribute
XmlElementRef
XmlElementRefs
XmlAnyElement

XmlElement.XmlAttribute.XmlElementRef.XmlElementRefs.XmlAnyElement.XmlAccessorType.XmlSchema.XmlSchemaType.XmlSchemaTypes

XmlAdapter
Use an adapter that implements XMLAdapter for custom marshaling.

Usage:

The @XmlJavaTypeAdapter annotation can be used with the following program elements:

- a JavaBean property
- field
- parameter
- package
- from within XmlJavaTypeAdapters

When @XmlJavaTypeAdapter annotation is defined on a class, it applies to all references to the class.

When @XmlJavaTypeAdapter annotation is defined at the package level it applies to all references from within the package to @XmlJavaTypeAdapter.type().

When @XmlJavaTypeAdapter annotation is defined on the field, property or parameter, then the annotation applies to the field, property or the parameter only.

A @XmlJavaTypeAdapter annotation on a field, property or parameter overrides the @XmlJavaTypeAdapter annotation associated with the class being referenced by the field, property or parameter.

A @XmlJavaTypeAdapter annotation on a class overrides the @XmlJavaTypeAdapter annotation specified at the package level for that class.

This annotation can be used with the following other annotations:
**Required Element Summary**

<table>
<thead>
<tr>
<th>Class&lt;? extends XmlAdapter&gt;</th>
<th>value</th>
</tr>
</thead>
</table>
|                              | Points to the class that converts a value type to a bound type or vice versa.

**Optional Element Summary**

<table>
<thead>
<tr>
<th>Class</th>
<th>type</th>
</tr>
</thead>
</table>
|       | If this annotation is used at the package level, then value of the type() must be specified.

**Element Detail**

abstract public Class<T> value()

value bound bound value

class value
public abstract Class<? extends XmlAdapter> value

Points to the class that converts a value type to a bound type or vice versa. See XmlAdapter for more details.

abstract public Class<T> type()

type()

public abstract Class type

If this annotation is used at the package level, then value of the type() must be specified.

Default:
javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter.DEFAULT.class
Class
javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter.DEFAULT

java.lang.Object
javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter.DEFAULT

Enclosing class:
  XmlJavaTypeAdapter

public static final class XmlJavaTypeAdapter.DEFAULT
extends Object

Contained within: XmlJavaTypeAdapter
type()

Used in XmlJavaTypeAdapter.type() to signal that the type be inferred from the signature of the field, property, parameter or the class.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlJavaTypeAdapter.DEFAULT()</td>
</tr>
</tbody>
</table>

Method Summary

Methods inherited from class java.lang.Object
cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Constructor Detail


public XmlJavaTypeAdapter.DEFAULT()

XmlJavaTypeAdapter.DEFAULT

public XmlJavaTypeAdapter.DEFAULT()
javax.xml.bind.annotation.adapters  Annotation Type
XmlJavaTypeAdapters

@Retention(value=RUNTIME)
@Target(value=PACKAGE)
public @interface XmlJavaTypeAdapters

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=PACKAGE)

@XmlJavaTypeAdapters

@XmlJavaTypeAdapter

@XmlJavaTypeAdapters

@XmlJavaTypeAdapters

javax.xml.bind.package javadoc ""
since        JAXB2.0
See also     javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter

A container for multiple @XmlJavaTypeAdapter annotations.

Multiple annotations of the same type are not allowed on a program
element. This annotation therefore serves as a container annotation for
multiple @XmlJavaTypeAdapter as follows:

@XmlJavaTypeAdapters

The @XmlJavaTypeAdapters annotation is useful for defining
XmlJavaTypeAdapter annotations for different types at the package level.

Since:
    JAXB2.0
Author:
    • Sekhar Vajjhala, Sun Microsystems, Inc.
See Also:
    XmlJavaTypeAdapter

---

### Required Element Summary

<table>
<thead>
<tr>
<th>XmlJavaTypeAdapter[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Collection of @XmlJavaTypeAdapter annotations</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public `XmlJavaTypeAdapter[]` value()

@`XmlJavaTypeAdapter`

value

public abstract `XmlJavaTypeAdapter[]` value

    Collection of @`XmlJavaTypeAdapter` annotations

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
javax.xml.bind.annotation Annotation Type XmlList

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})
public @interface XmlList

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})

@XmlList

- JavaBean

@XmlElement

@XmlElement
@XmlRootElement
class Foo {
    @XmlElement
    List<String> data;
}

XML

&lt;foo&gt;
&lt;data&gt;abc&lt;/data&gt;
&lt;data&gt;def&lt;/data&gt;
&lt;/foo&gt;

@XmlList

@XmlRootElement
class Foo {
    @XmlElement
    @XmlList
List<String> data;
}

XML

&lt;foo&gt;
 &lt;data&gt;abc def&lt;/data&gt;
 &lt;/foo&gt;

XmlElement XmlAttribute XmlValue XmlIDREF

- @XmlList XmlValue XmlList XmlValue
- @XmlList XmlAttribute XmlList XmlAttribute

since JAXB2.0

Used to map a property to a list simple type.

Usage

The @XmlList annotation can be used with the following program elements:

- JavaBean property
- field

When a collection property is annotated just with @XmlElement, each item in the collection will be wrapped by an element. For example,

@XmlRootElement
class Foo {
    @XmlElement
    List<String> data;
}

would produce XML like this:
@XmlList annotation, on the other hand, allows multiple values to be represented as whitespace-separated tokens in a single element. For example,

```java
@XmlElement
@XmlList
List<String> data;
```

the above code will produce XML like this:

```xml
<foo>
  <data>abc def</data>
</foo>
```

This annotation can be used with the following annotations: `XmlElement`, `XmlAttribute`, `XmlValue`, `XmlIDREF`.

- The use of `@XmlList` with `XmlValue` while allowed, is redundant since `XmlList` maps a collection type to a simple schema type that derives by list just as `XmlValue` would.
- The use of `@XmlList` with `XmlAttribute` while allowed, is redundant since `XmlList` maps a collection type to a simple schema type that derives by list just as `XmlAttribute` would.

Since: JAXB2.0

Author:
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Sekhar Vajjhala, Sun Microsystems, Inc.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

PREV CLASS  NEXT CLASS
SUMMARY: REQUIRED | OPTIONAL
FRAMES | NO FRAMES
DETAIL: ELEMENT
javax.xml.bind.annotation Annotation Type XmlMimeType

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})
public @interface XmlMimeType

Implements: Annotation  
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PARAMETER})

XML MIME

XML base64-encoded  

xmime:contentType  
since  

JAXB2.0

Associates the MIME type that controls the XML representation of the property.

This annotation is used in conjunction with datatypes such as Image or Source that are bound to base64-encoded binary in XML.

If a property that has this annotation has a sibling property bound to the xmime:contentType attribute, and if in the instance the property has a value, the value of the attribute takes precedence and that will control the marshalling.

Since:  
JAXB2.0

Author:  
Kohsuke Kawaguchi

---

Required Element Summary
The textual representation of the MIME type, such as "image/jpeg" "image/*", "text/xml; charset=iso-8859-1" and so on.

### Element Detail

**abstract public String value()**

```java
MIME "image/jpeg" "image/*" "text/xml; charset=iso-8859-1"
```

**value**

```java
public abstract String value
```

The textual representation of the MIME type, such as "image/jpeg" "image/*", "text/xml; charset=iso-8859-1" and so on.

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
javax.xml.bind.annotation Annotation Type XmlMixed

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlMixed

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean

- @XmlElementRef @XmlElementRefs @XmlAnyElement

@XmlMixed

- XML java.lang.String
- JAXB @XmlRootElement
- JAXB Element @XmlAnyElement

&lt;!-- schema fragment having mixed content --&gt;
&lt;xs:complexType name="letterBody" mixed="true"&gt;
  &lt;xs:sequence&gt;
    &lt;xs:element name="name" type="xs:string"&gt;
      &lt;/xs:element&gt;&lt;xs:element name="quantity" type="xs:positiveInteger"&gt;
      &lt;/xs:element&gt;&lt;xs:element name="productName" type="xs:string"&gt;
      &lt;/xs:element&gt;&lt;!-- etc. --&gt;
  &lt;/xs:sequence&gt;
&lt;/xs:complexType&gt;

// Schema-derived Java code:
// (Only annotations relevant to mixed content are shown below, others are omitted.)
import java.math.BigInteger;
public class ObjectFactory {
// element instance factories
JAXBElement<letterbody> createLetterBody(LetterBody va
JAXBElement&lt;string&gt; createLetterBodyName(String va
JAXBElement&lt;biginteger&gt; createLetterBodyQuantity(BigInteger	value);
JAXBElement&lt;/biginteger&gt;&lt;/string&gt;&lt;/string&gt; createLetterBodyProductName(String	value);
// type instance factory
LetterBody&}; createLetterBody();
}

public class LetterBody {
    // Mixed content can contain instances of Element classes
    // Name, Quantity and ProductName. Text data is represented
    // java.util.String for text.
    @XmlMixed
    @XmlElementRefs({
        @XmlElementRef(name="productName", type=JAXBElement.class),
        @XmlElementRef(name="quantity", type=JAXBElement.class),
        @XmlElementRef(name="name", type=JAXBElement.class)
    })
    List getContent(){...}
}

XML

&lt;letterbody&gt;
Dear Mr.&lt;name&gt;Robert Smith&lt;/name&gt;
Your order of &lt;quantity&gt;1&lt;/quantity&gt; &lt;productname&gt;Baby Monitor&lt;/productname&gt; shipped from our warehouse. ....
&lt;/letterbody&gt;

JAXB API

LetterBody lb = ObjectFactory.createLetterBody();
JAXBElement&lt;letterbody&gt; lbe = ObjectFactory.createLetterBody(lb); List gcl = lb.getContent(); //add mixed content to general content
gcl.add(&amp;quot;Dear Mr.&amp;quot;); // add text information item

// add child element information item
    gcl.add(ObjectFactory.createLetterBodyName(&amp;quot;Robert Smith&amp;quot;));
gcl.add(&amp;quot;Your order of &amp;quot;); // add text information

// add children element information items
    gcl.add(ObjectFactory.
        createLetterBodyQuantity(new BigInteger(&amp;
    gcl.add(ObjectFactory.createLetterBodyProductName(&amp;quot;Baby Monitor&amp;quot;)); // add text
Annotate a JavaBean multi-valued property to support mixed content.

The usage is subject to the following constraints:

- can be used with @XmlElementRef, @XmlElementRefs or @XmlAnyElement

The following can be inserted into @XmlMixed annotated multi-valued property

- XML text information items are added as values of java.lang.String.
- Children element information items are added as instances of JAXBElement or instances with a class that is annotated with @XmlRootElement.
- Unknown content that is not be bound to a JAXB mapped class is inserted as Element. (Assumes property annotated with @XmlAnyElement)

Below is an example of binding and creation of mixed content.

```xml
&lt;!-- schema fragment having mixed content --&gt;
&lt;xs:complexType name="letterBody" mixed="true"><
  &lt;xs:sequence&
    &lt;xs:element name="name" type="xs:string"&
    &lt;/xs:element>&
    &lt;xs:element name="quantity" type="xs:positiveInteger"&
    &lt;/xs:element>&
    &lt;!-- etc. --&gt;
  &lt;/xs:sequence&
&lt;/xs:complexType&

// Schema-derived Java code:
// (Only annotations relevant to mixed content are shown below,
// others are ommitted.)
import java.math.BigInteger;
public class ObjectFactory {
```
The following is an XML instance document with mixed content

```xml
<letterbody>
  Dear Mr.<name>Robert Smith</name>
  Your order of <quantity>1</quantity> <productname>Baby Monitor</productname> shipped from our warehouse. ....
</letterbody>
```

that can be constructed using following JAXB API calls.

```java
LetterBody lb = ObjectFactory.createLetterBody();
JAXBElement<letterbody> lbe = ObjectFactory.createLetterBody(lb);
List gcl = lb.getContent();  //add mixed content to general content
  gcl.add("Dear Mr.");  // add text information item as a String.
  // add child element information item
  gcl.add(ObjectFactory.createLetterBodyName("Robert Smith"));
  gcl.add("Your order of ");  // add text information item as a String
  // add children element information items
  gcl.add(ObjectFactory.createLetterBodyQuantity(new BigInteger("1")));
  gcl.add(ObjectFactory.createLetterBodyProductName("Baby Monitor"));
  gcl.add("shipped from our warehouse");  // add text information item
```

Since:
   JAXB2.0

Author:
   Kohsuke Kawaguchi
Annotation Type XmlNs

@Retention(value=RUNTIME)
@Target(value={})
public @interface XmlNs

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={})

XML URI

@XmlNs

javax.xml.bind.package javadoc ""

XmlSchema
    since JAXB2.0
    version $Revision: 1.3 $

 Associates a namespace prefix with a XML namespace URI.

Usage

@XmlNs annotation is intended for use from other program annotations.


Example: See XmlSchema annotation type for an example.

Since: JAXB2.0
Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>namespaceURI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Namespace URI</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Namespace prefix</td>
</tr>
</tbody>
</table>

Element Detail

abstract public String prefix()

prefix

public abstract String prefix

   Namespace prefix

abstract public String namespaceURI()

namespaceURI

public abstract String namespaceURI

   Namespace URI
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
<table>
<thead>
<tr>
<th>PREV CLASS</th>
<th>NEXT CLASS</th>
<th>SUMMARY: NESTED</th>
<th>ENUM CONSTANTS</th>
<th>FIELD</th>
<th>METHOD</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>DETAIL</td>
<td>ENUM CONSTANTS</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
javax.xml.bind.annotation Enum XmlNsForm

java.lang.Object
   ↓ java.lang.Enum<XmlNsForm>
      ↓ javax.xml.bind.annotation.XmlNsForm

All Implemented Interfaces:
   Serializable, Comparable<XmlNsForm>

public enum XmlNsForm
extends Enum<XmlNsForm>

Extends: Enum<E>

XML

javax.xml.bind.package javadoc ""

<table>
<thead>
<tr>
<th></th>
<th>XML</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNQUALIFIED</td>
<td></td>
</tr>
<tr>
<td>QUALIFIED</td>
<td></td>
</tr>
<tr>
<td>UNSET</td>
<td>XML</td>
</tr>
</tbody>
</table>

since
version

JAXB2.0
$Revision: 1.2 $

Enumeration of XML Schema namespace qualifications.

Usage

The namespace qualification values are used in the annotations defined in this package. The enumeration values are mapped as follows:

<table>
<thead>
<tr>
<th>Enum Value</th>
<th>XML Schema Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNQUALIFIED</td>
<td>unqualified</td>
</tr>
<tr>
<td>QUALIFIED</td>
<td>qualified</td>
</tr>
<tr>
<td>UNSET</td>
<td>namespace qualification attribute is absent from the XML Schema fragment</td>
</tr>
</tbody>
</table>

Since: JAXB2.0

Version: $Revision: 1.2$

Author: Sekhar Vajjhala, Sun Microsystems, Inc.

---

### Enum Constant Summary

<table>
<thead>
<tr>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>QUALIFIED</td>
</tr>
<tr>
<td>UNQUALIFIED</td>
</tr>
<tr>
<td>UNSET</td>
</tr>
</tbody>
</table>

### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static XmlNsForm.valueOf(String name)</td>
<td>Returns the enum constant of this type with the specified name.</td>
</tr>
<tr>
<td>static XmlNsForm[] values()</td>
<td>Returns an array containing the constants of this enum type, in the order they're declared.</td>
</tr>
<tr>
<td>Methods inherited from class java.lang.Enum</td>
<td></td>
</tr>
<tr>
<td>clone, compareTo, equals, getDeclaringClass, hashCode, name, ordinal, toString, valueOf</td>
<td></td>
</tr>
</tbody>
</table>

| Methods inherited from class java.lang.Object |
| finalize, getClass, notify, notifyAll, wait, wait, wait |

**Enum Constant Detail**

**UNQUALIFIED**

public static final `XmlNsForm` UNQUALIFIED

**QUALIFIED**

public static final `XmlNsForm` QUALIFIED

**UNSET**

public static final `XmlNsForm` UNSET

**Method Detail**

final public static `XmlNsForm[]` values()
values

public static final XmlNsForm[] values()

Returns an array containing the constants of this enum type, in the order they're declared. This method may be used to iterate over the constants as follows:

for(XmlNsForm c : XmlNsForm.values())
    System.out.println(c);

Returns:
an array containing the constants of this enum type, in the order they're declared

valueOf

public static XmlNsForm valueOf(String name)

valueOf

Returns the enum constant of this type with the specified name. The string must match exactly an identifier used to declare an enum constant in this type. (Extraneous whitespace characters are not permitted.)

Parameters:
   name - the name of the enum constant to be returned.

Returns:
   the enum constant with the specified name

Throws:
   IllegalArgumentException - if this enum type has no constant with the specified name
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.stream  **Class XMLOutputFactory**

java.lang.Object
   \javax.xml.stream.XMLOutputFactory

public abstract class XMLOutputFactory
   extends Object

   XMLEventWriter  XMLStreamWriter ""

t javax.xml.stream.isRepairingNamespaces
   Boolean False

   setProperty("javax.xml.stream.isRepairingNamespaces",new
   Boolean(true|false));

   writer false

   writer  RepairingNamespaces StartElement
   StartElement URI StartElement StartElement
   defaultNamespace URI StartElement QName

   URI

   / URI URI

   URI  URI

   URI  URI

   version  1.0

   See also  javax.xml.stream.XMLInputFactory,
   javax.xml.stream.XMLEventWriter,
   javax.xml.stream.XMLStreamWriter
Defines an abstract implementation of a factory for getting XMLEventWriters and XMLStreamWriters. The following table defines the standard properties of this specification. Each property varies in the level of support required by each implementation. The level of support required is described in the 'Required' column.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Behavior</th>
<th>Return type</th>
<th>Default Value</th>
<th>Required</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.stream.isRepairingNamespaces</td>
<td>defaults prefixes on the output side</td>
<td>Boolean</td>
<td>False</td>
<td>Yes</td>
</tr>
</tbody>
</table>

The following paragraphs describe the namespace and prefix repair algorithm:

The property can be set with the following code line:

```java
setProperty("javax.xml.stream.isRepairingNamespaces", new Boolean(true|false));
```

This property specifies that the writer default namespace prefix declarations. The default value is false.

If a writer isRepairingNamespaces it will create a namespace declaration on the current StartElement for any attribute that does not currently have a namespace declaration in scope. If the StartElement has a uri but no prefix specified a prefix will be assigned, if the prefix has not been declared in a parent of the current StartElement it will be declared on the current StartElement. If the defaultNamespace is bound and in scope and the default namespace matches the URI of the attribute or StartElement QName no prefix will be assigned.

If an element or attribute name has a prefix, but is not bound to any namespace URI, then the prefix will be removed during serialization.

If element and/or attribute names in the same start or empty-element tag
are bound to different namespace URIs and are using the same prefix then the element or the first occurring attribute retains the original prefix and the following attributes have their prefixes replaced with a new prefix that is bound to the namespace URIs of those attributes.

If an element or attribute name uses a prefix that is bound to a different URI than that inherited from the namespace context of the parent of that element and there is no namespace declaration in the context of the current element then such a namespace declaration is added.

If an element or attribute name is bound to a prefix and there is a namespace declaration that binds that prefix to a different URI then that namespace declaration is either removed if the correct mapping is inherited from the parent context of that element, or changed to the namespace URI of the element or attribute using that prefix.

Version:
    1.0
Author:
    Copyright (c) 2003 by BEA Systems. All Rights Reserved. 
See Also:
    XMLInputFactory, XMLEventWriter, XMLStreamWriter

<table>
<thead>
<tr>
<th>Field Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>static String</strong> IS_REPAIRING_NAMESPACES</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected XMLOutputFactory()</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>createXMLEventWriter(OutputStream stream)</td>
</tr>
</tbody>
</table>
### `createXMLStreamWriter` Method Variants

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createXMLStreamWriter(OutputStream stream)</code></td>
<td>Create a new <code>XMLStreamWriter</code> that writes to a stream.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamWriter(OutputStream stream, String encoding)</code></td>
<td>Create a new <code>XMLStreamWriter</code> that writes to a stream.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamWriter(Result result)</code></td>
<td>Create a new <code>XMLStreamWriter</code> that writes to a JAXP result.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLStreamWriter(Writer stream)</code></td>
<td>Create a new <code>XMLStreamWriter</code> that writes to a writer.</td>
<td></td>
</tr>
</tbody>
</table>

### `createXMLEventWriter` Method Variants

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>createXMLEventWriter(OutputStream stream)</code></td>
<td>Create a new <code>XMLEventWriter</code> that writes to a stream.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventWriter(OutputStream stream, String encoding)</code></td>
<td>Create a new <code>XMLEventWriter</code> that writes to a stream.</td>
<td></td>
</tr>
<tr>
<td><code>createXMLEventWriter(Result result)</code></td>
<td>Create a new <code>XMLEventWriter</code> that writes to a JAXP result.</td>
<td></td>
</tr>
</tbody>
</table>

### Other Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Parameters</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getProperty(String name)</code></td>
<td>Get a feature/property on the underlying implementation.</td>
<td></td>
</tr>
<tr>
<td><code>isPropertySupported(String name)</code></td>
<td>Query the set of properties that this factory supports.</td>
<td></td>
</tr>
<tr>
<td><code>newInstance()</code></td>
<td>Create a new instance of the factory.</td>
<td></td>
</tr>
<tr>
<td><code>newInstance(String factoryId, ClassLoader classLoader)</code></td>
<td>Create a new instance of the factory.</td>
<td></td>
</tr>
</tbody>
</table>
abstract void `setProperty(String name, Object value)`

Allows the user to set specific features/properties on the underlying implementation.

Methods inherited from class `java.lang.Object`

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Field Detail

**IS_REPAIRING_NAMESPACES**

`public static final String IS_REPAIRING_NAMESPACES`

Property used to set prefix defaulting on the output side

See Also: [Constant Field Values](#)

Constructor Detail

**protected XMLOutputFactory()**

XMLOutputFactory

**protected XMLOutputFactory()**

Method Detail

`public static XMLOutputFactory newInstance() throws`
**FactoryConfigurationError**

Throws FactoryConfigurationError:

newInstance

public static XMLOutputFactory newInstance() throws FactoryConfigurationError

Create a new instance of the factory.

Throws:

FactoryConfigurationError - if an instance of this factory cannot be loaded

public static XMLInputFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationError

factoryId
classLoader
return

Throws FactoryConfigurationError:

newInstance

public static XMLInputFactory newInstance(String factoryId, ClassLoader classLoader) throws FactoryConfigurationError

Create a new instance of the factory

Parameters:

factoryId - Name of the factory to find, same as a property name
classLoader - classLoader to use

Returns:
the factory implementation

Throws:
FactoryConfigurationException - if an instance of this factory cannot be loaded

abstract public XMLStreamWriter createXMLStreamWriter(java.io.Writer stream) throws XMLStreamException
writer XMLStreamWriter
stream XMLStreamWriter writer

Create a new XMLStreamWriter that writes to a writer

Parameters:
stream - the writer to write to

Throws:
XMLStreamException

abstract public XMLStreamWriter createXMLStreamWriter(java.io.OutputStream stream) throws XMLStreamException

Create a new XMLStreamWriter that writes to a writer

Parameters:
stream - the writer to write to

Throws:
XMLStreamException
createXMLStreamWriter

public abstract XMLStreamWriter createXMLStreamWriter(OutputStream stream) throws XMLStreamException

Create a new XMLStreamWriter that writes to a stream

Parameters:
  stream - the stream to write to

Throws:
  XMLStreamException

abstract public XMLStreamWriter createXMLStreamWriter(java.io.OutputStream stream, String encoding) throws XMLStreamException

XMLStreamWriter
  stream
  encoding

Throws
  XMLStreamException:

createXMLStreamWriter

public abstract XMLStreamWriter createXMLStreamWriter(OutputStream stream, String encoding) throws XMLStreamException

Create a new XMLStreamWriter that writes to a stream

Parameters:
  stream - the stream to write to
  encoding - the encoding to use

Throws:
  XMLStreamException

abstract public XMLStreamWriter createXMLStreamWriter(javax.xml.transform.Result result)
createXMLStreamWriter

public abstract XMLStreamWriter createXMLStreamWriter(Result result)
throws XMLStreamException

Create a new XMLStreamWriter that writes to a JAXP result. This method is optional.

Parameters:
result - the result to write to

Throws:
UnsupportedOperationException - if this method is not supported by this XMLOutputFactory
XMLStreamException

abstract public XMLEventWriter createXMLEventWriter(java.xml.transform.Result result)
throws XMLStreamException

Create a new XMLEventWriter that writes to a JAXP result. This
method is optional.

**Parameters:**
- `result` - the result to write to

**Throws:**
- `UnsupportedOperationException` - if this method is not supported by this XMLOutputFactory
- `XMLStreamException`

```java
abstract public XMLEventWriter createXMLEventWriter(java.io.OutputStream stream)
throws XMLStreamException
```

Create a new XMLEventWriter that writes to a stream

**Parameters:**
- `stream` - the stream to write to

**Throws:**
- `XMLStreamException`

```java
public abstract XMLEventWriter createXMLEventWriter(java.io.OutputStream stream,
String encoding)
throws XMLStreamException
```

abstract public XMLEventWriter createXMLEventWriter(java.io.OutputStream stream,
String encoding) throws XMLStreamException

**Parameters:**
- `stream`
- `encoding`

**Throws:**
- `XMLStreamException`
createXMLEventWriter

public abstract XMLEventWriter createXMLEventWriter(OutputStream stream, String encoding) throws XMLStreamException

Create a new XMLEventWriter that writes to a stream

Parameters:
stream - the stream to write to
encoding - the encoding to use

Throws:
XMLStreamException

abstract public XMLEventWriter createXMLEventWriter(java.io.Writer stream) throws XMLStreamException

writer XMLEventWriter stream

throws XMLStreamException:

createXMLEventWriter

public abstract XMLEventWriter createXMLEventWriter(Writer stream) throws XMLStreamException

Create a new XMLEventWriter that writes to a writer

Parameters:
stream - the stream to write to

Throws:
XMLStreamException

abstract public void setProperty(String name, Object value)
Throws IllegalArgumentException
/

name
value
Throws IllegalArgumentExpection:

setProperty

public abstract void `setProperty`(String name,
Object value)
throws IllegalArgumentExpection

Allows the user to set specific features/properties on the underlying implementation.

Parameters:
name - The name of the property
value - The value of the property

Throws:
IllegalArgumentExpection - if the property is not supported

getPropert

abstract public Object `getProperty`(String name) throws
IllegalArgumentExpection
/

name
return
Throws IllegalArgumentExpection:

getPropert

public abstract Object `getProperty`(String name)
throws IllegalArgumentExpection

Get a feature/property on the underlying implementation
Parameters:

name - The name of the property

Returns:

The value of the property

Throws:

IllegalArgumentException - if the property is not supported

abstract public boolean isPropertySupported(String name)

name null

return true false

isPropertySupported

public abstract boolean isPropertySupported(String name)

Query the set of properties that this factory supports.

Parameters:

name - The name of the property (may not be null)

Returns:

ture if the property is supported and false otherwise
javax.xml.bind.annotation  Annotation Type XmlRegistry

@Retention(value=RUNTIME)  
@Target(value=TYPE)  
public @interface XmlRegistry  

Implementes: Annotation  
@Retention(value=RUNTIME)  
@Target(value=TYPE)  

XmlElementDecl  

since  
JAXB 2.0  
See also  
javax.xml.bind.annotation.XmlElementDecl  

Marks a class that has XmlElementDecl's.

Since:
JAXB 2.0

Author:
- Kohsuke Kawaguchi, Sun Microsystems, Inc.
- Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:
XmlElementDecl

Overview  Package  Tree  Deprecated  Index  Help

PREV CLASS  NEXT CLASS  
SUMMARY: REQUIRED | OPTIONAL  
FRAMES  NO FRAMES  
DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.stream  Interface XMLReporter

public interface XMLReporter

  version 1.0

This interface is used to report non-fatal errors. Only warnings should be echoed through this interface.

**Version:**

  1.0

**Author:**

  Copyright (c) 2003 by BEA Systems. All Rights Reserved.

---

### Method Summary

<table>
<thead>
<tr>
<th>method</th>
<th>signature</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>report</td>
<td>void report(String message, String errorType, Object relatedInformation, Location location)</td>
<td>Report the desired message in an application specific format.</td>
</tr>
</tbody>
</table>

### Method Detail

```java
public void report(String message, String errorType, Object relatedInformation, Location location) throws XMLStreamException
```

message

errorType

relatedInformation
void report(String message,
String errorType,
Object relatedInformation,
Location location)
throws XMLStreamException

Report the desired message in an application specific format. Only warnings and non-fatal errors should be reported through this interface. Fatal errors should be thrown as XMLStreamException.

Parameters:
message - the error message
errorType - an implementation defined error type
relatedInformation - information related to the error, if available
location - the location of the error, if available

Throws:
XMLStreamException

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
public interface XMLResolver

XML setXMLResolver XMLInputFactory

version 1.0

This interface is used to resolve resources during an XML parse. If an application wishes to perform custom entity resolution it must register an instance of this interface with the XMLInputFactory using the setXMLResolver method.

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>resolveEntity(String publicID, String systemID, Object String baseURI, String namespace)</td>
</tr>
<tr>
<td>Retrieves a resource.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method Detail</th>
</tr>
</thead>
<tbody>
<tr>
<td>public Object resolveEntity(String publicID, String systemID, String baseURI, String namespace) throws XMLStreamException</td>
</tr>
<tr>
<td>(1) java.io.InputStream (2) javax.xml.stream.XMLStreamReader (3)</td>
</tr>
</tbody>
</table>
resolveEntity

Object resolveEntity(String publicID, String systemID, String baseURI, String namespace) throws XMLStreamException

Retrieves a resource. This resource can be of the following three return types: (1) java.io.InputStream (2) javax.xml.stream.XMLStreamReader (3) java.xml.stream.XMLEventReader. If this method returns null the processor will attempt to resolve the entity using its default mechanism.

Parameters:
- publicID - The public identifier of the external entity being referenced, or null if none was supplied.
- systemID - The system identifier of the external entity being referenced.
- baseURI - Absolute base URI associated with systemId.
- namespace - The namespace of the entity to resolve.

Returns:
The resource requested or null.

Throws:
- XMLStreamException - if there was a failure attempting to resolve the resource.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.annotation  Annotation Type XmlRootElement

@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface XmlRootElement

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=TYPE)

XML

@XmlElement

javax.xml.bind.package javadoc ""

@XmlElement  XML  XML

XmlTypeXmlEnumXmlAccessorTypeXmlAccessorOrder

1  XML

// Example: Code fragment
@XmlElement
class Point {
    int x;
    int y;
    Point(int _x, int _y) {x=_x; y=_y;}
}

//Example: Code fragment corresponding to XML output
marshal( new Point(3,5), System.out);

&amp;lt;!-- Example: XML output --&amp;gt;
&lt;point&gt;
  &lt;x&gt;3 &lt;/x&gt;
  &lt;y&gt;5 &lt;/y&gt;
&lt;/point&gt;

XML

&lt;xs:element name="point" type="point"&gt;
  &lt;xs:complexType name="point"&gt;
    &lt;xs:sequence&gt;
    &lt;/xs:sequence&gt;
  &lt;/xs:complexType&gt;&lt;/xs:element&gt;
&lt;xs:element name="x" type="xs:int"&gt;
  &lt;xs:complexType name="point3D"&gt;
    &lt;xs:complexContent&gt;
      &lt;xs:extension base="point"&gt;
        &lt;xs:sequence&gt;
          &lt;xs:element name="z" type="xs:int"/&gt;
        &lt;/xs:sequence&gt;
      &lt;/xs:extension&gt;
    &lt;/xs:complexContent&gt;&lt;/xs:complexType&gt;&lt;/xs:element&gt;

2

// Example: Code fragment
@XmlElement
class Point3D extends Point {
    int z;
    Point3D(int _x, int _y, int _z) {super(_x,_y);z=_z;}
}

// Example: Code fragment corresponding to XML output *
marshal( new Point3D(3,5,0), System.out );

&lt;x&gt;3&lt;/x&gt;
&lt;y&gt;5&lt;/y&gt;
&lt;z&gt;0&lt;/z&gt;
&lt;/point3D&gt;

<!-- Example: XML schema definition --&gt;
&lt;xs:element name="point3D" type="point3D"&gt;
&lt;xs:complexType name="point3D"&gt;
  &lt;xs:complexContent&gt;
    &lt;xs:extension base="point"&gt;
      &lt;xs:sequence&gt;
        &lt;xs:element name="z" type="xs:int"/&gt;
      &lt;/xs:sequence&gt;
    &lt;/xs:extension&gt;
  &lt;/xs:complexContent&gt;
&lt;/xs:complexType&gt;&lt;/xs:element&gt;
Maps a class or an enum type to an XML element.

Usage

The @XmlRootElement annotation can be used with the following program elements:

- a top level class
- an enum type


When a top level class or an enum type is annotated with the @XmlRootElement annotation, then its value is represented as XML element in an XML document.

This annotation can be used with the following annotations: XmlType,
Example 1: Associate an element with XML Schema type

```java
@XmlRootElement
class Point {
    int x;
    int y;
    Point(int _x, int _y) {x=_x; y=_y;}
}
```

```xml
<point>
    <x>3</x>
    <y>5</y>
</point>
```

The annotation causes an global element declaration to be produced in the schema. The global element declaration is associated with the XML schema type to which the class is mapped.

Example 2: Orthogonality to type inheritance

An element declaration annotated on a type is not inherited by its derived types. The following example shows this.

```java
@XmlRootElement
class Point {
    int x;
    int y;
    Point(int _x, int _y) {x=_x; y=_y;}
}
```

```xml
<point>
    <x>3</x>
    <y>5</y>
</point>
```
class Point3D extends Point {
    int z;
    Point3D(int _x, int _y, int _z) {super(_x, _y); z=_z;}
}

//Example: Code fragment corresponding to XML output *
marshal( new Point3D(3,5,0), System.out );

<!-- Example: XML output -->
<!-- The element name is point3D not point -->
<point3D>
    <x>3</x>
    <y>5</y>
    <z>0</z>
</point3D>

<!-- Example: XML schema definition -->
<xs:element name="point3D" type="point3D"/>
<xs:complexType name="point3D">
    <xs:complexContent>
        <xs:extension base="point">
            <xs:sequence>
                <xs:element name="z" type="xs:int"/>
            </xs:sequence>
        </xs:extension>
    </xs:complexContent>
</xs:complexType>

Example 3: Associate a global element with XML Schema type to which
the class is mapped.

//Example: Code fragment
@XmlElement(name="PriceElement")
public class USPrice {
    @XmlElement
    public java.math.BigDecimal price;
}

<!-- Example: XML schema definition -->
<xs:element name="PriceElement" type="USPrice"/>
<xs:complexType name="USPrice">
    <xs:sequence>
        <xs:element name="price" type="xs:decimal"/>
    </xs:sequence>
</xs:complexType>

Since:
Optional Element Summary

<table>
<thead>
<tr>
<th>String name</th>
<th>local name of the XML element.</th>
</tr>
</thead>
<tbody>
<tr>
<td>String namespace</td>
<td>namespace name of the XML element.</td>
</tr>
</tbody>
</table>

abstract public String namespace()

XML

"##default" XML ( XmlSchema) XML

namespace

public abstract String namespace

namespace name of the XML element.

If the value is "##default", then the XML namespace name is derived from the package of the class (XmlSchema). If the package is unnamed, then the XML namespace is the default empty namespace.

Default:

"##default"

abstract public String name()

XML
"##default"

**name**

```java
public abstract String name
```

local name of the XML element.

If the value is "##default", then the name is derived from the class name.

**Default:**

"##default"

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).

PS:
javax.xml.bind.annotation Annotation Type XmlSchema

@Retention(value=RUNTIME)
@Target(value=PACKAGE)
public @interface XmlSchema

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=PACKAGE)

XML

XmlSchema


JSR 175 """

• JSR 175 """
• JSR 175 package-info.java JAXB package-info.java

1 XML

    @javax.xml.bind.annotation.XmlSchema (namespace = "http://www.example.com/MYPO1"
)

    <!-- XML Schema fragment -->
    <schema
        xmlns=...
        xmlns:po=....
        targetNamespace="http://www.example.com/MYPO1"
>
    <!-- prefixes generated by default are implementation dependent -->

2 URI
// Package level annotation
@javax.xml.bind.annotation.XmlSchema (  
    xmlns = {
        @javax.xml.bind.annotation.XmlNs(prefix = "po",  
            namespaceURI="http://www.example.com/myPO1"),

        @javax.xml.bind.annotation.XmlNs(prefix="xs",  
            namespaceURI="http://www.w3.org/2001/XMLSchema")
    }
)

<!-- XML Schema fragment -->
<schema
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    xmlns:po="http://www.example.com/PO1"
    targetNamespace="http://www.example.com/PO1">

3    elementFormDefault

@javax.xml.bind.annotation.XmlSchema (  
    elementFormDefault=XmlNsForm.UNQUALIFIED
    ...
)

<!-- XML Schema fragment -->
<schema
    xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:po="http://www.example.com/PO1"
    elementFormDefault="unqualified">

    since JAXB2.0

    version $Revision: 1.9 $  

Maps a package name to a XML namespace.

Usage

The XmlSchema annotation can be used with the following program elements:

- package
This is a package level annotation and follows the recommendations and restrictions contained in JSR 175, section III, "Annotations". Thus the usage is subject to the following constraints and recommendations.

- There can only be one package declaration as noted in JSR 175, section III, "Annotations".
- JSR 175 recommends package-info.java for package level annotations. JAXB Providers that follow this recommendation will allow the package level annotations to be defined in package-info.java.

**Example 1:** Customize name of XML namespace to which package is mapped.

```java
@javax.xml.bind.annotation.XmlSchema (  
   namespace = "http://www.example.com/MYP01"
)

<!-- XML Schema fragment -->
<schema  
   xmlns=...  
   xmlns:po=...  
   targetNamespace="http://www.example.com/MYP01"
  >
  <!-- prefixes generated by default are implementation depenedent -->
</schema>
```

**Example 2:** Customize namespace prefix, namespace URI mapping

```java
// Package level annotation  
@javax.xml.bind.annotation.XmlSchema (  
   xmlns = {  
      @javax.xml.bind.annotation.XmlNs(prefix = "po",  
         namespaceURI="http://www.example.com/myPO1"),

      @javax.xml.bind.annotation.XmlNs(prefix="xs",  
         namespaceURI="http://www.w3.org/2001/XMLSchema")  
   }
)

<!-- XML Schema fragment -->
<schema  
   xmlns:xs="http://www.w3.org/2001/XMLSchema"  
   xmlns:po="http://www.example.com/P01"  
   targetNamespace="http://www.example.com/P01">
```
Example 3: Customize elementFormDefault

```java
@javax.xml.bind.annotation.XmlSchema (elementFormDefault=XmlNsForm.UNQUALIFIED ... )

<schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:po="http://www.example.com/PO1"
    elementFormDefault="unqualified">
```

Since: JAXB2.0
Version: $Revision: 1.9 $
Author: Sekhar Vajjhala, Sun Microsystems, Inc.

---

Optional Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlNsForm</td>
<td><code>attributeFormDefault</code></td>
<td>Namespace qualification for attributes.</td>
</tr>
<tr>
<td>XmlNsForm</td>
<td><code>elementFormDefault</code></td>
<td>Namespace qualification for elements.</td>
</tr>
<tr>
<td>String</td>
<td><code>namespace</code></td>
<td>Name of the XML namespace.</td>
</tr>
<tr>
<td>XmlNs[]</td>
<td><code>xmlns</code></td>
<td>Customize the namespace URI, prefix associations.</td>
</tr>
</tbody>
</table>

abstract public `XmlNs[]` `xmlns()`

URI XML JAXB
xmlNs

public abstract XmlNs[] xmlns

Customize the namespace URI, prefix associations. By default, the namespace prefixes for a XML namespace are generated by a JAXB Provider in an implementation dependent way.

Default:

{}

abstract public String namespace()
XML

namespace

public abstract String namespace

Name of the XML namespace.

Default:

""

abstract public XmlNsForm elementFormDefault()
XML

elementFormDefault

public abstract XmlNsForm elementFormDefault

Namespace qualification for elements. By default, element default attribute will be absent from the XML Schema fragment.
abstract public XmlNsForm attributeFormDefault()
attributesFormDefault XML

attributeFormDefault

public abstract XmlNsForm attributeFormDefault

Namespace qualification for attributes. By default, attributesFormDefault will be absent from the XML Schema fragment.

Default:
UNSET
javax.xml.bind.annotation  Annotation Type XmlSchemaType

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PACKAGE})
public @interface XmlSchemaType

**Implements:** Annotation

**Inner classes:** XmlSchemaType.DEFAULT

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD, PACKAGE})

Java

```java
@XmlSchemaType
JavaBean
```

Java

```java
@XmlSchemaType
```

**XmlElement**

**XmlAttribute**

1 XMLGregorianCalendar

//Example: Code fragment
public class USPrice {
    @XmlElement
    @XmlSchemaType(name="date")
    public XMLGregorianCalendar date;
}

<!--[-- Example: Local XML Schema element -->
<x:s:complexType name="USPrice="/>
    <x:s:sequence>
        <x:s:element name="date" type="x:s:date"/>
    </sequence>
</x:s:complexType>
Maps a Java type to a simple schema built-in type.

**Usage**

@XmlSchemaType annotation can be used with the following program elements:

- a JavaBean property
- field
- package

@XmlSchemaType annotation defined for Java type applies to all references to the Java type from a property/field. A @XmlSchemaType annotation specified on the property/field overrides the @XmlSchemaType annotation specified at the package level.

This annotation can be used with the following annotations: `XmlElement`, `XmlAttribute`.

**Example 1:** Customize mapping of XMLGregorianCalendar on the field.

```java
//Example: Code fragment
public class USPrice {
    @XmlElement
    @XmlSchemaType(name="date")
    public XMLGregorianCalendar date;
}

<!-- Example: Local XML Schema element -->
<xs:complexType name="USPrice"/>
```
Example 2: Customize mapping of XMLGregorianCalendar at package level

```java
package foo;
@javax.xml.bind.annotation.XmlSchemaType(
    name="date", type=javax.xml.datatype.XMLGregorianCalendar.
)}
```

Since:
JAXB2.0

### Required Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
</table>

### Optional Element Summary

<table>
<thead>
<tr>
<th>String</th>
<th>namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>type</td>
</tr>
<tr>
<td></td>
<td>If this annotation is used at the package level, then value of the type() must be specified.</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public String name()
public abstract String name

abstract public String namespace()

namespace

public abstract String namespace

  Default:
  "http://www.w3.org/2001/XMLSchema"

abstract public Class<T> type()

type()

type

public abstract Class type

  If this annotation is used at the package level, then value of the type() must be specified.

  Default:
  javax.xml.bind.annotation.XmlSchemaType.DEFAULT.class
javax.xml.bind.annotation  Class XmlSchemaType.DEFAULT

java.lang.Object
   └─javax.xml.bind.annotation.XmlSchemaType.DEFAULT

Enclosing class:
   XmlSchemaType

public static final class XmlSchemaType.DEFAULT
   extends Object

Contained within: XmlSchemaType

   type()

Used in XmlSchemaType.type() to signal that the type be inferred from the signature of the property.

---

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlSchemaType.DEFAULT()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object: clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>

Constructor Detail
public XmlSchemaType.DEFAULT()

XmlSchemaType.DEFAULT

public XmlSchemaType.DEFAULT()
javax.xml.bind.annotation Annotation Type

XmlSchemaTypes

@Retention(value=RUNTIME)
@Target(value=PACKAGE)
public @interface XmlSchemaTypes

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value=PACKAGE)

@ XmlSchemaType

@XmlSchemaType

@XmlSchemaTypes({ @XmlSchemaType(...), @XmlSchemaType(...) })

@XmlSchemaTypes  XmlSchemaType

javax.xml.bind.package javadoc "" since JAXB2.0
See also  javax.xml.bind.annotation.XmlSchemaType

A container for multiple @XmlSchemaType annotations.

Multiple annotations of the same type are not allowed on a program element. This annotation therefore serves as a container annotation for multiple @XmlSchemaType annotations as follows:

@XmlSchemaTypes({ @XmlSchemaType(...), @XmlSchemaType(...) })

The @XmlSchemaTypes annotation can be used to define XmlSchemaType for different types at the package level.

Since:
   JAXB2.0

Author:
   • Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:
   XmlSchemaType

### Required Element Summary

<table>
<thead>
<tr>
<th>XmlSchemaType[]</th>
<th>value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Collection of @XmlSchemaType annotations</td>
</tr>
</tbody>
</table>

### Element Detail

abstract public `XmlSchemaType[] value()`

@`XmlSchemaType`

**value**

public abstract `XmlSchemaType[] value`

   Collection of @`XmlSchemaType` annotations
PS :
javax.xml.stream  Interface XMLStreamConstants

All Known Subinterfaces:
  Attribute, Characters, Comment, DTD, EndDocument, EndElement,
  EntityDeclaration, EntityReference, Namespace,
  NotationDeclaration, ProcessingInstruction, StartDocument,
  StartElement, XMLEvent, XMLStreamReader

All Known Implementing Classes:
  StreamReaderDelegate

public interface XMLStreamConstants

Implemented by:  XMLEvent, XMLStreamReader

API  0  256

This interface declares the constants used in this API. Numbers in the range 0 to 256 are reserved for the specification, user defined events must use event codes outside that range.

Field Summary

<table>
<thead>
<tr>
<th>static int</th>
<th>ATTRIBUTE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Indicates an event is an attribute</td>
</tr>
<tr>
<td>static int</td>
<td>CDATA</td>
</tr>
<tr>
<td></td>
<td>Indicates an event is a CDATA section</td>
</tr>
<tr>
<td>static int</td>
<td>CHARACTERS</td>
</tr>
<tr>
<td></td>
<td>Indicates an event is characters</td>
</tr>
<tr>
<td>static int</td>
<td>COMMENT</td>
</tr>
<tr>
<td></td>
<td>Indicates an event is a comment</td>
</tr>
<tr>
<td>static int</td>
<td>DTD</td>
</tr>
<tr>
<td></td>
<td>Indicates an event is a DTD</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>END_DOCUMENT</td>
<td>Indicates an event is an end document</td>
</tr>
<tr>
<td>END_ELEMENT</td>
<td>Indicates an event is an end element</td>
</tr>
<tr>
<td>ENTITY_DECLARATION</td>
<td>Indicates a Entity Declaration</td>
</tr>
<tr>
<td>ENTITY_REFERENCE</td>
<td>Indicates an event is an entity reference</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>Indicates the event is a namespace declaration</td>
</tr>
<tr>
<td>NOTATION_DECLARATION</td>
<td>Indicates a Notation</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION</td>
<td>Indicates an event is a processing instruction</td>
</tr>
<tr>
<td>SPACE</td>
<td>The characters are white space (see [XML], 2.10 &quot;White Space Handling&quot;).</td>
</tr>
<tr>
<td>START_DOCUMENT</td>
<td>Indicates an event is a start document</td>
</tr>
<tr>
<td>START_ELEMENT</td>
<td>Indicates an event is a start element</td>
</tr>
</tbody>
</table>

**Field Detail**

**START_ELEMENT**

static final int START_ELEMENT

Indicates an event is a start element

**See Also:**  
StartElement, Constant Field Values
**END_ELEMENT**

static final int **END_ELEMENT**

Indicates an event is an end element

**See Also:**  
[EndElement], [Constant Field Values]

---

**PROCESSING_INSTRUCTION**

static final int **PROCESSING_INSTRUCTION**

Indicates an event is a processing instruction

**See Also:**  
[ProcessingInstruction], [Constant Field Values]

---

**CHARACTERS**

static final int **CHARACTERS**

Indicates an event is characters

**See Also:**  
[Characters], [Constant Field Values]

---

**COMMENT**
static final int COMMENT

Indicates an event is a comment

See Also:
Comment, Constant Field Values

--------------------------------------------------

SPACE

static final int SPACE

The characters are white space (see [XML], 2.10 "White Space Handling"). Events are only reported as SPACE if they are ignorable white space. Otherwise they are reported as CHARACTERS.

See Also:
Characters, Constant Field Values

--------------------------------------------------

START_DOCUMENT

static final int START_DOCUMENT

Indicates an event is a start document

See Also:
StartDocument, Constant Field Values

--------------------------------------------------

END_DOCUMENT

static final int END_DOCUMENT
Indicates an event is an end document

**See Also:**
[EndDocument](#), [Constant Field Values](#)

---

**ENTITYREFERENCE**

`static final int ENTITYREFERENCE`

Indicates an event is an entity reference

**See Also:**
[EntityReference](#), [Constant Field Values](#)

---

**ATTRIBUTE**

`static final int ATTRIBUTE`

Indicates an event is an attribute

**See Also:**
[Attribute](#), [Constant Field Values](#)

---

**DTD**

`static final int DTD`

Indicates an event is a DTD

**See Also:**
[dtd](#), [Constant Field Values](#)
CDATA

static final int CDATA

Indicates an event is a CDATA section

See Also:
Characters, Constant Field Values

NAMESPACE

static final int NAMESPACE

Indicates the event is a namespace declaration

See Also:
Namespace, Constant Field Values

NOTATION_DECLARATION

static final int NOTATION_DECLARATION

Indicates a Notation

See Also:
NotationDeclaration, Constant Field Values

ENTITY_DECLARATION
static final int ENTITY_DECLARATION

Indicates a Entity Declaration

See Also:
NotationDeclaration, Constant Field Values

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.stream Class XMLStreamException

java.lang.Object
  └ java.lang.Throwable
      └ java.lang.Exception
          └ javax.xml.stream.XMLStreamException

All Implemented Interfaces:
  Serializable

public class XMLStreamException
extends Exception

Extends: Throwable > Exception

version 1.0

The base exception for unexpected processing errors. This Exception class is used to report well-formedness errors as well as unexpected processing conditions.

Version: 1.0
Author: Copyright (c) 2003 by BEA Systems. All Rights Reserved.
See Also: Serialized Form

Field Summary

| protected Location location |
| protected nested          |
## Constructor Summary

<table>
<thead>
<tr>
<th>Constructor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>XMLStreamException()</code></td>
<td>Default constructor</td>
</tr>
<tr>
<td><code>XMLStreamException(String msg)</code></td>
<td>Construct an exception with the associated message.</td>
</tr>
<tr>
<td><code>XMLStreamException(String msg, Location location)</code></td>
<td>Construct an exception with the associated message, exception and location.</td>
</tr>
<tr>
<td><code>XMLStreamException(String msg, Location location, Throwable th)</code></td>
<td>Construct an exception with the associated message, exception and location.</td>
</tr>
<tr>
<td><code>XMLStreamException(String msg, Throwable th)</code></td>
<td>Construct an exception with the associated message and exception</td>
</tr>
<tr>
<td><code>XMLStreamException(Throwable th)</code></td>
<td>Construct an exception with the associated exception</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>getLocation()</code></td>
<td>Gets the location of the exception</td>
</tr>
<tr>
<td><code>getNestedException()</code></td>
<td>Gets the nested exception.</td>
</tr>
</tbody>
</table>

## Methods inherited from class java.lang.Throwable

- `fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`, `printStackTrace`, `setStackTrace`, `toString`

## Methods inherited from class java.lang.Object

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`
Field Detail

nested

protected Throwable nested

location

protected Location location

Constructor Detail

public XMLStreamException()

XMLStreamException

public XMLStreamException()

    Default constructor

public XMLStreamException(String msg)

    msg

XMLStreamException
public XMLStreamException(String msg)

Construct an exception with the associated message.

**Parameters:**

msg - the message to report

---

public XMLStreamException(Throwable th)

th

---

XMLStreamException

public XMLStreamException(Throwable th)

Construct an exception with the associated exception

**Parameters:**

th - a nested exception

---

public XMLStreamException(String msg, Throwable th)

th
msg

---

XMLStreamException

public XMLStreamException(String msg, Throwable th)

Construct an exception with the associated message and exception

**Parameters:**

th - a nested exception
msg - the message to report
public XMLStreamException(String msg, Location location, Throwable th)

    th
    msg
    location

XMLStreamException

public XMLStreamException(String msg, Location location, Throwable th)

    Construct an exception with the associated message, exception and location.

    Parameters:
    th - a nested exception
    msg - the message to report
    location - the location of the error

public XMLStreamException(String msg, Location location)

    msg
    location

XMLStreamException

public XMLStreamException(String msg, Location location)

    Construct an exception with the associated message, exception and location.
Parameters:
  msg - the message to report
  location - the location of the error

Method Detail

public Throwable getNestedException()

  return

getNestedException

public Throwable getNestedException()

  Gets the nested exception.

  Returns:
  Nested exception

public Location getLocation()

  return null

getLocation

public Location getLocation()

  Gets the location of the exception

  Returns:
  the location of the exception, may be null if none is available
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAME</td>
<td>NO FRAMES</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

SUMMARY: NESTED | FIELD | CONSTR | METHOD |

DETAIL: FIELD | CONSTR | METHOD |
javax.xml.stream  Interface XMLStreamReader

All Superinterfaces:
   XMLStreamConstants

All Known Implementing Classes:
   StreamReaderDelegate

public interface XMLStreamReader
   extends XMLStreamConstants

Implements: XMLStreamConstants
Implemented by: StreamReaderDelegate

XMLStreamReader  XML  XML  XML

XMLStreamReader next() hasNext() XML
getEventType()getNamespaceURI()getLocalName() getText()

next()  reader next()

   getEventType()

XML DTD

XML 1.0XML  API javax.xml.stream.notations
javax.xml.stream.entities DTD  Notation
getProperty("javax.xml.stream.notations");  List l = (List)
getProperty("javax.xml.stream.entities");  DTD  null

java.lang.IllegalStateException

<table>
<thead>
<tr>
<th>getProperty()</th>
<th>hasNext()</th>
</tr>
</thead>
<tbody>
<tr>
<td>require()</td>
<td>close()</td>
</tr>
<tr>
<td>getNamespaceURI()</td>
<td></td>
</tr>
<tr>
<td>Command</td>
<td>Methods</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>isStartElement()</td>
<td>isEndElement(isCharacters()) isWhiteSpace() getNamespaceContext() getEventTypeName() getLocation() hasText() hasName()</td>
</tr>
<tr>
<td>next()</td>
<td>getNamespaceContext() getEventType() getLocation()</td>
</tr>
<tr>
<td>getName()</td>
<td>getPrefix() getAttributeXXX() isAttributeSpecified() getNamespaceXXX() getElementText() nextTag()</td>
</tr>
<tr>
<td>getLocalName()</td>
<td></td>
</tr>
<tr>
<td>hasName()</td>
<td></td>
</tr>
<tr>
<td>getPrefix()</td>
<td></td>
</tr>
<tr>
<td>getNext()</td>
<td></td>
</tr>
<tr>
<td>getNamespaceXXX()</td>
<td></td>
</tr>
<tr>
<td>getElementText()</td>
<td></td>
</tr>
<tr>
<td>nextTag()</td>
<td></td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td>getAttributeXXX()</td>
</tr>
<tr>
<td>getLocalName()</td>
<td></td>
</tr>
<tr>
<td>isAttributeSpecified()</td>
<td></td>
</tr>
<tr>
<td>END_ELEMENT</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td>getName() getLocalName() hasName() getPrefix() getNamespaceXXX()</td>
</tr>
<tr>
<td>getLocalName()</td>
<td></td>
</tr>
<tr>
<td>hasName()</td>
<td></td>
</tr>
<tr>
<td>getPrefix()</td>
<td></td>
</tr>
<tr>
<td>getNamespaceXXX()</td>
<td></td>
</tr>
<tr>
<td>getElementText()</td>
<td></td>
</tr>
<tr>
<td>nextTag()</td>
<td></td>
</tr>
<tr>
<td>CHARACTERS</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td>getEncoding() getVersion() isStandalone() standaloneSet()</td>
</tr>
<tr>
<td>getEncoding()</td>
<td></td>
</tr>
<tr>
<td>getVersion()</td>
<td></td>
</tr>
<tr>
<td>isStandalone()</td>
<td></td>
</tr>
<tr>
<td>standaloneSet()</td>
<td></td>
</tr>
<tr>
<td>getCharacterEncodingScheme()</td>
<td></td>
</tr>
<tr>
<td>nextTag()</td>
<td></td>
</tr>
<tr>
<td>START_DOCUMENT</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td></td>
</tr>
<tr>
<td>END_DOCUMENT</td>
<td>close()</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td>getPITarget()</td>
</tr>
<tr>
<td>getPIData()</td>
<td></td>
</tr>
<tr>
<td>nextTag()</td>
<td></td>
</tr>
<tr>
<td>ENTITYREFERENCE</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td>getLocalName()</td>
</tr>
<tr>
<td>getText()</td>
<td></td>
</tr>
<tr>
<td>nextTag()</td>
<td></td>
</tr>
<tr>
<td>DTD</td>
<td></td>
</tr>
<tr>
<td>next()</td>
<td></td>
</tr>
<tr>
<td>version</td>
<td>1.0</td>
</tr>
<tr>
<td>See</td>
<td>javax.xml.stream.events.XMLEvent, javax.xml.stream.XMLInputFactory,</td>
</tr>
</tbody>
</table>
The XMLStreamReader interface allows forward, read-only access to XML. It is designed to be the lowest level and most efficient way to read XML data.

The XMLStreamReader is designed to iterate over XML using next() and hasNext(). The data can be accessed using methods such as getEventType(), getNamespaceURI(), getLocalName() and getText();

The `next()` method causes the reader to read the next parse event. The `next()` method returns an integer which identifies the type of event just read.

The event type can be determined using `getEventType()`.

Parsing events are defined as the XML Declaration, a DTD, start tag, character data, white space, end tag, comment, or processing instruction. An attribute or namespace event may be encountered at the root level of a document as the result of a query operation.

For XML 1.0 compliance an XML processor must pass the identifiers of declared unparsed entities, notation declarations and their associated identifiers to the application. This information is provided through the property API on this interface. The following two properties allow access to this information: `javax.xml.stream.notations` and `javax.xml.stream.entities`. When the current event is a DTD the following call will return a list of Notations:

```java
List l = (List) getProperty("javax.xml.stream.notations");
```

The following call will return a list of entity declarations:

```java
List l = (List) getProperty("javax.xml.stream.entities");
```

These properties can only be accessed during a DTD event and are defined to return null if the information is not available.

The following table describes which methods are valid in what state. If a method is called in an invalid state the method will throw a java.lang.IllegalStateException.
<table>
<thead>
<tr>
<th>Event Type</th>
<th>Valid Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>All States</td>
<td>getProperty(), hasNext(), require(), close(),</td>
</tr>
<tr>
<td></td>
<td>getNamespaceURI(), isStartElement(),</td>
</tr>
<tr>
<td></td>
<td>isEndElement(), isCharacters(), isWhiteSpace(),</td>
</tr>
<tr>
<td></td>
<td>getNamespaceContext(), getEventType(), getLocation(), hasText(), hasName()</td>
</tr>
<tr>
<td>START_ELEMENT</td>
<td>next(), getName(), getLocalName(), hasName(), getPrefix(), attributeXXX(),</td>
</tr>
<tr>
<td></td>
<td>isAttributeSpecified(), getNamespaceXXX(), getElementText(), nextTag()</td>
</tr>
<tr>
<td>ATTRIBUTE</td>
<td>next(), nextTag()</td>
</tr>
<tr>
<td></td>
<td>getAttributeXXX(), isAttributeSpecified(),</td>
</tr>
<tr>
<td>END_ELEMENT</td>
<td>next(), getName(), getLocalName(), hasName(), getPrefix(), getNamespaceXXX(),</td>
</tr>
<tr>
<td></td>
<td>nextTag()</td>
</tr>
<tr>
<td>CHARACTERS</td>
<td>next(), getTextXXX(), nextTag()</td>
</tr>
<tr>
<td>CDATA</td>
<td>next(), getTextXXX(), nextTag()</td>
</tr>
<tr>
<td>COMMENT</td>
<td>next(), getTextXXX(), nextTag()</td>
</tr>
<tr>
<td>SPACE</td>
<td>next(), getTextXXX(), nextTag()</td>
</tr>
<tr>
<td>START_DOCUMENT</td>
<td>next(), getEncoding(), getVersion(), isStandalone(), standaloneSet(),</td>
</tr>
<tr>
<td></td>
<td>getCharacterEncodingScheme(), nextTag()</td>
</tr>
<tr>
<td>END_DOCUMENT</td>
<td>close()</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION</td>
<td>next(), getPITarget(), getPIData(), nextTag()</td>
</tr>
<tr>
<td>ENTITY_REFERENCE</td>
<td>next(), getLocalName(), getText(), nextTag()</td>
</tr>
</tbody>
</table>
## Field Summary

Fields inherited from interface javax.xml.stream.XMLStreamConstants

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE</td>
<td>CDATA, CHARACTERS, COMMENT, DTD, END_DOCUMENT, END_ELEMENT, ENTITY_DECLARATION, ENTITY_REFERENCE, NAMESPACE, NOTATION_DECLARATION, PROCESSING_INSTRUCTION, SPACE, START_DOCUMENT, START_ELEMENT</td>
</tr>
</tbody>
</table>

## Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void close()</td>
<td>Frees any resources associated with this Reader.</td>
</tr>
<tr>
<td>int getAttributeCount()</td>
<td>Returns the count of attributes on this START_ELEMENT, this method is only valid on a START_ELEMENT or ATTRIBUTE.</td>
</tr>
<tr>
<td>String getAttributeLocalName(int index)</td>
<td>Returns the localName of the attribute at the provided index</td>
</tr>
<tr>
<td>QName getAttributeName(int index)</td>
<td>Returns the qname of the attribute at the provided index</td>
</tr>
<tr>
<td>String getAttributeNamespace(int index)</td>
<td>Returns the namespace of the attribute at the provided index</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>getAttributePrefix(int index)</code></td>
<td>Returns the prefix of this attribute at the provided index</td>
</tr>
<tr>
<td><code>getAttributeType(int index)</code></td>
<td>Returns the XML type of the attribute at the provided index</td>
</tr>
<tr>
<td><code>getAttributeValue(int index)</code></td>
<td>Returns the value of the attribute at the index</td>
</tr>
<tr>
<td><code>getAttributeValue(String namespaceURI, String localName)</code></td>
<td>Returns the normalized attribute value of the attribute with the namespace and localName. If the namespaceURI is null the namespace is not checked for equality.</td>
</tr>
<tr>
<td><code>getCharacterEncodingScheme()</code></td>
<td>Returns the character encoding declared on the xml declaration Returns null if none was declared</td>
</tr>
<tr>
<td><code>getElementText()</code></td>
<td>Reads the content of a text-only element, an exception is thrown if this is not a text-only element.</td>
</tr>
<tr>
<td><code>getEncoding()</code></td>
<td>Return input encoding if known or null if unknown.</td>
</tr>
<tr>
<td><code>getEventType()</code></td>
<td>Returns an integer code that indicates the type of the event the cursor is pointing to.</td>
</tr>
<tr>
<td><code>getLocalName()</code></td>
<td>Returns the (local) name of the current event.</td>
</tr>
<tr>
<td><code>getLocation()</code></td>
<td>Return the current location of the processor.</td>
</tr>
<tr>
<td><code>getName()</code></td>
<td>Returns a QName for the current START_ELEMENT or END_ELEMENT event</td>
</tr>
<tr>
<td><code>getNamespaceContext()</code></td>
<td>Returns a read only namespace context for the current position.</td>
</tr>
<tr>
<td><code>getNamespaceCount()</code></td>
<td></td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>int getNamespaceCount()</td>
<td>Returns the count of namespaces declared on this START_ELEMENT or END_ELEMENT, this method is only valid on a START_ELEMENT, END_ELEMENT or NAMESPACE.</td>
</tr>
<tr>
<td>String getNamespacePrefix(int index)</td>
<td>Returns the prefix for the namespace declared at the index.</td>
</tr>
<tr>
<td>String getNamespaceURI()</td>
<td>If the current event is a START_ELEMENT or END_ELEMENT this method returns the URI of the prefix or the default namespace.</td>
</tr>
<tr>
<td>String getNamespaceURI(int index)</td>
<td>Returns the uri for the namespace declared at the index.</td>
</tr>
<tr>
<td>String getNamespaceURI(String prefix)</td>
<td>Return the uri for the given prefix.</td>
</tr>
<tr>
<td>String getPIData()</td>
<td>Get the data section of a processing instruction</td>
</tr>
<tr>
<td>String getPITarget()</td>
<td>Get the target of a processing instruction</td>
</tr>
<tr>
<td>String getPrefix()</td>
<td>Returns the prefix of the current event or null if the event does not have a prefix.</td>
</tr>
<tr>
<td>Object getProperty(String name)</td>
<td>Get the value of a feature/property from the underlying implementation.</td>
</tr>
<tr>
<td>String getText()</td>
<td>Returns the current value of the parse event as a string, this returns the string value of a CHARACTERS event, returns the value of a COMMENT, the replacement value for an ENTITY_REFERENCE, the string value of a CDATA section, the string value for a SPACE event, or the String value of the internal subset of the DTD.</td>
</tr>
<tr>
<td>char[] getTextCharacters()</td>
<td>Returns an array which contains the characters from this event.</td>
</tr>
</tbody>
</table>
**getTextCharacters(int sourceStart, char[] target, int targetStart, int length)**

Gets the text associated with a CHARACTERS, SPACE or CDATA event.

**getTextLength()**

Returns the length of the sequence of characters for this Text event within the text character array.

**getTextStart()**

Returns the offset into the text character array where the first character (of this text event) is stored.

**getVersion()**

Get the xml version declared on the xml declaration. Returns null if none was declared.

**hasName()**

Returns true if the current event has a name (is a START_ELEMENT or END_ELEMENT) returns false otherwise

**hasNext()**

Returns true if there are more parsing events and false if there are no more events.

**hasText()**

Return true if the current event has text, false otherwise. The following events have text: CHARACTERS, DTD, ENTITY_REFERENCE, COMMENT, SPACE

**isAttributeSpecified(int index)**

Returns a boolean which indicates if this attribute was created by default.

**isCharacters()**

Returns true if the cursor points to a character data event.

**isEndElement()**

Returns true if the cursor points to an end tag (otherwise false)

**isStandalone()**
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>boolean isStartElement()</td>
<td>Returns true if the cursor points to a start tag (otherwise false)</td>
</tr>
<tr>
<td>boolean isWhiteSpace()</td>
<td>Returns true if the cursor points to a character data event that consists of all whitespace</td>
</tr>
<tr>
<td>int next()</td>
<td>Get next parsing event - a processor may return all contiguous character data in a single chunk, or it may split it into several chunks.</td>
</tr>
<tr>
<td>int nextTag()</td>
<td>Skips any white space (isWhiteSpace() returns true), COMMENT, or PROCESSING_INSTRUCTION, until a START_ELEMENT or END_ELEMENT is reached.</td>
</tr>
<tr>
<td>void require(int type, String namespaceURI, String localName)</td>
<td>Test if the current event is of the given type and if the namespace and name match the current namespace and name of the current event.</td>
</tr>
<tr>
<td>boolean standaloneSet()</td>
<td>Checks if standalone was set in the document</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public Object getProperty(String name) throws IllegalArgumentException
/
```

**name**
null

**return**

**Throws**
IllegalArgumentException: name null
getProperty

`Object getProperty(String name)`

throws `IllegalArgumentException`

Get the value of a feature/property from the underlying implementation.

**Parameters:**
- `name` - The name of the property, may not be null

**Returns:**
- The value of the property

**Throws:**
- `IllegalArgumentException` - if name is null

---

```java
public int next() throws XMLStreamException
    —— javax.xml.stream.isCoalescing true
    CDATA CHARACTERS
    "" CHARACTERS

    XML
    <foo>!--description--><!
    [CDATA[<greeting>Hello</greeting>]]></foo>
    foo next()
    1- (COMMENT)
    2- (CHARACTERS)
    3- CDATA CHARACTERS
    4- CHARACTERS
    5- END_ELEMENT

    <tag/>START_ELEMENT END_ELEMENT
    —— <tag></tag> hasNext() false
IllegalStateException
    return
    Throws NoSuchElementException: hasNext() false
```
next

`int next()`

Throws `XMLStreamException`: XML

Get next parsing event - a processor may return all contiguous character data in a single chunk, or it may split it into several chunks. If the property `javax.xml.stream.isCoalescing` is set to true element content must be coalesced and only one `CHARACTERS` event must be returned for contiguous element content or CDATA Sections. By default entity references must be expanded and reported transparently to the application. An exception will be thrown if an entity reference cannot be expanded. If element content is empty (i.e. content is "") then no `CHARACTERS` event will be reported.

Given the following XML:
```
<foo><!--description-->content text<![CDATA[<greeting>Hello</greeting>]]>other content</foo>
```
The behavior of calling `next()` when being on `foo` will be:
1. the comment (COMMENT)
2. then the characters section (CHARACTERS)
3. then the CDATA section (another CHARACTERS)
4. then the next characters section (another CHARACTERS)
5. then the END_ELEMENT

**NOTE:** empty element (such as `<tag/>`) will be reported with two separate events: `START_ELEMENT`, `END_ELEMENT` - This preserves parsing equivalency of empty element to `<tag></tag>`. This method will throw an `IllegalStateException` if it is called after `hasNext()` returns false.

**Returns:**
the integer code corresponding to the current parse event

**Throws:**
NoSuchElementException - if this is called when hasNext() returns false

XMLStreamException - if there is an error processing the underlying XML source

See Also:

XMLEvent

---

```java
public void require(int type, String namespaceURI, String localName) throws XMLStreamException
namespaceURI null localName

    type
namespaceURI URI null
    localName localName null

    Throws XMLStreamException:
```

### require

```java
void require(int type,
    String namespaceURI,
    String localName)
throws XMLStreamException
```

Test if the current event is of the given type and if the namespace and name match the current namespace and name of the current event. If the namespaceURI is null it is not checked for equality, if the localName is null it is not checked for equality.

**Parameters:**

- `type` - the event type
- `namespaceURI` - the uri of the event, may be null
- `localName` - the localName of the event, may be null

**Throws:**

- `XMLStreamException` - if the required values are not matched.

---

```java
public String getElementText() throws
```

---
if(getEventType() != XMLStreamConstants.START_ELEMENT) {
    throw new XMLStreamException("parser must be on START_ELEMENT to read next text", getLocation());
}
int eventType = next();
StringBuffer content = new StringBuffer();
while(eventType != XMLStreamConstants.END_ELEMENT) {
    if(eventType == XMLStreamConstants.CHARACTERS ||
        eventType == XMLStreamConstants.CDATA ||
        eventType == XMLStreamConstants.SPACE ||
        eventType == XMLStreamConstants.ENTITY_REFERENCE) {
        buf.append(getText());
    } else if(eventType == XMLStreamConstants.PROCESSING_INSTRUCTION ||
        eventType == XMLStreamConstants.COMMENT) {
        // skipping
    } else if(eventType == XMLStreamConstants.END_DOCUMENT) {
        throw new XMLStreamException("unexpected end of document when reading element text content", getLocation());
    } else if(eventType == XMLStreamConstants.START_ELEMENT) {
        throw new XMLStreamException("element text content may not contain START_ELEMENT", getLocation());
    } else {
        throw new XMLStreamException("Unexpected event type "+eventType, getLocation());
    }
    eventType = next();
} return buf.toString();

Throws XMLStreamException: START_ELEMENT

g getElementText
String getElementText() throws XMLStreamException

Reads the content of a text-only element, an exception is thrown if this is not a text-only element. Regardless of value of javax.xml.stream.isCoalescing this method always returns coalesced content.
Precondition: the current event is START_ELEMENT.
Postcondition: the current event is the corresponding END_ELEMENT.
The method does the following (implementations are free to optimized but must do equivalent processing):

    if(getEventType() != XMLStreamConstants.START_ELEMENT) {
        throw new XMLStreamException("parser must be on START_ELEMENT to read next text", getLocation());
        int eventType = next();
        StringBuffer content = new StringBuffer();
        while(eventType != XMLStreamConstants.END_ELEMENT) {
            if(eventType == XMLStreamConstants.CHARACTERS || eventType == XMLStreamConstants.CDATA || eventType == XMLStreamConstants.SPACE || eventType == XMLStreamConstants.ENTITY_REFERENCE) {
                buf.append(getText());
            } else if(eventType == XMLStreamConstants.PROCESSING_INSTRUCTION || eventType == XMLStreamConstants.COMMENT) {
                // skipping
            } else if(eventType == XMLStreamConstants.END_DOCUMENT) {
                throw new XMLStreamException("unexpected end of document when reading element text content", getLocation());
            } else if(eventType == XMLStreamConstants.START_ELEMENT) {
                throw new XMLStreamException("element text content may not contain START_ELEMENT", getLocation());
            } else {
                throw new XMLStreamException("Unexpected event type "+eventType, getLocation());
            }
            eventType = next();
        }
        return buf.toString();
    }

Throws:
    XMLStreamException - if the current event is not a START_ELEMENT or if a non text element is encountered
public int nextTag() throws XMLStreamException

START_ELEMENT END_ELEMENT

isWhiteSpace() true

COMMENT

PROCESSING_INSTRUCTION

COMMENT

PROCESSING_INSTRUCTION

START_ELEMENT

END_ELEMENT

START_ELEMENT END_ELEMENT

int eventType = next();
while((eventType == XMLStreamConstants.CHARACTERS && isWhiteSpace())
|| (eventType == XMLStreamConstants.CDATA && isWhiteSpace())
// skip whitespace
|| eventType == XMLStreamConstants.SPACE
|| eventType == XMLStreamConstants.PROCESSING_INSTRUCTION
|| eventType == XMLStreamConstants.COMMENT
)
{
    eventType = next();
}
if (eventType != XMLStreamConstants.START_ELEMENT && eventType != XMLStreamConstants.END_ELEMENT)
throw new String XMLStreamException("expected start or end tag", getLocation());
return eventType;

return START_ELEMENT END_ELEMENT

Throws XMLStreamException: PROCESSING_INSTRUCTION

START_ELEMENT END_ELEMENT

Throws NoSuchElementException: hasNext() false

nextTag

int nextTag()
    throws XMLStreamException

Skips any white space (isWhiteSpace() returns true), COMMENT, or
PROCESSING_INSTRUCTION, until a START_ELEMENT or END_ELEMENT is reached. If other than white space characters, COMMENT, PROCESSING_INSTRUCTION, START_ELEMENT, END_ELEMENT are encountered, an exception is thrown. This method should be used when processing element-only content separated by white space.

Precondition: none
Postcondition: the current event is START_ELEMENT or END_ELEMENT and cursor may have moved over any whitespace event.

Essentially it does the following (implementations are free to optimized but must do equivalent processing):

```java
int eventType = next();
while((eventType == XMLStreamConstants.CHARACTERS && isWhiteSpace())
     || (eventType == XMLStreamConstants.CDATA && isWhiteSpace())
     // skip whitespace
     || eventType == XMLStreamConstants.SPACE
     || eventType == XMLStreamConstants.PROCESSING_INSTRUCTION
     || eventType == XMLStreamConstants.COMMENT
     ) {
    eventType = next();
}
if (eventType != XMLStreamConstants.START_ELEMENT && eventType != XMLStreamConstants.END_ELEMENT)
    throw new StringXMLStreamException("expected start or end tag"
}
return eventType;
```

**Returns:**
the event type of the element read (START_ELEMENT or END_ELEMENT)

**Throws:**

- XMLStreamException - if the current event is not white space, PROCESSING_INSTRUCTION, START_ELEMENT or END_ELEMENT
- NoSuchElementException - if this is called when hasNext() returns false

---

public boolean hasNext() throws XMLStreamException
true false XMLStreamReader
hasNext

boolean hasNext() throws XMLStreamException

Returns true if there are more parsing events and false if there are no more events. This method will return false if the current state of the XMLStreamReader is END_DOCUMENT

Returns:
true if there are more events, false otherwise

Throws:
XMLStreamException - if there is a fatal error detecting the next state

public void close() throws XMLStreamException

Frees any resources associated with this Reader. This method does not close the underlying input source.

Throws:
XMLStreamException - if there are errors freeing associated resources
public String getNamespaceURI(String prefix)
URI URI

**Namespaces in XML** 'xml'
"http://www.w3.org/XML/1998/namespace"

'xmlns'  
http://www.w3.org/2000/xmlns/
  
prefix  null
return  URI null
Throws  IllegalArgumentException: null

getNamespaceURI

String getNamespaceURI(String prefix)

Return the uri for the given prefix. The uri returned depends on the current state of the processor.

**NOTE:** The 'xml' prefix is bound as defined in **Namespaces in XML** specification to "http://www.w3.org/XML/1998/namespace".

**NOTE:** The 'xmlns' prefix must be resolved to following namespace http://www.w3.org/2000/xmlns/

Parameters:
- prefix - The prefix to lookup, may not be null

Returns:
- the uri bound to the given prefix or null if it is not bound

Throws:
- IllegalArgumentException - if the prefix is null

public boolean isStartElement()
true false
  
return  true false
isStartElement

boolean isStartElement()

Returns true if the cursor points to a start tag (otherwise false)

Returns:
true if the cursor points to a start tag, false otherwise

public boolean isEndElement()
true false
return true false

isEndElement

boolean isEndElement()

Returns true if the cursor points to an end tag (otherwise false)

Returns:
true if the cursor points to an end tag, false otherwise

public boolean isCharacters()
true
return true false

isCharacters

boolean isCharacters()

Returns true if the cursor points to a character data event
public boolean isWhiteSpace()
true
return true false

isWhiteSpace

boolean isWhiteSpace()

Returns true if the cursor points to a character data event that consists of all whitespace

Returns:
true if the cursor points to all whitespace, false otherwise

public String getAttributeValue(String namespaceURI,
String localName)
localName namespaceURI null
namespaceURI

namespaceURI

localName
null

return null

Throws
IllegalStateException: START_ELEMENT ATTRIBUTE

getAttributeValue

String getAttributeValue(String namespaceURI,
String localName)

Returns the normalized attribute value of the attribute with the namespace and localName If the namespaceURI is null the namespace is not checked for equality
Parameters:
namespaceURI - the namespace of the attribute
localName - the local name of the attribute, cannot be null

Returns:
returns the value of the attribute, returns null if not found

Throws:
IllegalStateException - if this is not a START_ELEMENT or ATTRIBUTE

public int getAttributeCount()
START_ELEMENT START_ELEMENT
ATTRIBUTE
return
Throws IllegalStateException: START_ELEMENT ATTRIBUTE

getAttributeCount

int getAttributeCount()

Returns the count of attributes on this START_ELEMENT, this method is only valid on a START_ELEMENT or ATTRIBUTE. This count excludes namespace definitions. Attribute indices are zero-based.

Returns:
returns the number of attributes

Throws:
IllegalStateException - if this is not a START_ELEMENT or ATTRIBUTE

public javax.xml.namespace.QName getAttributeName(int index)
qname
index


getAttributeName

```java
QName getAttributeName(int index)
```

Returns the qname of the attribute at the provided index

**Parameters:**
- index - the position of the attribute

**Returns:**
- the QName of the attribute

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or ATTRIBUTE

---

getAttributeNamespace

```java
public String getAttributeNamespace(int index)
```

```java
index
return URI null
Throws IllegalStateException: START_ELEMENT ATTRIBUTE
```

getattributeNamespace

```java
String getAttributeNamespace(int index)
```

Returns the namespace of the attribute at the provided index

**Parameters:**
- index - the position of the attribute

**Returns:**
- the namespace URI (can be null)

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or
public String getAttributeLocalName(int index) {
    String localName = attributeName;
    return localName;
}

getAttributeLocalName

String getAttributeLocalName(int index)

    Returns the localName of the attribute at the provided index

Parameters:
    index - the position of the attribute

Returns:
    the localName of the attribute

Throws:
    IllegalStateException - if this is not a START_ELEMENT or ATTRIBUTE

public String getAttributePrefix(int index) {
    String prefix = attributePrefix;
    return prefix;
}

getAttributePrefix

String getAttributePrefix(int index)

    Returns the prefix of this attribute at the provided index

    IllegalStateException - if this is not a START_ELEMENT or ATTRIBUTE
**getAttributeType**

*public String getAttributeType(int index)*

**Parameters:**
- `index` - the position of the attribute

**Returns:**
the prefix of the attribute

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or ATTRIBUTE

---

**getValue**

*public String getAttributeValue(int index)*

**Parameters:**
- `index` - the position of the attribute

**Returns:**
the XML type of the attribute

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or ATTRIBUTE
**getAttributeValue**

`String getAttributeValue(int index)`

Returns the value of the attribute at the index

**Parameters:**
- index - the position of the attribute

**Returns:**
- the attribute value

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or ATTRIBUTE

---

**public boolean isAttributeSpecified(int index)**

`boolean isAttributeSpecified(int index)`

`index`

`return true`

**Throws** `IllegalStateException`: START_ELEMENT ATTRIBUTE

---

**isAttributeSpecified**

`boolean isAttributeSpecified(int index)`

Returns a boolean which indicates if this attribute was created by default

**Parameters:**
- index - the position of the attribute

**Returns:**
- true if this is a default attribute

**Throws:**
- `IllegalStateException` - if this is not a START_ELEMENT or ATTRIBUTE
public int getNamespaceCount()

    START_ELEMENT END_ELEMENT
    START_ELEMENTEND_ELEMENT NAMESPACE
    END_ELEMENT SAX
    return
    Throws  IllegalStateException: START_ELEMENTEND_ELEMENT NAMESPACE

getNamespaceCount

    int getNamespaceCount()

    Returns the count of namespaces declared on this
    START ELEMENT or END ELEMENT, this method is only valid on
    a START ELEMENT, END ELEMENT or NAMESPACE. On an
    END ELEMENT the count is of the namespaces that are about to go
    out of scope. This is the equivalent of the information reported by
    SAX callback for an end element event.

    Returns:
    returns the number of namespace declarations on this specific
    element
    Throws:
    IllegalStateException - if this is not a START_ELEMENT,
    END_ELEMENT or NAMESPACE

public String getNamespacePrefix(int index)

    index  null
    index
    return
    Throws  IllegalStateException: START_ELEMENTEND_ELEMENT NAMESPACE

getNamespacePrefix
public String getNamespacePrefix(int index)

    Returns the prefix for the namespace declared at the index. Returns null if this is the default namespace declaration

    Parameters:
    index - the position of the namespace declaration

    Returns:
    returns the namespace prefix

    Throws:
    IllegalStateException - if this is not a START_ELEMENT, END_ELEMENT or NAMESPACE

public String getNamespaceURI(int index)

    Returns the uri for the namespace declared at the index.

    Parameters:
    index - the position of the namespace declaration

    Returns:
    returns the namespace uri

    Throws:
    IllegalStateException - if this is not a START_ELEMENT, END_ELEMENT or NAMESPACE

public javax.xml.namespace.NamespaceContext
getNamespaceContext()

transient next() reader

return

getNamespaceContext

NamespaceContext getNamespaceContext()

Returns a read only namespace context for the current position. The context is transient and only valid until a call to next() changes the state of the reader.

Returns:
return a namespace context

public int getEventType()

gETEVENType

int getEventType()

Returns an integer code that indicates the type of the event the cursor is pointing to.

public String getText()

getText

VARCHAR CHARACTERS COMMENT
ENTITY_REFERENCE CDATA SPACE DTD
ENTITY_REFERENCE CHARACTERS

return null

Throws IllegalStateException:
**getText**

`String getText()`

Returns the current value of the parse event as a string, this returns the string value of a CHARACTERS event, returns the value of a COMMENT, the replacement value for an ENTITY_REFERENCE, the string value of a CDATA section, the string value for a SPACE event, or the String value of the internal subset of the DTD. If an ENTITY_REFERENCE has been resolved, any character data will be reported as CHARACTERS events.

**Returns:**
the current text or null

**Throws:**
`IllegalStateException` - if this state is not a valid text state.

---

**public char[] getTextCharacters()**

`transient XMLStreamReader return`

`return` `transient` `XMLStreamReader`

```java
    return transient XMLStreamReader;
```

**Throws**
`IllegalStateException`: 

---

**getTextCharacters**

`char[] getTextCharacters()`

Returns an array which contains the characters from this event. This array should be treated as read-only and transient. I.e. the array will contain the text characters until the XMLStreamReader moves on to the next event. Attempts to hold onto the character array beyond that time or modify the contents of the array are breaches of the contract for this interface.

**Returns:**
the current text or an empty array

**Throws:**
`IllegalStateException` - if this state is not a valid text state.
public int getTextCharacters(int sourceStart, char[] target, int targetStart, int length) throws XMLStreamException

CHARACTERSSPACE CDATA "sourceStart" "targetStart" "target" "length" "sourceStart" 0 "sourceStart" 0

"length"

myBuffer = new char[length]; for (int sourceStart = 0; ; sourceStart += length) {
    int nCopied = stream.getTextCharacters(sourceStart, myBuffer, 0, length);
    if (nCopied < length) break;
} XML

XMLStreamException "targetStart" 0 "target"

Length 0 "targetStart + length" "target"

sourceStart

target

targetStart

length

return

Throws XMLStreamException: XML

Throws IndexOutOfBoundsException: targetStart < 0 

Throws IndexOutOfBoundsException: length < 0 targetStart + length

Throws UnsupportedOperationException:

Throws NullPointerException: null

getTextCharacters

int getTextCharacters(int sourceStart, char[] target, int targetStart, int length) throws XMLStreamException

Gets the the text associated with a CHARACTERS, SPACE or CDATA event. Text starting a "sourceStart" is copied into "target"
starting at "targetStart". Up to "length" characters are copied. The number of characters actually copied is returned. The "sourceStart" argument must be greater or equal to 0 and less than or equal to the number of characters associated with the event. Usually, one requests text starting at a "sourceStart" of 0. If the number of characters actually copied is less than the "length", then there is no more text. Otherwise, subsequent calls need to be made until all text has been retrieved. For example: int length = 1024; char[] myBuffer = new char[ length ]; for ( int sourceStart = 0 ; ; sourceStart += length ) { int nCopied = stream.getTextCharacters( sourceStart, myBuffer, 0, length ); if (nCopied < length) break; } XMLStreamException may be thrown if there are any XML errors in the underlying source. The "targetStart" argument must be greater than or equal to 0 and less than the length of "target", Length must be greater than 0 and "targetStart + length" must be less than or equal to length of "target".

Parameters:
- sourceStart - the index of the first character in the source array to copy
- target - the destination array
- targetStart - the start offset in the target array
- length - the number of characters to copy

Returns:
- the number of characters actually copied

Throws:
- XMLStreamException - if the underlying XML source is not well-formed
- IndexOutOfBoundsException - if targetStart < 0 or > than the length of target
- IndexOutOfBoundsException - if length < 0 or targetStart + length > length of target
- UnsupportedOperation Exception - if this method is not supported
- NullPointerException - if target is null

public int getTextStart()
**getTextStart**

```java
int getTextStart()
```

Returns the offset into the text character array where the first character (of this text event) is stored.

**Throws:**

*IllegalStateException* - if this state is not a valid text state.

**getTextLength**

```java
public int getTextLength()
```

**Throws**

IllegalStateException: 

**public String getEncoding()**

```java
null
```

```java
return null
```

**getEncoding**

```java
String getEncoding()
```
Return input encoding if known or null if unknown.

Returns: the encoding of this instance or null

public boolean hasText()
  true false CHARACTERSDTD ENTITY_REFERENCECOMMENTSPACE

hasText

boolean hasText()

Return true if the current event has text, false otherwise The following events have text: CHARACTERS, DTD, ENTITY_REFERENCE, COMMENT, SPACE

public Location getLocation()
  Location Location -1 publicId
systemId  null next()

getLocation

Location getLocation()

Return the current location of the processor. If the Location is unknown the processor should return an implementation of Location that returns -1 for the location and null for the publicId and systemId. The location information is only valid until next() is called.

public javax.xml.namespace.QName getName()
  START_ELEMENT END_ELEMENT QName

getName
**getLocalName**

Declares:

```
public String getLocalName()
```

**Returns:**

the (local) name of the current event. For START_ELEMENT or END_ELEMENT returns the (local) name of the current element. For ENTITY_REFERENCE it returns entity name. The current event must be START_ELEMENT or END_ELEMENT, or ENTITY_REFERENCE

**Throws:**

IllegalStateException - if this is not a START_ELEMENT or END_ELEMENT
Returns:
the localName

Throws:

IllegalStateException - if this not a START_ELEMENT, END_ELEMENT or ENTITY_REFERENCE

public boolean hasName()

START_ELEMENT  END_ELEMENT  true  false

hasName

boolean hasName()

returns true if the current event has a name (is a START_ELEMENT or END_ELEMENT) returns false otherwise

public String getNamespaceURI()

START_ELEMENT  END_ELEMENT  URI

null

return

URI null

getNamespaceURI

String getNamespaceURI()

If the current event is a START_ELEMENT or END_ELEMENT this method returns the URI of the prefix or the default namespace. Returns null if the event does not have a prefix.

Returns:
the URI bound to this elements prefix, the default namespace, or null
public String getPrefix()
null
        return null

getPrefix

String getPrefix()

Returns the prefix of the current event or null if the event does not have a prefix

Returns:
the prefix or null

public String getVersion()
xml xml null
        return XML null

getVersion

String getVersion()

Get the xml version declared on the xml declaration Returns null if none was declared

Returns:
the XML version or null

public boolean isStandalone()
xml
        return true false
isStandalone

boolean isStandalone()

Get the standalone declaration from the xml declaration

**Returns:**
true if this is standalone, or false otherwise

---

public boolean standaloneSet()

    return true false

standaloneSet

boolean standaloneSet()

Checks if standalone was set in the document

**Returns:**
true if standalone was set in the document, or false otherwise

---

public String getCharacterEncodingScheme()

    xml null

String getCharacterEncodingScheme()

    return null

getCharacterEncodingScheme

**Returns:**
Returns the character encoding declared on the xml declaration
Returns null if none was declared
the encoding declared in the document or null

```java
public String getPITarget()
```

```
return null
```

**getPITarget**

```java
String getPITarget()
```

Get the target of a processing instruction

**Returns:**

the target or null

```java
public String getPIData()
```

```
return null
```

**getPIData**

```java
String getPIData()
```

Get the data section of a processing instruction

**Returns:**

the data or null
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.stream Interface XMLStreamWriter

public interface XMLStreamWriter

XMLStreamWriter XMLXMLStreamWriter writeCharacters
<& >writeAttribute "
javax.xml.stream.isPrefixDefaulting false URI
javax.xml.stream.isPrefixDefaulting true XMLStreamWriter
URI

    version 1.0

See javax.xml.stream.XMLOutputFactory,
also javax.xml.stream.XMLStreamReader

The XMLStreamWriter interface specifies how to write XML. The
XMLStreamWriter does not perform well formedness checking on its
input. However the writeCharacters method is required to escape & , <
and > For attribute values the writeAttribute method will escape the
above characters plus " to ensure that all character content and attribute
values are well formed. Each NAMESPACE and ATTRIBUTE must be
individually written. If javax.xml.stream.isPrefixDefaulting is set to false it
is a fatal error if an element is written with namespace URI that has not
been bound to a prefix. If javax.xml.stream.isPrefixDefaulting is set to
true the XMLStreamWriter implementation must write a prefix for each
unbound URI that it encounters in the current scope.

Version:
    1.0
Author:
    Copyright (c) 2003 by BEA Systems. All Rights Reserved.
See Also:
    XMLOutputFactory, XMLStreamReader
### Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>void close()</code></td>
<td></td>
<td>Close this writer and free any resources associated with the writer.</td>
</tr>
<tr>
<td><code>void flush()</code></td>
<td></td>
<td>Write any cached data to the underlying output mechanism.</td>
</tr>
<tr>
<td><code>NamespaceContext getNamespaceContext()</code></td>
<td></td>
<td>Returns the current namespace context.</td>
</tr>
<tr>
<td><code>String getPrefix(String uri)</code></td>
<td></td>
<td>Gets the prefix the uri is bound to.</td>
</tr>
<tr>
<td><code>Object get(Property(String name))</code></td>
<td></td>
<td>Get the value of a feature/property from the underlying implementation.</td>
</tr>
<tr>
<td><code>void setDefaultNamespace(String uri)</code></td>
<td></td>
<td>Binds a URI to the default namespace This URI is bound in the scope of the current START_ELEMENT / END_ELEMENT pair.</td>
</tr>
<tr>
<td><code>void setNamespaceContext(NamespaceContext context)</code></td>
<td></td>
<td>Sets the current namespace context for prefix and uri bindings.</td>
</tr>
<tr>
<td><code>void setPrefix(String prefix, String uri)</code></td>
<td></td>
<td>Sets the prefix the uri is bound to.</td>
</tr>
<tr>
<td><code>void writeAttribute(String localName, String value)</code></td>
<td></td>
<td>Writes an attribute to the output stream without a prefix.</td>
</tr>
<tr>
<td><code>void writeAttribute(String namespaceURI, String localName, String value)</code></td>
<td></td>
<td>Writes an attribute to the output stream.</td>
</tr>
<tr>
<td><code>void writeAttribute(String prefix, String namespaceURI, String localName, String value)</code></td>
<td></td>
<td>Writes an attribute to the output stream.</td>
</tr>
<tr>
<td><code>void writeCData(String data)</code></td>
<td></td>
<td>Writes a CData section.</td>
</tr>
<tr>
<td><code>void writeCharacters(char[] text, int start, int len)</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Write text to the output

```java
void writeCharacters(String text)
    Write text to the output
```

```java
void writeComment(String data)
    Writes an xml comment with the data enclosed
```

```java
void writeDefaultNamespace(String namespaceURI)
    Writes the default namespace to the stream
```

```java
void writeDTD(String dtd)
    Write a DTD section.
```

```java
void writeEmptyElement(String localName)
    Writes an empty element tag to the output
```

```java
void writeEmptyElement(String namespaceURI, String localName)
    Writes an empty element tag to the output
```

```java
void writeEmptyElement(String prefix, String namespaceURI)
    Writes an empty element tag to the output
```

```java
void writeEndDocument()
    Closes any start tags and writes corresponding end tags.
```

```java
void writeEndElement()
    Writes an end tag to the output relying on the internal state of the writer to determine the prefix and local name of the event.
```

```java
void writeEntityRef(String name)
    Writes an entity reference
```

```java
void writeNamespace(String prefix, String namespaceURI)
    Writes a namespace to the output stream If the prefix argument to this method is the empty string, "xmlns", or null this method will delegate to writeDefaultNamespace
```

```java
void writeProcessingInstruction(String target)
    Writes a processing instruction
```

```java
void writeProcessingInstruction(String target, String data)
```
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>void writeStartDocument()</td>
<td>Write the XML Declaration.</td>
</tr>
<tr>
<td>void writeStartDocument(String version)</td>
<td>Write the XML Declaration.</td>
</tr>
<tr>
<td>void writeStartDocument(String encoding, String version)</td>
<td>Write the XML Declaration.</td>
</tr>
<tr>
<td>void writeStartElement(String localName)</td>
<td>Writes a start tag to the output.</td>
</tr>
<tr>
<td>void writeStartElement(String namespaceURI, String localName)</td>
<td>Writes a start tag to the output</td>
</tr>
<tr>
<td>void writeStartElement(String prefix, String localName, String namespaceURI)</td>
<td>Writes a start tag to the output</td>
</tr>
</tbody>
</table>

**Method Detail**

```java
public void writeStartElement(String localName) throws XMLStreamException {
    // Implementation
}
```

**writeStartElement**

```java
void writeStartElement(String localName) throws XMLStreamException {
    // Implementation
}
```

Writes a start tag to the output. All writeStartElement methods open a new scope in the internal namespace context. Writing the corresponding EndElement causes the scope to be closed.

**Parameters:**
public void writeStartElement(String namespaceURI, String localName) throws XMLStreamException

namespaceURI null
localName null

Writes a start tag to the output

Parameters:
namespaceURI - the namespaceURI of the prefix to use, may not be null
localName - local name of the tag, may not be null

Throws:
XMLStreamException - if the namespace URI has not been bound to a prefix and javax.xml.stream.isPrefixDefaulting has not been set to true

public void writeStartElement(String prefix, String localName, String namespaceURI) throws XMLStreamException

localName null
prefix null
writeStartElement

```java
void writeStartElement(String prefix,
                       String localName,
                       String namespaceURI)
  throws XMLStreamException
```

Writes a start tag to the output

**Parameters:**
- `localName` - local name of the tag, may not be null
- `prefix` - the prefix of the tag, may not be null
- `namespaceURI` - the uri to bind the prefix to, may not be null

**Throws:**
- `XMLStreamException`

writeEmptyElement

```java
public void writeEmptyElement(String namespaceURI,
                               String localName)
  throws XMLStreamException
```

```
namespaceURI URI null
localName URI null
Throws XMLStreamException: URI
javax.xml.stream.isPrefixDefaulting true
```

**Parameters:**
- `namespaceURI` - URI null
- `localName` - null

**Throws:**
- `XMLStreamException`

**javax.xml.stream.isPrefixDefaulting** is set to `true`.
namespaceURI - the uri to bind the tag to, may not be null
localName - local name of the tag, may not be null

Throws:
XMLStreamException - if the namespace URI has not been bound to a prefix and javax.xml.stream.isPrefixDefaulting has not been set to true

public void writeEmptyElement(String prefix, String localName, String namespaceURI) throws XMLStreamException

<table>
<thead>
<tr>
<th>prefix</th>
<th>null</th>
</tr>
</thead>
<tbody>
<tr>
<td>localName</td>
<td>null</td>
</tr>
<tr>
<td>namespaceURI</td>
<td>URI null</td>
</tr>
</tbody>
</table>

Throws: XMLStreamException:

writeEmptyElement

void writeEmptyElement(String prefix, String localName, String namespaceURI) throws XMLStreamException

Writes an empty element tag to the output

Parameters:
prefix - the prefix of the tag, may not be null
localName - local name of the tag, may not be null
namespaceURI - the uri to bind the tag to, may not be null

Throws: XMLStreamException

public void writeEmptyElement(String localName) throws XMLStreamException

writeEmptyElement

void writeEmptyElement(String prefix, String localName, String namespaceURI) throws XMLStreamException

Throws: XMLStreamException:

---

public void writeEmptyElement(String localName) throws XMLStreamException
writeEmptyElement

```java
void writeEmptyElement(String localName)
  throws XMLStreamException
```

Writes an empty element tag to the output

**Parameters:**
localName - local name of the tag, may not be null

**Throws:**
XMLStreamException

writeEndElement

```java
public void writeEndElement() throws XMLStreamException
```

writes an end tag to the output relying on the internal state of the writer to determine the prefix and local name of the event.

**Throws:**
XMLStreamException

writeEndDocument

```java
public void writeEndDocument() throws XMLStreamException
```

**Throws:**
XMLStreamException
writeEndDocument

void writeEndDocument() throws XMLStreamException

Closes any start tags and writes corresponding end tags.

Throws:
XMLStreamException

public void close() throws XMLStreamException

writer writer

Throws
XMLStreamException:

close

void close() throws XMLStreamException

Close this writer and free any resources associated with the writer. This must not close the underlying output stream.

Flushes:
XMLStreamException

public void flush() throws XMLStreamException

Flushes
XMLStreamException:

flush

void flush()
throws XMLStreamException

Write any cached data to the underlying output mechanism.

Throws:
   XMLStreamException

public void writeAttribute(String localName, String value)
throws XMLStreamException

localName
value

Throws     IllegalStateException:  Attribute
            XMLStreamException:

writeAttribute

void writeAttribute(String localName, String value)
throws XMLStreamException

Writes an attribute to the output stream without a prefix.

Parameters:
   localName - the local name of the attribute
   value - the value of the attribute

Throws:
   IllegalStateException - if the current state does not allow
   Attribute writing
   XMLStreamException

public void writeAttribute(String prefix, String namespaceURI, String localName, String value)
throws XMLStreamException
void writeAttribute(String prefix, String namespaceURI, String localName, String value)
throws XMLStreamException

writes an attribute to the output stream

Parameters:
  prefix - the prefix for this attribute
  namespaceURI - the uri of the prefix for this attribute
  localName - the local name of the attribute
  value - the value of the attribute

Throws:
  IllegalStateException - if the current state does not allow Attribute writing
  XMLStreamException - if the namespace URI has not been bound to a prefix and javax.xml.stream.isPrefixDefaulting has not been set to true

public void writeAttribute(String namespaceURI, String localName, String value) throws XMLStreamException
writeAttribute

```java
void writeAttribute(String namespaceURI,
                     String localName,
                     String value)
throws XMLStreamException
```

Writes an attribute to the output stream

**Parameters:**
- namespaceURI - the uri of the prefix for this attribute
- localName - the local name of the attribute
- value - the value of the attribute

**Throws:**
- `IllegalStateException` - if the current state does not allow Attribute writing
- `XMLStreamException` - if the namespace URI has not been bound to a prefix and `javax.xml.stream.isPrefixDefaulting` has not been set to true

writeDefaultNamespace

```java
public void writeNamespace(String prefix, String namespaceURI) throws XMLStreamException
```

prefix "xmlns" null

writeNamespace

```java
prefix
namespaceURI
```

**Throws:**
- `IllegalStateException` - if the current state does not allow Namespace writing
- `XMLStreamException` - if the namespace URI has not been bound to a prefix and `javax.xml.stream.isPrefixDefaulting` has not been set to true
void writeNamespace(String prefix, String namespaceURI) throws XMLStreamException

Writes a namespace to the output stream. If the prefix argument to this method is the empty string, "xmlns", or null this method will delegate to writeDefaultNamespace

**Parameters:**
- prefix - the prefix to bind this namespace to
- namespaceURI - the uri to bind the prefix to

**Throws:**
- IllegalStateException - if the current state does not allow Namespace writing
- XMLStreamException

---

public void writeDefaultNamespace(String namespaceURI) throws XMLStreamException

**namespaceURI**

**Throws**
- IlleagalStateException: Namespace writing
- XMLStreamException

---

writeDefaultNamespace

void writeDefaultNamespace(String namespaceURI) throws XMLStreamException

Writes the default namespace to the stream

**Parameters:**
- namespaceURI - the uri to bind the default namespace to

**Throws:**
- IlleagalStateException - if the current state does not allow Namespace writing
- XMLStreamException
public void writeComment(String data) throws XMLStreamException

 XML
data
null
Throws
XMLStreamException:

writeComment

void writeComment(String data)
throws XMLStreamException

Writes an xml comment with the data enclosed

Parameters:
data - the data contained in the comment, may be null

Throws:
XMLStreamException

public void writeProcessingInstruction(String target)
throws XMLStreamException

target
null
Throws
XMLStreamException:

writeProcessingInstruction

void writeProcessingInstruction(String target)
throws XMLStreamException

Writes a processing instruction

Parameters:
target - the target of the processing instruction, may not be null

Throws:
XMLStreamException
public void writeProcessingInstruction(String target, String data) throws XMLStreamException

    target null
    data null

    Throws XMLStreamException:

writeProcessingInstruction

void writeProcessingInstruction(String target, String data)
throws XMLStreamException

    Writes a processing instruction

    Parameters:
    target - the target of the processing instruction, may not be null
    data - the data contained in the processing instruction, may not be null

    Throws:
    XMLStreamException

public void writeCData(String data) throws XMLStreamException

    CData

    data CData null

    Throws XMLStreamException:

writeCData

void writeCData(String data)
throws XMLStreamException

    Writes a CData section
public void writeDTD(String dtd) throws XMLStreamException

Write a DTD section. This string represents the entire doctypedecl production from the XML 1.0 specification.

Parameters:
  dtd - the DTD to be written

Throws:
  XMLStreamException

public void writeEntityRef(String name) throws XMLStreamException

void writeEntityRef(String name)

Throws:
  XMLStreamException
Wants to write an entity reference

**Parameters:**
- **name** - the name of the entity

**Throws:**
- [XMLStreamException](#)

---

```java
public void writeStartDocument() throws XMLStreamException

XML XML 1.0 utf-8

Throws XMLStreamException:
```

**writeStartDocument**

```java
void writeStartDocument() throws XMLStreamException

Write the XML Declaration. Defaults the XML version to 1.0, and the encoding to utf-8

Throws:
- XMLStreamException
```

---

```java
public void writeStartDocument(String version) throws XMLStreamException

XML XML 1.0

version XML

Throws XMLStreamException:
```

**writeStartDocument**

```java
void writeStartDocument(String version) throws XMLStreamException

Throws XMLStreamException
```
Write the XML Declaration. Defaults the XML version to 1.0

Parameters:
version - version of the xml document

Throws:
XMLStreamException

public void writeStartDocument(String encoding, String version) throws XMLStreamException
XML XMLOutputFactory
XMLStreamWriter
encoding  XML
version  XML

Throws  XMLStreamException:

writeStartDocument

void writeStartDocument(String encoding, String version) throws XMLStreamException

Write the XML Declaration. Note that the encoding parameter does not set the actual encoding of the underlying output. That must be set when the instance of the XMLStreamWriter is created using the XMLOutputFactory

Parameters:
encoding - encoding of the xml declaration
version - version of the xml document

Throws:
XMLStreamException

public void writeCharacters(String text) throws XMLStreamException
XMLStreamException
writeCharacters

void writeCharacters(String text)
throws XMLStreamException

Write text to the output

Parameters:
  text - the value to write

Throws:
  XMLStreamException

public void writeCharacters(char[] text, int start, int len)
throws XMLStreamException

  text
  start
  len

Throws
  XMLStreamException:

writeCharacters

void writeCharacters(char[] text, int start, int len)
throws XMLStreamException

Write text to the output

Parameters:
  text - the value to write
  start - the starting position in the array
  len - the number of characters to write
public String getPrefix(String uri) throws XMLStreamException
URI

    return null

getPrefix

String getPrefix(String uri)
throws XMLStreamException

    Gets the prefix the uri is bound to

Returns:
    the prefix or null

Throws:
    XMLStreamException

public void setPrefix(String prefix, String uri) throws XMLStreamException
URI START_ELEMENT / END_ELEMENT
START_ELEMENT
    prefix URI null
    uri URI null

setPrefix

void setPrefix(String prefix,
    String uri)
throws XMLStreamException
Sets the prefix the uri is bound to. This prefix is bound in the scope of the current START_ELEMENT / END_ELEMENT pair. If this method is called before a START_ELEMENT has been written the prefix is bound in the root scope.

Parameters:
  prefix - the prefix to bind to the uri, may not be null
  uri - the uri to bind to the prefix, may be null

Throws:
  XMLStreamException

public void setDefaultNamespace(String uri) throws XMLStreamException

Binds a URI to the default namespace This URI is bound in the scope of the current START_ELEMENT / END_ELEMENT pair. If this method is called before a START_ELEMENT has been written the uri is bound in the root scope.

Parameters:
  uri - the uri to bind to the default namespace, may be null

Throws:
  XMLStreamException

public void setNamespaceContext(javax.xml.namespace.NamespaceContext


setNamespaceContext

```java
type void setNamespaceContext(NamespaceContext context)
throws XMLStreamException
```

Sets the current namespace context for prefix and uri bindings. This context becomes the root namespace context for writing and will replace the current root namespace context. Subsequent calls to setPrefix and setDefaultNamespace will bind namespaces using the context passed to the method as the root context for resolving namespaces. This method may only be called once at the start of the document. It does not cause the namespaces to be declared. If a namespace URI to prefix mapping is found in the namespace context it is treated as declared and the prefix may be used by the StreamWriter.

**Parameters:**
- `context` - the namespace context to use for this writer, may not be null

**Throws:**
- `XMLStreamException`

---

```java
public javax.xml.namespace.NamespaceContext getNamespaceContext()
```

```java
return NamespaceContext
```

getNamespaceContext
NamespaceContext getNamespaceContext()

Returns the current namespace context.

Returns:
the current NamespaceContext

public Object getProperty(String name) throws
IllegalArgumentException

/)

        name null
        return
        Throws IllegalArgumentException:
        Throws NullPointerException: null

getProperty

Object getProperty(String name)
throws IllegalArgumentException

Get the value of a feature/property from the underlying implementation

Parameters:
name - The name of the property, may not be null

Returns:
The value of the property

Throws:
IllegalArgumentException - if the property is not supported
NullPointerException - if the name is null
Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS :
javax.xml.bind.annotation Annotation Type XmlTransient

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlTransient

Implements: Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

JavaBean XML

@XmlTransient JavaBean / JavaBean

JavaBean

@XmlTransient

• JavaBean

@XmlTransient JAXB

javax.xml.bind.package javadoc ""

JavaBean

// Example: Code fragment
public class USAddress {

    // The field name "name" collides with the property name
    // obtained by bean decapitalization of getName() below
    @XmlTransient public String name;

    String getName() {..};
    String setName() {..};

}
Prevents the mapping of a JavaBean property to XML representation.

The `@XmlTransient` annotation is useful for resolving name collisions between a JavaBean property name and a field name or preventing the mapping of a field/property. A name collision can occur when the decapitalized JavaBean property name and a field name are the same. If the JavaBean property refers to the field, then the name collision can be resolved by preventing the mapping of either the field or the JavaBean property using the `@XmlTransient` annotation.

**Usage**

The `@XmlTransient` annotation can be used with the following program elements:

- a JavaBean property
- field

`@XmlTransient` is mutually exclusive with all other JAXB defined annotations.


**Example:** Resolve name collision between JavaBean property and field name

```java
// Example: Code fragment
class USAddress {
```
// The field name "name" collides with the property name
// obtained by bean decapitalization of getName() below
@XmlTransient public String name;

String getName() { ..};
String setName() { ..};
}

<!-- Example: XML Schema fragment -->
<xs:complexType name="USAddress">
  <xs:sequence>
    <xs:element name="name" type="xs:string"/>
  </xs:sequence>
</xs:complexType>

Since:
  JAXB2.0
Version:
  $Revision: 1.8$
Author:
  Sekhar Vajjhala, Sun Microsystems, Inc.

Overview Package Tree Deprecated Index Help
PREV CLASS NEXT CLASS SUMMARY: REQUIRED | OPTIONAL
FRAMES NO FRAMES DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
javax.xml.bind.annotation Annotation Type XmlType

@Retention(value=RUNTIME)
@Target(value=TYPE)
public @interface XmlType

Implements: Annotation
Inner classes: XmlType.DEFAULT
@Retention(value=RUNTIME)
@Target(value=TYPE)

XML

@XmlType

@XmlElement

javax.xml.bind.package javadoc ""

XML

XML XML XML JavaBean
@XmlType(name="")

• @ XmlRootElement 3
• JavaBean @XmlType(name="") 4
• JavaBean @XmlType(name="") 5
- @XmlType(name="") XmlName()
- propOrder()
- XML targetnamespace namespace()

XML

- @XmlValue XML @ XMLValue

XML

- -> :
- [x]+ :x
- [ @XmlValue property ]: @XmlValue JavaBean
- X:

<table>
<thead>
<tr>
<th>propOrder</th>
<th>ClassBody</th>
<th>ComplexType</th>
<th>SimpleType</th>
</tr>
</thead>
<tbody>
<tr>
<td>{}</td>
<td>[property]+ -&gt; elements</td>
<td>complexcontent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xs:all</td>
<td></td>
<td></td>
</tr>
<tr>
<td>non empty</td>
<td>[property]+ -&gt; elements</td>
<td>complexcontent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>xs:sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>no property -&gt; element</td>
<td>complexcontent empty</td>
<td></td>
</tr>
<tr>
<td></td>
<td>sequence</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>1 [ @XmlValue property ]</td>
<td>simplecontent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp;&amp; [property]+ -&gt; attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>1 [ @XmlValue property ]</td>
<td>simplecontent</td>
<td></td>
</tr>
<tr>
<td></td>
<td>&amp;&amp; no properties -&gt; attribute</td>
<td></td>
<td>simpletype</td>
</tr>
</tbody>
</table>

enumeration facet XML
factoryClass()}
1  xs:sequence  JavaBean

@XmlType(propOrder={"street", "city", "state", "zip", "name" })
public class USAddress {
    String getName() { .. };
    void setName(String) { .. };

    String getStreet() { .. };
    void setStreet(String) { .. };

    String getCity() { .. };
    void setCity(String) { .. };

    String getState() { .. };
    void setState(String) { .. };

    java.math.BigDecimal getZip() { .. };
    void setZip(java.math.BigDecimal) { .. };
}

<!-- XML Schema mapping for USAddress -->
<xs:complexType name="USAddress">
    <xs:sequence>
        <xs:element name="street" type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="state" type="xs:string"/>
        <xs:element name="zip" type="xs:decimal"/>
        <xs:element name="name" type="xs:string"/>
    </xs:all>
</xs:complexType>

2  xs:all

@XmlType(propOrder={})
public class USAddress { ...}

<!-- XML Schema mapping for USAddress -->
<xs:complexType name="USAddress">
    <xs:all>
        <xs:element name="name" type="xs:string"/>
        <xs:element name="street" type="xs:string"/>
    </xs:all>
</xs:complexType>
3

@XmlElement
@XmlType(name="")
public class USAddress { ...}

<!-- XML Schema mapping for USAddress -->
<xs:element name="USAddress">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="name" type="xs:string"/>
      <xs:element name="street" type="xs:string"/>
      <xs:element name="city" type="xs:string"/>
      <xs:element name="state" type="xs:string"/>
      <xs:element name="zip" type="xs:decimal"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>

4

//Example: Code fragment
public class Invoice {
    USAddress addr;
    ...
}

@XmlElement(name="")
public class USAddress { ... } }

<!-- XML Schema mapping for USAddress -->
<xs:complexType name="Invoice">
  <xs:sequence>
    <xs:element name="addr">
      <xs:complexType>
        <xs:element name="name", type="xs:string"/>
        <xs:element name="city", type="xs:string"/>
        <xs:element name="city" type="xs:string"/>
        <xs:element name="state" type="xs:string"/>
        <xs:element name="zip" type="xs:decimal"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
// Example: Code fragment
public class Item {
    public String name;
    @XmlAttribute
    public USPrice price;
}

// map class to anonymous simple type.
@XmlType(name="")
public class USPrice {
    @XmlValue
    public java.math.BigDecimal price;
}

<!-- Example: XML Schema fragment -->
<xs:complexType name="Item">
    <xs:sequence>
        <xs:element name="name" type="xs:string"/>
        <xs:attribute name="price">
            <xs:simpleType>
                <xs:restriction base="xs:decimal"/>
            </xs:simpleType>
        </xs:attribute>
    </xs:sequence>
</xs:complexType>

6 factoryClass factoryMethod

@XmlType(name="USAddressType", factoryClass=USAddressFactory.class,
factoryMethod="getUSAddress")
public class USAddress {
    private String city;
    private String name;
    private String state;
    private String street;
    private int zip;

    public USAddress(String name, String street, String city,
        String state, int zip) {

```java
this.name = name;
this.street = street;
this.city = city;
this.state = state;
this.zip = zip;
}
}

public class USAddressFactory {
    public static USAddress getUSAddress() {
        return new USAddress("Mark Baker", "23 Elm St", "Dayton", "OH", 90952);
    }
}

7 factoryMethod factoryClass

@XmlElement(name="USAddressType", factoryMethod="getNewInstance")
public class USAddress {

    private String city;
    private String name;
    private String state;
    private String street;
    private int zip;

    private USAddress() {} 

    public static USAddress getNewInstance() {
        return new USAddress();
    }
}

since JAXB2.0 version $Revision: 1.20$

See javax.xml.bind.annotation.XmlElement,
javax.xml.bind.annotation.XmlAttribute,
javax.xml.bind.annotation.XmlValue,
javax.xml.bind.annotation.XmlSchema

Maps a class or an enum type to a XML Schema type.
Usage

The @XmlType annotation can be used with the following program elements:

- a top level class
- an enum type


Mapping a Class

A class maps to a XML Schema type. A class is a data container for values represented by properties and fields. A schema type is a data container for values represented by schema components within a schema type's content model (e.g. model groups, attributes etc).

To be mapped, a class must either have a public no-arg constructor or a static no-arg factory method. The static factory method can be specified in factoryMethod() and factoryClass() annotation elements. The static factory method or the no-arg constructor is used during unmarshalling to create an instance of this class. If both are present, the static factory method overrides the no-arg constructor.

A class maps to either a XML Schema complex type or a XML Schema simple type. The XML Schema type is derived based on the mapping of JavaBean properties and fields contained within the class. The schema type to which the class is mapped can either be named or anonymous. A class can be mapped to an anonymous schema type by annotating the class with @XmlType(name="").

Either a global element, local element or a local attribute can be associated with an anonymous type as follows:

- **global element**: A global element of an anonymous type can be derived by annotating the class with @XmlElement. See Example 3 below.
- **local element**: A JavaBean property that references a class
annotated with @XmlType(name="") and is mapped to the element associated with the anonymous type. See Example 4 below.

- **attribute**: A JavaBean property that references a class annotated with @XmlType(name="") and is mapped to the attribute associated with the anonymous type. See Example 5 below.

**Mapping to XML Schema Complex Type**

- If class is annotated with @XmlType(name="") , it is mapped to an anonymous type otherwise, the class name maps to a complex type name. The XmlName() annotation element can be used to customize the name.
- Properties and fields that are mapped to elements are mapped to a content model within a complex type. The annotation element propOrder() can be used to customize the content model to be xs:all or xs:sequence. It is used for specifying the order of XML elements in xs:sequence.
- Properties and fields can be mapped to attributes within the complex type.
- The targetnamespace of the XML Schema type can be customized using the annotation element namespace().

**Mapping class to XML Schema simple type**

A class can be mapped to a XML Schema simple type using the @XmlValue annotation. For additional details and examples, see @XmlValue annotation type.

The following table shows the mapping of the class to a XML Schema complex type or simple type. The notational symbols used in the table are:

- <-> : represents a mapping
- [x]+ : one or more occurrences of x
- [ @XmlValue property ]: JavaBean property annotated with @XmlValue
- X : don't care

<table>
<thead>
<tr>
<th>Target</th>
<th>propOrder</th>
<th>ClassBody</th>
<th>ComplexType</th>
<th>SimpleType</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>[property]+ &lt;-&gt;</td>
<td>complexcontent</td>
<td></td>
</tr>
</tbody>
</table>
Mapping an enum type

An enum type maps to a XML schema simple type with enumeration facets. The following annotation elements are ignored since they are not meaningful: propOrder(), factoryMethod(), factoryClass().

Usage with other annotations

This annotation can be used with the following annotations: XmlRootElement, XmlAccessorType, XmlAccessorOrder, XmlAccessorType, XmlEnum. However, XmlAccessorOrder and XmlAccessorType are ignored when this annotation is used on an enum type.

Example 1: Map a class to a complex type with xs:sequence with a customized ordering of JavaBean properties.

```java
@XmlElement(propOrder={"street", "city", "state", "zip", "name"})
public class USAddress {
    String getName() {..};
    void setName(String) {..};

    String getStreet() {..};
    void setStreet(String) {..};
```
String getCity() {..};
void setCity(String) {..};

String getState() {..};
void setState(String) {..};

java.math.BigDecimal getZip() {..};
void setZip(java.math.BigDecimal) {..};

</xs:complexType>
</xs:typedef>

Example 2: Map a class to a complex type with xs:all

@XmlElement(propOrder={})
public class USAddress { ...}

</xs:complexType>
</xs:typedef>

Example 3: Map a class to a global element with an anonymous type.

@XmlRootElement
@XmlType(name="")
public class USAddress { ...}

</xs:complexType>
Example 4: Map a property to a local element with anonymous type.

```java
//Example: Code fragment
class Invoice {
    USAddress addr;
    ...
}
@XmlType(name="")
class USAddress {
    ...
}

<!-- XML Schema mapping for USAddress -->
<complexType name="Invoice">
    <sequence>
        <element name="addr">
            <complexType>
                <element name="name", type="xs:string"/>
                <element name="city", type="xs:string"/>
                <element name="state" type="xs:string"/>
                <element name="zip" type="xs:decimal"/>
            </complexType>
        ...
    </sequence>
</complexType>
```

Example 5: Map a property to an attribute with anonymous type.

```java
//Example: Code fragment
public class Item {
    public String name;
    @XmlAttribute
    public USPrice price;
}

// map class to anonymous simple type.
@XmlType(name="")
public class USPrice {
    @XmlValue
    public java.math.BigDecimal price;
}

<!-- Example: XML Schema fragment -->
<xsd:complexType name="Item">
    <xsd:sequence>
        <xsd:element name="name" type="xsd:string"/>
        <xsd:attribute name="price">
            <xsd:simpleType>
                <xsd:restriction base="xsd:decimal"/>
            </xsd:simpleType>
        </xsd:attribute>
    </xsd:sequence>
</xsd:complexType>

Example 6: Define a factoryClass and factoryMethod

@XmlType(name="USAddressType", factoryClass=USAddressFactory.class,
        factoryMethod="getUSAddress")
public class USAddress {
    private String city;
    private String name;
    private String state;
    private String street;
    private int zip;

    public USAddress(String name, String street, String city,
                      String state, int zip) {
        this.name = name;
        this.street = street;
        this.city = city;
        this.state = state;
        this.zip = zip;
    }
}

public class USAddressFactory {
    public static USAddress getUSAddress() {
        return new USAddress("Mark Baker", "23 Elm St", "Dayton", "OH", 90952);
    }
}
Example 7: Define factoryMethod and use the default factoryClass

```java
@XmlType(name="USAddressType", factoryMethod="getNewInstance")
public class USAddress {

    private String city;
    private String name;
    private String state;
    private String street;
    private int zip;

    private USAddress() {}

    public static USAddress getNewInstance(){
        return new USAddress();
    }
}
```

Since: JAXB2.0
Version: $Revision: 1.20$
Author: Sekhar Vajjhala, Sun Microsystems, Inc.
See Also: XElement,XmlAttribute,XmlValue,XmlSchema

---

### Optional Element Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>factoryClass</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Class containing a no-arg factory method for creating an instance of this class.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>factoryMethod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of a no-arg factory method in the class specified in factoryClass().</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the XML Schema type which the class is mapped.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>String</th>
<th>namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the target namespace of the XML Schema type.</td>
</tr>
<tr>
<td>propOrder</td>
<td>Specifies the order for XML Schema elements when class is mapped to a XML Schema complex type.</td>
</tr>
</tbody>
</table>

abstract public String name()

XML

name

public abstract String name

Name of the XML Schema type which the class is mapped.

Default:

"##default"

abstract public String[] propOrder()

XML XML

propOrder

propOrder JavaBean JavaBean Java
JavaBean JavaBean XML

XML JavaBean

propOrder JavaBean transient

JavaBean @ XmlAccessorOrder
propOrder

public abstract String[] propOrder

Specifies the order for XML Schema elements when class is mapped to a XML Schema complex type.

Refer to the table for how the propOrder affects the mapping of class

The propOrder is a list of names of JavaBean properties in the class. Each name in the list is the name of a Java identifier of the JavaBean property. The order in which JavaBean properties are listed is the order of XML Schema elements to which the JavaBean properties are mapped.

All of the JavaBean properties being mapped to XML Schema elements must be listed.

A JavaBean property or field listed in propOrder must not be transient or annotated with @XmlTransient.

The default ordering of JavaBean properties is determined by @XmlAccessorOrder.

Default:


abstract public String namespace()

XML

namespace

public abstract String namespace

Name of the target namespace of the XML Schema type. By default, this is the target namespace to which the package containing the class is mapped.
abstract public Class\(<T>\) factoryClass()

factoryClass DEFAULT.class factoryMethod ""

factoryClass DEFAULT.class factoryMethod ""
factoryMethod

factoryClass DEFAULT.class factoryMethod ""
factoryClass

Class containing a no-arg factory method for creating an instance of this class. The default is this class.

If factoryClass is DEFAULT.class and factoryMethod is "", then there is no static factory method.

If factoryClass is DEFAULT.class and factoryMethod is not "", then factoryMethod is the name of a static factory method in this class.

If factoryClass is not DEFAULT.class, then factoryMethod must not be "" and must be the name of a static factory method specified in factoryClass.

Default:
javax.xml.bind.annotation.XmlType.DEFAULT.class
abstract public String factoryMethod()

factoryClass factoryClass()

factoryMethod

public abstract String factoryMethod

Name of a no-arg factory method in the class specified in factoryClass factoryClass().

Default:

""

Overview Package Tree Deprecated Index Help

PREV CLASS NEXT CLASS SUMMARY: REQUIRED | OPTIONAL

FRAMES NO FRAMES DETAIL: ELEMENT

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Class XMLType

public class XMLType

extends Object

XML SOAP 1.1

version 1.0

Constants for common XML Schema and SOAP 1.1 types.

Version: 1.0

Author: Rahul Sharma

Field Summary

<table>
<thead>
<tr>
<th>static QName</th>
<th>SOAP_ARRAY</th>
<th>The name of the SOAP-ENC:Array type.</th>
</tr>
</thead>
<tbody>
<tr>
<td>static QName</td>
<td>SOAP_BASE64</td>
<td>The name of the SOAP-ENC:base64 type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_BOOLEAN</td>
<td>The name of the SOAP-ENC:boolean type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_BYTE</td>
<td>The name of the SOAP-ENC:byte type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_DOUBLE</td>
<td>The name of the SOAP-ENC:double type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_FLOAT</td>
<td>The name of the SOAP-ENC:float type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_INT</td>
<td>The name of the SOAP-ENC:int type.</td>
</tr>
<tr>
<td>-------------</td>
<td>----------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_LONG</td>
<td>The name of the SOAP-ENC:long type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_SHORT</td>
<td>The name of the SOAP-ENC:short type.</td>
</tr>
<tr>
<td>static QName</td>
<td>SOAP_STRING</td>
<td>The name of the SOAP-ENC:string type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_BASE64</td>
<td>The name of the xsd:base64Binary type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_BOOLEAN</td>
<td>The name of the xsd:boolean type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_BYTE</td>
<td>The name of the xsd:byte type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_DATETIME</td>
<td>The name of the xsd:dateTime type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_DECIMAL</td>
<td>The name of the xsd:decimal type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_DOUBLE</td>
<td>The name of the xsd:double type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_FLOAT</td>
<td>The name of the xsd:float type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_HEXBINARY</td>
<td>The name of the xsd:hexBinary type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_INT</td>
<td>The name of the xsd:int type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_INTEGER</td>
<td>The name of the xsd:integer type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_LONG</td>
<td>The name of the xsd:long type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_QNAME</td>
<td>The name of the xsd:QName type.</td>
</tr>
<tr>
<td>static QName</td>
<td>XSD_SHORT</td>
<td>The name of the xsd:short type.</td>
</tr>
</tbody>
</table>
Constructor Summary

XMLType()

Method Summary

Methods inherited from class java.lang.Object
cloned, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

Field Detail

XSD_STRING

public static final QName XSD_STRING

The name of the xsd:string type.

XSD_FLOAT

public static final QName XSD_FLOAT

The name of the xsd:float type.
XSD_BOOLEAN

public static final QName XSD_BOOLEAN

   The name of the xsd:boolean type.

XSD_DOUBLE

public static final QName XSD_DOUBLE

   The name of the xsd:double type.

XSD_INTEGER

public static final QName XSD_INTEGER

   The name of the xsd:integer type.

XSD_INT

public static final QName XSD_INT

   The name of the xsd:int type.

XSD_LONG

public static final QName XSD_LONG
The name of the xsd:long type.

XSD_SHORT

public static final QName XSD_SHORT

The name of the xsd:short type.

XSD_DECIMAL

public static final QName XSD_DECIMAL

The name of the xsd:decimal type.

XSD_BASE64

public static final QName XSD_BASE64

The name of the xsd:base64Binary type.

XSD_HEXBINARY

public static final QName XSD_HEXBINARY

The name of the xsd:hexBinary type.
XSD_BYTE
public static final QName XSD_BYTE

The name of the xsd:byte type.

XSD_DATETIME
public static final QName XSD_DATETIME

The name of the xsd:dateTime type.

XSD_QNAME
public static final QName XSD_QNAME

The name of the xsd:QName type.

SOAP_STRING
public static final QName SOAP_STRING

The name of the SOAP-ENC:string type.

SOAP_BOOLEAN
public static final QName SOAP_BOOLEAN
The name of the SOAP-ENC:boolean type.

SOAP_DOUBLE
public static final QName SOAP_DOUBLE

The name of the SOAP-ENC:double type.

SOAP_BASE64
public static final QName SOAP_BASE64

The name of the SOAP-ENC:base64 type.

SOAP_FLOAT
public static final QName SOAP_FLOAT

The name of the SOAP-ENC:float type.

SOAP_INT
public static final QName SOAP_INT

The name of the SOAP-ENC:int type.
SOAP_LONG

public static final QName SOAP_LONG

     The name of the SOAP-ENC:long type.

-----------------------------

SOAP_SHORT

public static final QName SOAP_SHORT

     The name of the SOAP-ENC:short type.

-----------------------------

SOAP_BYTE

public static final QName SOAP_BYTE

     The name of the SOAP-ENC:byte type.

-----------------------------

SOAP_ARRAY

public static final QName SOAP_ARRAY

     The name of the SOAP-ENC:Array type.

Constructors

Constructor Detail

public XMLType()
public XMLType()
javax.xml.bind.annotation  Class XmlType.DEFAULT

java.lang.Object
   ↓ javax.xml.bind.annotation.XmlType.DEFAULT

Enclosing class:
   XmlType

public static final class XmlType.DEFAULT
extends Object

Contained within: XmlType

   factoryClass()   XmlType

Used in XmlType.factoryClass() to signal that either factory method is not used or that it's in the class with this XmlType itself.

Constructor Summary

<table>
<thead>
<tr>
<th>Constructor Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlType.DEFAULT()</td>
</tr>
</tbody>
</table>

Method Summary

<table>
<thead>
<tr>
<th>Method Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Methods inherited from class java.lang.Object</td>
</tr>
<tr>
<td>clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait</td>
</tr>
</tbody>
</table>

Constructor Detail
public XmlType.DEFAULT()
<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Tree</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
<tbody>
<tr>
<td>PREV CLASS</td>
<td>NEXT CLASS</td>
<td>FRAMES</td>
<td>NO FRAMES</td>
<td>SUMMARY: REQUIRED</td>
<td>OPTIONAL</td>
</tr>
</tbody>
</table>
javax.xml.bind.annotation  Annotation Type XmlValue

@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})
public @interface XmlValue

**Implements:** Annotation
@Retention(value=RUNTIME)
@Target(value={FIELD, METHOD})

simpleContent XML XML

@XmlValue

- JavaBean
- static transient

javax.xml.bind.package javadoc ""

- @XmlValue
- @XmlValue XmlList XmlList XmlValue
- XML

JavaBean XML
JavaBean simpleContent

1 XML simpleType

    // Example 1: Code fragment
    public class USPrice {
        @XmlValue
        public java.math.BigDecimal price;
    }
<!--
Example 1: XML Schema fragment -->
<xs:simpleType name="USPrice">
  <xs:restriction base="xs:decimal"/>
</xs:simpleType>

2 simpleContent XML complexType

// Example 2: Code fragment
public class InternationalPrice {
  @XmlValue
  public java.math.BigDecimal price;

  @XmlAttribute
  public String currency;
}

<!--
Example 2: XML Schema fragment -->
<xs:complexType name="InternationalPrice">
  <xs:simpleContent>
    <xs:extension base="xs:decimal">
      <xs:attribute name="currency" type="xs:string"/>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>

since JAXB2.0
version $Revision: 1.6 $
See also javax.xml.bind.annotation.XmlType

Enables mapping a class to a XML Schema complex type with a simpleContent or a XML Schema simple type.

Usage:

The @XmlValue annotation can be used with the following program elements:

- a JavaBean property.
- non static, non transient field.

The usage is subject to the following usage constraints:

- At most one field or property can be annotated with the @XmlValue annotation.
- @XmlValue can be used with the following annotations: XmlList. However this is redundant since XmlList maps a type to a simple schema type that derives by list just as XmlValue would.
- If the type of the field or property is a collection type, then the collection item type must map to a simple schema type.
- If the type of the field or property is not a collection type, then the type must map to a XML Schema simple type.

If the annotated JavaBean property is the sole class member being mapped to XML Schema construct, then the class is mapped to a simple type. If there are additional JavaBean properties (other than the JavaBean property annotated with @XmlValue annotation) that are mapped to XML attributes, then the class is mapped to a complex type with simpleContent.

**Example 1:** Map a class to XML Schema simpleType

```java
// Example 1: Code fragment
public class USPrice {
    @XmlValue
    public java.math.BigDecimal price;
}

<!-- Example 1: XML Schema fragment -->
<xs:simpleType name="USPrice">
    <xs:restriction base="xs:decimal"/>
</xs:simpleType>
```

**Example 2:** Map a class to XML Schema complexType with with simpleContent.

```java
// Example 2: Code fragment
```
public class InternationalPrice {
    @XmlValue
    public java.math.BigDecimal price;

    @XmlAttribute
    public String currency;
}

<xs:complexType name="InternationalPrice">
    <xs:simpleContent>
        <xs:extension base="xs:decimal">
            <xs:attribute name="currency" type="xs:string"/>
        </xs:extension>
    </xs:simpleContent>
</xs:complexType>

Since:  
JAXB2.0

Version:  
$Revision: 1.6 $  

Author:  
Sekhar Vajjhala, Sun Microsystems, Inc.

See Also:  
XmlType

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
An Event class describing ConfigBeans being added/subtracted from a server configuration.

### Field Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>static</td>
<td>Object</td>
<td>BEAN_ADDED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Adding a DDBean</td>
</tr>
<tr>
<td>static</td>
<td>Object</td>
<td>BEAN_CHANGED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Changing a DDBean</td>
</tr>
<tr>
<td>static</td>
<td>Object</td>
<td>BEAN_REMOVED</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Removing a DDBean</td>
</tr>
</tbody>
</table>

### Constructor Summary

```java
XpathEvent(DDBean bean, Object typ)
```

A description of a change in the ConfigBean tree.

### Method Summary

<table>
<thead>
<tr>
<th>Type</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDBean</td>
<td>getBean()</td>
<td>The bean being added/removed/changed</td>
</tr>
<tr>
<td></td>
<td>getChangeEvent()</td>
<td></td>
</tr>
</tbody>
</table>
**PropertyChangeEvent**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>boolean isAddEvent()</code></td>
<td>Is this an add event?</td>
</tr>
<tr>
<td><code>boolean isChangeEvent()</code></td>
<td>Is this a change event?</td>
</tr>
<tr>
<td><code>boolean isRemoveEvent()</code></td>
<td>Is this a remove event?</td>
</tr>
<tr>
<td><code>void setChangeEvent(PropertyChangeEvent pce)</code></td>
<td></td>
</tr>
</tbody>
</table>

**Methods inherited from class java.lang.Object**

- `clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

**Field Detail**

**BEAN_ADDED**

public static final `Object` BEAN_ADDED

Adding a DDBean

**BEAN_REMOVED**

public static final `Object` BEAN_REMOVED

Removing a DDBean
public static final Object BEAN_CHANGED

Changing a DDBean

## Constructor Detail

**public XpathEvent(DDBean bean, Object typ)**

**ConfigBean**

```java
public XpathEvent(
    DDBean bean,
    Object typ)
```

A description of a change in the ConfigBean tree.

**Parameters:**
- **bean** - The ConfigBean being added/removed.
- **typ** - Indicates an add/remove event.

## Method Detail

**public java.beans.PropertyChangeEvent getChangeEvent()**

**getChangeEvent**

```java
public PropertyChangeEvent getChangeEvent()
```

---

**public void**
setChangeEvent(PropertyChangeEvent pce)

setChangeEvent
public void setChangeEvent(PropertyChangeEvent pce)

public DDBean getBean()
// Bean
            return // Bean

getBean
public DDBean getBean()

    The bean being added/removed/changed.

    Returns:
    The bean being added/removed/changed.

public boolean isAddEvent()
            return true

isAddEvent
public boolean isAddEvent()

    Is this an add event?

    Returns:
    true if this is an add event.
public boolean isRemoveEvent()

    return true

isRemoveEvent

public boolean isRemoveEvent()

    Is this a remove event?

    Returns:
    true if this is a remove event.

public boolean isChangeEvent()

    return true

isChangeEvent

public boolean isChangeEvent()

    Is this a change event?

    Returns:
    true if this is a change event.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
PS:
public interface XpathListener

XpathEvent

The listener interface for receiving XpathEvents

Author:
gfink

Method Summary

<table>
<thead>
<tr>
<th>Method</th>
<th>Signature</th>
</tr>
</thead>
<tbody>
<tr>
<td>void fireXpathEvent(XpathEvent xpe)</td>
<td></td>
</tr>
</tbody>
</table>

Method Detail

public void fireXpathEvent(XpathEvent xpe)

fireXpathEvent

void fireXpathEvent(XpathEvent xpe)
Java™ Platform Enterprise Edition, v 5.0
API Specifications

### Java EE 5 Platform Packages

<table>
<thead>
<tr>
<th>Package</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.activation</code></td>
<td>The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data.</td>
</tr>
<tr>
<td><code>javax.annotation</code></td>
<td></td>
</tr>
<tr>
<td><code>javax.annotation.security</code></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb</code></td>
<td>The <code>javax.ejb</code> package contains the Enterprise JavaBeans classes and interfaces that define the contracts between the enterprise bean and its clients and between the enterprise bean and the EJB container.</td>
</tr>
<tr>
<td><code>javax.ejb.spi</code></td>
<td>The <code>javax.ejb.spi</code> package defines interfaces that are implemented by the EJB container.</td>
</tr>
<tr>
<td><code>javax.el</code></td>
<td>Provides the API for the <a href="http://docs.oracle.com/javaee/5/api/javax/el/package-summary.html">Unified Expression Language</a> shared by the JSP 2.1 and JSF 1.2 technologies.</td>
</tr>
<tr>
<td><code>javax.enterprise.deploy.model</code></td>
<td>Provides Tool Vendor implementation classes.</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javax.enterprise.deploy.model.exceptions</td>
<td>Provides Tool Vendor exception implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.shared</td>
<td>Provides shared objects for Tool Vendor and Product Vendor implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.shared.factories</td>
<td>Provides shared factory manager object for Tool Vendor and Product Vendor implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.spi</td>
<td>Provides J2EE Product Vendor implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.spi.exceptions</td>
<td>Provides J2EE Product Vendor deployment exception implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.spi.factories</td>
<td>Provides J2EE Product Vendor deployment factory implementation classes.</td>
</tr>
<tr>
<td>javax.enterprise.deploy.spi.status</td>
<td>Provides J2EE Product Vendor deployment status implementation classes.</td>
</tr>
<tr>
<td>javax.faces</td>
<td>Top level classes for the JavaServer(tm) Faces API.</td>
</tr>
<tr>
<td>javax.faces.application</td>
<td>APIs that are used to link an application’s business logic objects to JavaServer Faces, as well as convenient pluggable mechanisms to manage the execution of an application that is based on JavaServer Faces.</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javax.faces.component</td>
<td>Fundamental APIs for user interface components.</td>
</tr>
<tr>
<td>javax.faces.component.html</td>
<td>Specialized user interface component classes for HTML.</td>
</tr>
<tr>
<td>javax.faces.context</td>
<td>Classes and interfaces defining per-request state information.</td>
</tr>
<tr>
<td>javax.faces.convert</td>
<td>Contains classes and interfaces defining converters.</td>
</tr>
<tr>
<td>javax.faces.el</td>
<td>DEPRECATED Classes and interfaces for evaluating and processing reference expressions.</td>
</tr>
<tr>
<td>javax.faces.event</td>
<td>Interfaces describing events and event listeners, and concrete event implementation classes.</td>
</tr>
<tr>
<td>javax.faces.lifecycle</td>
<td>Classes and interfaces defining lifecycle management for the JavaServer Faces implementation.</td>
</tr>
<tr>
<td>javax.faces.model</td>
<td>Standard model data beans for JavaServer Faces.</td>
</tr>
<tr>
<td>javax.faces.render</td>
<td>Classes and interfaces defining the rendering model.</td>
</tr>
<tr>
<td>javax.faces.validator</td>
<td>Interface defining the validator model, and concrete validator implementation classes.</td>
</tr>
<tr>
<td></td>
<td>Classes required for</td>
</tr>
<tr>
<td>javax.faces.webapp</td>
<td>Integration of JavaServer Faces into web applications, including a standard servlet, base classes for JSP custom component tags, and concrete tag implementations for core tags.</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javax.interceptor</td>
<td>The javax.interceptor package contains classes and interfaces for use with EJB interceptors.</td>
</tr>
<tr>
<td>javax.jms</td>
<td>The Java Message Service (JMS) API provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages.</td>
</tr>
<tr>
<td>javax.jws</td>
<td></td>
</tr>
<tr>
<td>javax.jws.soap</td>
<td></td>
</tr>
<tr>
<td>javax.mail</td>
<td>The JavaMail™ API provides classes that model a mail system.</td>
</tr>
<tr>
<td>javax.mail.event</td>
<td>Listeners and events for the JavaMail API.</td>
</tr>
<tr>
<td>javax.mail.internet</td>
<td>Classes specific to Internet mail systems.</td>
</tr>
<tr>
<td>javax.mail.search</td>
<td>Message search terms for the JavaMail API.</td>
</tr>
<tr>
<td>javax.mail.util</td>
<td>Utility classes.</td>
</tr>
<tr>
<td>javax.management.j2ee</td>
<td>Provides the J2EE Management Enterprise Bean component (MEJB)</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javax.management.j2ee.statistics</td>
<td>Provides the standard interfaces for accessing performance data from J2EE managed objects. Package Specification JSR 77, J2EE Management Related Documentation For overviews, tutorials, examples, guides, and tool documentation, please see: J2EE Tools</td>
</tr>
<tr>
<td>javax.persistence</td>
<td>The javax.persistence package contains the classes and interfaces that define the contracts between a persistence provider and the managed classes and the clients of the Java Persistence API.</td>
</tr>
<tr>
<td>javax.persistence.spi</td>
<td>The javax.persistence.spi package defines the classes and interfaces that are implemented by the persistence provider and the Java EE container for use by the container, provider, and/or Persistence bootstrap class in deployment and bootstrapping.</td>
</tr>
<tr>
<td>javax.resource</td>
<td>The javax.resource package is the top-level package for the J2EE Connector API specification.</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>javax.resource.cci</code></td>
<td>The <code>javax.resource.cci</code> package contains API specification for the Common Client Interface (CCI).</td>
</tr>
<tr>
<td><code>javax.resource.spi</code></td>
<td>The <code>javax.resource.spi</code> package contains APIs for the system contracts defined in the J2EE Connector Architecture specification.</td>
</tr>
<tr>
<td><code>javax.resource.spi.endpoint</code></td>
<td>This package contains system contracts for service endpoint interactions.</td>
</tr>
<tr>
<td><code>javax.resource.spi.security</code></td>
<td>The <code>javax.resource.spi.security</code> package contains APIs for the security management contract.</td>
</tr>
<tr>
<td><code>javax.resource.spi.work</code></td>
<td>This package contains APIs for the work management contract.</td>
</tr>
<tr>
<td><code>javax.security.jacc</code></td>
<td>This package contains the Java Authorization Contract for Containers API</td>
</tr>
<tr>
<td><code>javax.servlet</code></td>
<td>The <code>javax.servlet</code> package contains a number of classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for an instance of such a class by a conforming servlet.</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>javax.servlet.http</code></td>
<td>The <code>javax.servlet.http</code> package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp</code></td>
<td>Classes and interfaces for the Core JSP 2.1 API.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.el</code></td>
<td>Provides the <code>ELResolver</code> classes that define the object resolution rules that must be supported by a JSP container with the new unified Expression Language.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.tagext</code></td>
<td>Classes and interfaces for the definition of JavaServer Pages Tag Libraries.</td>
</tr>
<tr>
<td><code>javax.transaction</code></td>
<td>Provides the API that defines the contract between the transaction manager and the various parties involved in a distributed transaction namely: resource manager, application, and application server.</td>
</tr>
<tr>
<td>Package Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>javax.transaction.xa</strong></td>
<td>Between the transaction manager and the resource manager, which allows the transaction manager to enlist and delist resource objects (supplied by the resource manager driver) in JTA transactions.</td>
</tr>
<tr>
<td><strong>javax.xml.bind</strong></td>
<td>Provides a runtime binding framework for client applications including unmarshalling, marshalling, and validation capabilities.</td>
</tr>
<tr>
<td><strong>javax.xml.bind.annotation</strong></td>
<td>Defines annotations for customizing Java program elements to XML Schema mapping.</td>
</tr>
<tr>
<td><strong>javax.xml.bind.annotation.adapters</strong></td>
<td>Defines annotations for customizing Java program elements to XML Schema mapping.</td>
</tr>
<tr>
<td><strong>javax.xml.bind.attachment</strong></td>
<td>This package is implemented by a MIME-based package processor that enables the interpretation and creation of optimized binary data within an MIME-based package format.</td>
</tr>
<tr>
<td><strong>javax.xml.bind.helpers</strong></td>
<td><strong>JAXB Provider Use Only:</strong> Provides partial default implementations for some of the javax.xml.bind interfaces.</td>
</tr>
<tr>
<td><strong>javax.xml.bind.util</strong></td>
<td>Useful client utility classes.</td>
</tr>
<tr>
<td>Package</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>javax.xml.registry</td>
<td>This package and its sub-packages describe the API classes and interfaces for the JAXR API.</td>
</tr>
<tr>
<td>javax.xml.registry.infomodel</td>
<td>This package describes the information model for the JAXR API.</td>
</tr>
<tr>
<td>javax.xml.rpc</td>
<td>This package contains the core JAX-RPC APIs for the client programming model.</td>
</tr>
<tr>
<td>javax.xml.rpc.encoding</td>
<td>This package defines APIs for the extensible type mapping framework.</td>
</tr>
<tr>
<td>javax.xml.rpc.handler</td>
<td>This package defines APIs for SOAP Message Handlers</td>
</tr>
<tr>
<td>javax.xml.rpc.handler.soap</td>
<td>This package defines APIs for SOAP Message Handlers</td>
</tr>
<tr>
<td>javax.xml.rpc.holders</td>
<td>This package contains the standard Java Holder classes.</td>
</tr>
<tr>
<td>javax.xml.rpc.server</td>
<td>This package defines APIs for the servlet based JAX-RPC endpoint model.</td>
</tr>
<tr>
<td>javax.xml.rpc.soap</td>
<td>This package defines APIs specific to the SOAP binding.</td>
</tr>
<tr>
<td>javax.xml.soap</td>
<td>Provides the API for creating and building SOAP messages.</td>
</tr>
<tr>
<td>javax.xml.stream</td>
<td></td>
</tr>
<tr>
<td>javax.xml.stream.events</td>
<td></td>
</tr>
<tr>
<td>javax.xml.stream.util</td>
<td></td>
</tr>
<tr>
<td>javax.xml.ws</td>
<td>This package contains the core JAX-WS APIs.</td>
</tr>
<tr>
<td>-------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>javax.xml.ws.handler</td>
<td>This package defines APIs for message handlers.</td>
</tr>
<tr>
<td>javax.xml.ws.handler.soap</td>
<td>This package defines APIs for SOAP message handlers.</td>
</tr>
<tr>
<td>javax.xml.ws.http</td>
<td>This package defines APIs specific to the HTTP binding.</td>
</tr>
<tr>
<td>javax.xml.ws.soap</td>
<td>This package defines APIs specific to the SOAP binding.</td>
</tr>
<tr>
<td>javax.xml.ws.spi</td>
<td>This package defines SPIs for JAX-WS 2.0.</td>
</tr>
</tbody>
</table>
Package javax.faces.event

Interfaces describing events and event listeners, and concrete event implementation classes.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionListener</td>
<td>A listener interface for receiving ActionEvents.</td>
</tr>
<tr>
<td>FacesListener</td>
<td>A generic base interface for event listeners for various types of FacesEvents.</td>
</tr>
<tr>
<td>PhaseListener</td>
<td>An interface implemented by objects that wish to be notified at the beginning and ending of processing for each standard phase of the request processing lifecycle.</td>
</tr>
<tr>
<td>ValueChangeListener</td>
<td>A listener interface for receiving ValueChangeEvent.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActionEvent</td>
<td>An ActionEvent represents the activation of a user interface component (such as a UICommand).</td>
</tr>
<tr>
<td>FacesEvent</td>
<td>FacesEvent is the base class of interface and application events that can be fired by UIComponents.</td>
</tr>
<tr>
<td>MethodExpressionActionListener</td>
<td>MethodExpressionActionListener is an ActionListener that wraps a MethodExpression.</td>
</tr>
<tr>
<td>MethodExpressionValueChangeListener</td>
<td>MethodExpressionValueChangeListener is a ValueChangeListener that wraps a MethodExpression.</td>
</tr>
</tbody>
</table>
| PhaseEvent               | PhaseEvent represents the beginning or
### PhaseEvent

Ending of processing for a particular phase of the request processing lifecycle, request encapsulated by the `FacesContext`.

### PhaseId

Typesafe enumeration of the legal values that may be returned by the `getPhaseId()` method of the `FacesEvent` interface.

### ValueChangeEvent

A `ValueChangeEvent` is a notification that the local value of the source component has been changed as a result of user interface activity.

## Exception Summary

<table>
<thead>
<tr>
<th>ExceptionSummary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AbortProcessingException</strong></td>
<td>An exception that may be thrown by event listeners to terminate the processing of the current event.</td>
</tr>
</tbody>
</table>
Package javax.faces.event Description

Interfaces describing events and event listeners, and concrete event implementation classes. All events extend from `FacesEvent` and all listeners extend from `FacesListener`.

For your convenience here is a UML class diagram of the classes in this package.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.faces.event

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.faces.event.**FacesEvent**
      - javax.faces.event.**ActionEvent**
      - javax.faces.event.**ValueChangeEvent**
    - javax.faces.event.**PhaseEvent**
  - javax.faces.event.**MethodExpressionActionListener**
    (implements javax.faces.event.**ActionListener**, javax.faces.component.**StateHolder**)
  - javax.faces.event.**MethodExpressionValueChangeListener**
    (implements javax.faces.component.**StateHolder**, javax.faces.event.**ValueChangeListener**)
  - javax.faces.event.**PhaseId** (implements java.lang.**Comparable**<T>)
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - java.lang.**RuntimeException**
        - javax.faces.**FacesException**
          - javax.faces.event.**AbortProcessingException**
Interface Hierarchy

- java.util.**EventListener**
  - javax.faces.event.**FacesListener**
    - javax.faces.event.**ActionListener**
    - javax.faces.event.**ValueChangeListener**
  - javax.faces.event.**PhaseListener** (also extends java.io.**Serializable**)
- java.io.**Serializable**
  - javax.faces.event.**PhaseListener** (also extends java.util.**EventListener**)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
## Deprecated API

### Contents

- [Deprecated Interfaces](#)
- [Deprecated Classes](#)
- [Deprecated Exceptions](#)
- [Deprecated Annotation Types](#)
- [Deprecated Fields](#)
- [Deprecated Methods](#)
- [Deprecated Constructors](#)
- [Deprecated Annotation Type Elements](#)

### Deprecated Interfaces

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.servlet.jsp.el.FunctionMapper</code></td>
<td>As of JSP 2.1, replaced by <code>FunctionMapper</code></td>
</tr>
<tr>
<td><code>javax.resource.spi.security.GenericCredential</code></td>
<td>The preferred way to represent generic credential information is via the <code>org.ietf.jgss.GSSCredential</code> interface in J2SE Version 1.4, which provides similar functionality.</td>
</tr>
<tr>
<td><code>javax.servlet.http.HttpSessionContext</code></td>
<td>As of Java(TM) Servlet API 2.1 for security reasons, with no replacement. This interface will be removed in a future version of this API.</td>
</tr>
<tr>
<td><code>javax.servlet.SingleThreadModel</code></td>
<td>As of Java Servlet API 2.4, with no direct replacement.</td>
</tr>
<tr>
<td><code>javax.xml.bind.Validator</code></td>
<td>since JAXB 2.0</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.el.VariableResolver</code></td>
<td>As of JSP 2.1, replaced by <code>ELResolver</code></td>
</tr>
</tbody>
</table>

### Deprecated Classes
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.faces.webapp.AttributeTag</code></td>
<td>The Faces implementation must now provide the implementation for this class.</td>
</tr>
<tr>
<td><code>javax.faces.webapp.ConverterTag</code></td>
<td>This has been partially replaced by <code>ConverterELTag</code>. The remainder of the functionality, namely, the binding facility and the implementation of the <code>ConverterTag.createConverter()</code> method, is now an implementation detail.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.el.Expression</code></td>
<td>As of JSP 2.1, replaced by <code>ValueExpression</code>.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.el.ExpressionEvaluator</code></td>
<td>As of JSP 2.1, replaced by <code>ExpressionFactory</code>.</td>
</tr>
<tr>
<td><code>javax.servlet.http.HttpUtils</code></td>
<td>As of Java(tm) Servlet API 2.3. These methods were only useful with the default encoding and have been moved to the request interfaces.</td>
</tr>
<tr>
<td><code>javax.faces.el.MethodBinding</code></td>
<td>This has been replaced by <code>MethodExpression</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.PropertyResolver</code></td>
<td>This has been replaced by <code>ELResolver</code>.</td>
</tr>
<tr>
<td><code>javax.xml.soap.SOAPElementFactory</code></td>
<td>- Use <code>javax.xml.soap.SOAPFactory</code> for creating <code>SOAPElements</code>.</td>
</tr>
<tr>
<td><code>javax.faces.application.StateManager.SerializedView</code></td>
<td>This class was not marked <code>Serializable</code> in the 1.0 version of the spec. It was also not a static inner class, so it can't be made to be <code>Serializable</code>. Therefore, it is being deprecated in version 1.2 of the spec. The replacement is to use an implementation dependent <code>Object</code>.</td>
</tr>
<tr>
<td><code>javax.faces.webapp.UIComponentBodyTag</code></td>
<td>All component tags now implement <code>BodyTag</code>. This class has been replaced by <code>UIComponentELTag</code>.</td>
</tr>
<tr>
<td><code>javax.faces.webapp.UIComponentTag</code></td>
<td>Use of this class has been replaced with <code>UIComponentELTag</code>, which extends <code>UIComponentClassicTagBase</code> to add properties that use the EL API introduced as part of JSP 2.1.</td>
</tr>
</tbody>
</table>
| `javax.faces.webapp.ValidatorTag`               | This has been partially replaced by `ValidatorELTag`. The `
The remainder of the functionality, namely, the binding facility and the implementation of the `ValidatorTag.createValidator()` method, is now an implementation detail.

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.faces.el.ValueBinding</code></td>
<td>This has been replaced by <code>ValueExpression</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.VariableResolver</code></td>
<td>This has been replaced by <code>ELResolver</code> when operating with a null base argument.</td>
</tr>
</tbody>
</table>

### Deprecated Exceptions

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.servlet.jsp.el.ELException</code></td>
<td>As of JSP 2.1, replaced by <code>ELException</code>.</td>
</tr>
<tr>
<td><code>javax.servlet.jsp.el.ELParseException</code></td>
<td>As of JSP 2.1, replaced by <code>ELException</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.EvaluationException</code></td>
<td>This has been replaced by <code>ELException</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.MethodNotFoundException</code></td>
<td>This has been replaced by <code>MethodNotFoundException</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.PropertyNotFoundException</code></td>
<td>This has been replaced by <code>PropertyNotFoundException</code>.</td>
</tr>
<tr>
<td><code>javax.faces.el.ReferenceSyntaxException</code></td>
<td>This has been replaced by <code>ELException</code>.</td>
</tr>
</tbody>
</table>

### Deprecated Annotation Types

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.jws.soap.InitParam</code></td>
<td>As of JSR-181 2.0 with no replacement.</td>
</tr>
<tr>
<td><code>javax.jws.soap.SOAPMessageHandler</code></td>
<td>As of JSR-181 2.0 with no replacement.</td>
</tr>
<tr>
<td><code>javax.jws.soap.SOAPMessageHandlers</code></td>
<td>As of JSR-181 2.0 with no replacement.</td>
</tr>
</tbody>
</table>
### Deprecated Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.servlet.jsp.tagext.BodyTag.EVAL_BODY_TAG</td>
<td>As of Java JSP API 1.2, use BodyTag.EVAL_BODY_BUFFERED or IterationTag.EVAL_BODY_AGAIN.</td>
</tr>
<tr>
<td>javax.faces.validator.Validator.NOT_IN_RANGE_MESSAGE_ID</td>
<td>Use DoubleRangeValidator.NOT_IN_RANGE_MESSAGE_ID or LongRangeValidator.NOT_IN_RANGE_MESSAGE_ID instead.</td>
</tr>
</tbody>
</table>

### Deprecated Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Replacement</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.soap.SOAPElementFactory.create(Name)</td>
<td>Use javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name)</td>
</tr>
<tr>
<td>javax.xml.soap.SOAPElementFactory.create(String)</td>
<td>Use javax.xml.soap.SOAPFactory.createElement(String localName)</td>
</tr>
<tr>
<td>javax.xml.soap.SOAPElementFactory.create(String, String, String)</td>
<td>Use javax.xml.soap.SOAPFactory.createElement(String localName)</td>
</tr>
<tr>
<td>javax.faces.application.Application.createComponent(ValueBinding, FacesContext, String)</td>
<td>This has been replaced by Application.createComponent(javax.el.ValueExpression, javax.faces.context.FacesContext, java.lang.String)</td>
</tr>
<tr>
<td>javax.faces.application.Application.createMethodBinding(String, Class[])</td>
<td>This has been replaced by calling Application&lt;ExpressionFactory.createMethodExpression(javax.el.ELContext, java.lang.String, java.lang.Class[]).</td>
</tr>
<tr>
<td>javax.xml.bind.JAXBContext.createValidator()</td>
<td>since JAXB2.0</td>
</tr>
<tr>
<td>javax.faces.application.Application.createComponent(ValueExpression)</td>
<td>This has been replaced by calling Application&lt;ExpressionFactory.createValueExpression(javax.el.ELContext, java.lang.String, java.lang.Class[]).</td>
</tr>
<tr>
<td>javax.enterprise.deploy.spi.DeploymentManager.distribute(Target[], InputStream, InputStream)</td>
<td>as of Java EE 5, replaced with DeploymentManager.distribute(Target[], ModuleType, InputStream, InputStream)</td>
</tr>
<tr>
<td>javax.faces.webapp.UIComponentClassicTagBase.encodeBegin()</td>
<td>No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.</td>
</tr>
</tbody>
</table>
javax.faces.webapp.UIComponentClassicTagBase.encodeChildren()
No encoding is done during JSP page execution. Encoding is deferred executing to allow the entire tree to be built before any encoding occurs.

javax.faces.webapp.UIComponentClassicTagBase.encodeEnd()
No encoding is done during JSP page execution. Encoding is deferred executing to allow the entire tree to be built before any encoding occurs.

javax.servlet.http.HttpServletResponse.encodeRedirectUrl(String)
As of version 2.1, use encodeRedirectURL(String url) instead

javax.servlet.http.HttpServletResponse.encodeUrl(String)
As of version 2.1, use encodeURL(String url) instead

javax.faces.component.ActionSource.getAction()
This has been replaced by ActionSource2.getActionExpression().

javax.faces.component.UICommand.getAction()
This has been replaced by UICommand.getActionExpression().

javax.faces.component.ActionSource.getActionListener()
Use ActionSource.getActionListeners() instead.

javax.faces.component.UICommand.getActionListener()
Use UICommand.getActionListeners() instead.

javax.ejb.EJBContext.getCallerIdentity()
Use Principal.getCallerPrincipal() instead.

javax.faces.render.ResponseStateManager.getComponentStateToRestore(FacesContext)
This method has been replaced by ResponseStateManager.getState(javax.faces.context.FacesContext, java.lang.String). The default implementation returns null.

javax.faces.application.StateManager.getComponentStateToSave(FacesContext)
the distinction between tree structure and component state is now an implementation detail returns null.

javax.ejb.EJBContext.getEnvironment()
Use the JNDI naming context java:comp/env to access enterprise bean's environment.

javax.xml.bind.Validator.getEventHandler()
since JAXB2.0

javax.servlet.jsp.JspContext.getExpressionEvaluator()
As of JSP 2.1, replaced by JspApplicationContext.getExpressionFactory()

javax.servlet.http.HttpSessionContext.getIds()
As of Java Servlet API 2.1 with no replacement. This method must be removed in a future version of this API.
**javax.resource.ResourceException.getLinkedException()**

J2SE release 1.4 supports a chained exception facility that allows a throwable, if any, that caused it to get thrown. Refer to `getCause` and `initCause` in `java.lang.Throwable`.

**javax.resource.cci.ResourceWarning.getLinkedWarning()**

J2SE release 1.4 supports a chained exception facility that allows a throwable, if any, that caused it to get thrown. Refer to `getCause` and `initCause` in `java.lang.Throwable`.

**javax.enterprise.deploy.model.DDBeanRoot.getModuleDTDVersion()**

As of version 1.1 replaced by `DDBeanRoot.getDDBeanRootVersion()`.

**javax.enterprise.deploy.model.DeployableObject.getModuleDTDVersion()**

As of version 1.1 replaced by `DDBeanRoot.getDDBeanRootVersion()`.

**javax.xml.bind.Validator.getProperty(String)**

Since JAXB2.0

**javax.faces.application.Application.getPropertyResolver()**

This has been replaced by `Application.getELResolver()`.

**javax.servlet.ServletRequest.getRealPath(String)**

As of Version 2.1 of the Java Servlet API, use `ServletContext.getRealPath()`.

**javax.servlet.jsp.JspException.getRootCause()**

As of JSP 2.1, replaced by `Throwable.getCause()`.

**javax.servlet.UnavailableException.getServlet()**

As of Java Servlet API 2.2, with no replacement. Returns the servlet that is reporting its unavailability.

**javax.servlet.ServletContext.getServlet(String)**

As of Java Servlet API 2.1, with no direct replacement. This method was originally defined to retrieve a servlet from a `ServletContext`. It returns `null` and remains only to preserve binary compatibility. This method will be permanently removed in future version of the Java Servlet API.

In lieu of this method, servlets can share information using the `ServletContext` business logic by invoking methods on common non-servlet classes.

**javax.servlet.ServletContext.getServletNames()**

As of Java Servlet API 2.1, with no replacement. This method was originally defined to return an `Enumeration` of all the servlet names known to this context. In this version, this method always returns an empty `Enumeration` and remains only to preserve binary compatibility.
method will be permanently removed in a future version of the Java Servlet API.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.servlet.ServletContext.getServlets()</td>
<td>As of Java Servlet API 2.0, with no replacement. This method was originally defined to return an <code>Enumeration</code> of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.</td>
</tr>
<tr>
<td>javax.servlet.http.HttpSessionContext.getSession(String)</td>
<td>As of Java Servlet API 2.1 with no replacement. This method must be replaced in a future version of this API.</td>
</tr>
<tr>
<td>javax.servlet.http.HttpSession.getSessionContext()</td>
<td>As of Version 2.1, this method is deprecated and has no replacement in the Java Servlet API.</td>
</tr>
<tr>
<td>javax.faces.render.ResponseStateManager.getTreeStructureToRestore(FacesContext, String)</td>
<td>This method has been replaced by <code>ResponseStateManager.getState(javax.faces.context.FacesContext, java.lang.String)</code>. The default implementation returns <code>null</code>.</td>
</tr>
<tr>
<td>javax.faces.application.StateManager.getTreeStructureToSave(FacesContext)</td>
<td>The distinction between tree structure and component state is now an implementation detail. The default implementation returns <code>null</code>.</td>
</tr>
<tr>
<td>javax.faces.component.EditableValueHolder.getValidator()</td>
<td><code>EditableValueHolder.getValidators()</code> should be used instead.</td>
</tr>
<tr>
<td>javax.faces.component.UIInput.getValidator()</td>
<td><code>UIInput.getValidators()</code> should be used instead.</td>
</tr>
<tr>
<td>javax.servlet.http.HttpSession.getValue(String)</td>
<td>As of Version 2.2, this method is replaced by <code>HttpSession.getAttribute(java.lang.String)</code>.</td>
</tr>
<tr>
<td>javax.faces.component.UIComponent.getValueBinding(String)</td>
<td>This has been replaced by <code>UIComponent.getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td>javax.faces.component.UIComponentBase.getValueBinding(String)</td>
<td>This has been replaced by <code>UIComponent.getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td>javax.faces.component.UIGraphic.getValueBinding(String)</td>
<td>This has been replaced by <code>UIGraphic.getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td>javax.faces.component.UISelectBoolean.getValueBinding(String)</td>
<td>This has been replaced by <code>UISelectBoolean.getValueExpression(java.lang.String)</code></td>
</tr>
<tr>
<td>javax.faces.component.UISelectMany.getValueBinding(String)</td>
<td>This has been replaced by <code>UISelectMany.getValueExpression(java.lang.String)</code></td>
</tr>
</tbody>
</table>
this has been replaced by `UISelectMany.getValueExpression(java.lang.String)`

`javax.faces.component.EditableValueHolder.getValueChangeListener()`  
Use `EditableValueHolder.getValueChangeListeners()` instead.

`javax.servlet.http.HttpSession.getValueNames()`  
As of Version 2.2, this method is replaced by `HttpSession.getAttributeNames()`.

`javax.faces.application.Application.getVariableResolver()`  
This has been replaced by `Application.getELResolver()`.

`javax.servlet.jsp.JspContext.getVariableResolver()`  
As of JSP 2.1, replaced by `ELContext.getELResolver()`, which can be obtained by `jspContext.getELContext().getELResolver()`.

`javax.ejb.EJBContext.isCallerInRole(Identity)`  
Use `boolean isCallerInRole(String roleName)` instead.

`javax.servlet.http.HttpServletRequest.isRequestedSessionIdFromUrl()`  
As of Version 2.1 of the Java Servlet API, use `HttpServletRequest.isRequestedSessionIdFromURL()` instead.

`javax.xml.bind.Unmarshaller.isValidating()`  
since JAXB2.0, please see `Unmarshaller.getSchema()`.

`javax.servlet.ServletContext.log(Exception, String)`  
As of Java Servlet API 2.1, use `ServletContext.log(String message, Throwable throwable)`.

This method was originally defined to write an exception’s stack trace and explanatory error message to the servlet log file.

`javax.activation.ActivationDataFlavor.normalizeMimeType(String)`

`javax.activation.ActivationDataFlavor.normalizeMimeTypeParameter(String, String)`

`javax.servlet.http.HttpSession.putValue(String, Object)`  
As of Version 2.2, this method is replaced by `HttpSession.setAttribute(java.lang.String, java.lang.Object)`.

`javax.servlet.http.HttpSession.removeValue(String)`  
As of Version 2.2, this method is replaced by `HttpSession.removeAttribute(java.lang.String)`.

`javax.faces.application.StateManager.restoreComponentState(FacesContext, UIViewRoot, String)`  
The distinction between tree structure and component state is now an implementation detail.
the distinction between tree structure and component state is now a
implementation returns null.

this has been replaced by StateManager.saveView(javax.faces.con
implementation returns null.

This has been replaced by ActionSource2.setActionExpression(javax.ax

This has been replaced by UICommand.setActionExpression(javax.faces.e

This has been replaced by UICommand.addActionListener(javax.faces.fac

since JAXB2.0

since JAXB2.0

J2SE release 1.4 supports a chained exception facility that allows a throwables, if any, that caused it to get thrown. Refer to getCause and ini
java.lang.Throwable class.

J2SE release 1.4 supports a chained exception facility that allows a throwables, if any, that caused it to get thrown. Refer to getCause and ini
java.lang.Throwable class.

since JAXB2.0

The recommended way to affect the execution of the EL is to provide a place in the application configuration resources which will be considered in eval
This method now will cause the argument resolver to be wrapped in an ELResolv
and exposed to the EL resolution system as if the user had called Application.addELResolv

As of version 2.1, due to ambiguous meaning of the message parameter, setStatus(int), to send an error with a description use sendError(int, S
javax.xml.bind.Unmarshaller.setValidating(boolean)
since JAXB2.0, please see Unmarshaller.setSchema(javax.xml.validation.Schema)

javax.faces.component.EditableValueHolder.setValidator(MethodBinding)
Use EditableValueHolder.addValidator(javax.faces.validator.Validator) by creating an instance of MethodExpressionValidator

javax.faces.component.UIInput.setValidator(MethodBinding)
Use UIInput.addValidator(javax.faces.validator.Validator) instead, obtaining the argument by creating an instance of MethodExpressionValidator.

javax.faces.component.UIComponent.setValueBinding(String, ValueBinding)
This has been replaced by UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.UIComponentBase.setValueBinding(String, ValueBinding)
This has been replaced by UIComponent.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.UIData.setValueBinding(String, ValueBinding)
This has been replaced by UIData.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.UIGraphic.setValueBinding(String, ValueBinding)
This has been replaced by UIGraphic.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.UISelectBoolean.setValueBinding(String, ValueBinding)
This has been replaced by UISelectBoolean.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.UISelectMany.setValueBinding(String, ValueBinding)
This has been replaced by UISelectMany.setValueExpression(java.lang.String, javax.el.ValueExpression).

javax.faces.component.EditableValueHolder.setValueChangeListener(MethodBinding)
Use EditableValueHolder.addValueChangeListener(javax.faces.event.ValueChangeListener) by creating an instance of MethodExpressionValueChangeListener.

javax.faces.component.UIInput.setValueChangeListener(MethodBinding)
Use UIInput.addValueChangeListener(javax.faces.event.ValueChangeListener) argument ValueChangeListener by creating an instance of MethodExpressionValueChangeListener.

javax.faces.application.Application.setVariableResolver(VariableResolver)
The recommended way to affect the execution of the EL is to provide place in the application configuration resources which will be considered in normal course of expression evaluation. This method now will cause the argument resolver to be wrapped inside an implementation of ELResolver and exposed to the EL resolution system as if the user had called Application.addELResolver(javax.el.ELResolver).
<table>
<thead>
<tr>
<th>Method</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.bind.Validator.validate(Object)</td>
<td>JAXB2.0</td>
</tr>
<tr>
<td>javax.xml.bind.Validator.validateRoot(Object)</td>
<td>JAXB2.0</td>
</tr>
<tr>
<td>javax.faces.application.StateManager.writeState(FacesContext, StateManager.SerializedView)</td>
<td></td>
</tr>
<tr>
<td>This method has been replaced by StateManager.writeValue(FacesContext, java.lang.object)</td>
<td></td>
</tr>
<tr>
<td>this method does nothing.</td>
<td></td>
</tr>
<tr>
<td>javax.faces.render.ResponseStateManager.writeState(FacesContext, StateManager.SerializedView)</td>
<td></td>
</tr>
<tr>
<td>This method has been replaced by ResponseStateManager.writeValue(FacesContext, java.lang.object)</td>
<td></td>
</tr>
<tr>
<td>implementation of this method does nothing.</td>
<td></td>
</tr>
</tbody>
</table>

### Deprecated Constructors

<table>
<thead>
<tr>
<th>Method</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.servletUnavailableException(int, Servlet, String)</td>
<td></td>
</tr>
<tr>
<td>As of Java Servlet API 2.2, use UnavailableException.UnavailableException(String, int) instead.</td>
<td></td>
</tr>
<tr>
<td>javax.servletUnavailableException(Servlet, String)</td>
<td></td>
</tr>
<tr>
<td>As of Java Servlet API 2.2, use UnavailableException.UnavailableException(String) instead.</td>
<td></td>
</tr>
</tbody>
</table>

### Deprecated Annotation Type Elements

<table>
<thead>
<tr>
<th>Method</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.jws.HandlerChain.name</td>
<td></td>
</tr>
<tr>
<td>As of JSR-181 2.0 with no replacement.</td>
<td></td>
</tr>
</tbody>
</table>
AbortProcessingException - Exception in javax.faces.event
An exception that may be thrown by event listeners to terminate the processing of the current event.

AbortProcessingException() - Constructor for exception javax.faces.event.AbortProcessingException
Construct a new exception with no detail message or root cause.

AbortProcessingException(String) - Constructor for exception javax.faces.event.AbortProcessingException
Construct a new exception with the specified detail message and no root cause.

AbortProcessingException(Throwable) - Constructor for exception javax.faces.event.AbortProcessingException
Construct a new exception with the specified root cause.

AbortProcessingException(String, Throwable) - Constructor for exception javax.faces.event.AbortProcessingException
Construct a new exception with the specified detail message and root cause.

AbstractMarshallerImpl - Class in javax.xml.bind.helpers
Partial default Marshaller implementation.

AbstractMarshallerImpl() - Constructor for class javax.xml.bind.helpers.AbstractMarshallerImpl

AbstractUnmarshallerImpl - Class in javax.xml.bind.helpers
Partial defaultUnmarshaller implementation.

AbstractUnmarshallerImpl() - Constructor for class javax.xml.bind.helpers.AbstractUnmarshallerImpl

accept(XMLEvent) - Method in interface javax.xml.stream.EventFilter
Tests whether this event is part of this stream.

accept(XMLStreamReader) - Method in interface javax.xml.stream.StreamFilter
Tests whether the current state is part of this stream.

AccessLocalException - Exception in javax.ejb
An AccessLocalException is thrown to indicate that the caller does not have permission to call the method.
**AccessException()** - Constructor for exception
javax.ejbAccessException

Constructs an AccessLocalException with no detail message.

**AccessException(String)** - Constructor for exception
javax.ejbAccessException

Constructs an AccessLocalException with the specified detail message.

**AccessException(String, Exception)** - Constructor for exception
javax.ejbAccessException

Constructs an AccessLocalException with the specified detail message and a nested exception.

**acknowledge()** - Method in interface javax.jms.Message

Acknowledges all consumed messages of the session of this consumed message.

**ActionEvent** - Class in javax.faces.event

An ActionEvent represents the activation of a user interface component (such as a UICommand).

**ActionEvent(UIComponent)** - Constructor for class
javax.faces.event.ActionEvent

Construct a new event object from the specified source component and action command.

**ActionListener** - Interface in javax.faces.event

A listener interface for receiving ActionEvents.

**ActionSource** - Interface in javax.faces.component

ActionSource is an interface that may be implemented by any concrete UIComponent that wishes to be a source of ActionEvents, including the ability to invoke application actions via the default ActionListener mechanism.

**ActionSource2** - Interface in javax.faces.component

ActionSource2 extends ActionSource and provides a JavaBeans property analogous to the "action" property on ActionSource.

**ActionType** - Class in javax.enterprise.deploy.shared

Class ActionTypes defines enumeration values for the J2EE DeploymentStatus actions.

**ActionType(int)** - Constructor for class
javax.enterprise.deploy.shared.ActionType

Construct a new enumeration value with the given integer value.

**ActivationConfigProperty** - Annotation Type in javax.ejb
**ActivationDataFlavor** - Class in `javax.activation`
   The ActivationDataFlavor class is a special subclass of `java.awt.datatransfer.DataFlavor`.

**ActivationDataFlavor(Class, String, String)** - Constructor for class `javax.activation.ActivationDataFlavor`
   Construct a DataFlavor that represents an arbitrary Java object.

**ActivationDataFlavor(Class, String)** - Constructor for class `javax.activation.ActivationDataFlavor`
   Construct a DataFlavor that represents a MimeType.

**ActivationDataFlavor(String, String)** - Constructor for class `javax.activation.ActivationDataFlavor`
   Construct a DataFlavor that represents a MimeType.

**ActivationSpec** - Interface in `javax.resource.spi`
   This interface serves as a marker.

**add(ELResolver)** - Method in class `javax.el.CompositeELResolver`
   Adds the given resolver to the list of component resolvers.

**add(FetchProfile.Item)** - Method in class `javax.mail.FetchProfile`
   Add the given special item as one of the attributes to be prefetched.

**add(String)** - Method in class `javax.mail.FetchProfile`
   Add the specified header-field to the list of attributes to be prefetched.

**add(Flags.Flag)** - Method in class `javax.mail.Flags`
   Add the specified system flag to this Flags object.

**add(String)** - Method in class `javax.mail.Flags`
   Add the specified user flag to this Flags object.

**add(Flags)** - Method in class `javax.mail.Flags`
   Add all the flags in the given Flags object to this Flags object.

**add(XMLEvent)** - Method in interface `javax.xml.stream.util.XMLEventConsumer`
   This method adds an event to the consumer.

**add(XMLEvent)** - Method in interface `javax.xml.stream.XMLEventWriter`
   Add an event to the output stream Adding a START_ELEMENT will open a new namespace scope that will be closed when the corresponding END_ELEMENT is written.

**add(XMLEventReader)** - Method in interface `javax.xml.stream.XMLEventWriter`
   Adds an entire stream to an output stream, calls next() on the inputStream argument until hasNext() returns false This should be treated as a convenience method that will perform the following loop
over all the events in an event reader and call add on each event.

**addActionListener(ActionListener)** - Method in interface
javax.faces.component.ActionSource

Add a new ActionListener to the set of listeners interested in being notified when ActionEvents occur.

**addActionListener(ActionListener)** - Method in class
javax.faces.component.UIComponent

**addAssociation(Association)** - Method in interface
javax.xml.registry.infomodel.RegistryObject

Adds specified Association to use this object as source.

**addAssociations(Collection)** - Method in interface
javax.xml.registry.infomodel.RegistryObject

Adds specified Associations to use this object as source.

**addAttachmentPart(AttachmentPart)** - Method in class
javax.xml.soap.SOAPMessage

Adds the given AttachmentPart object to this SOAPMessage object.

**addAttribute(Name, String)** - Method in interface
javax.xml.soap.SOAPElement

Adds an attribute with the specified name and value to this SOAPElement object.

**addAttribute(QName, String)** - Method in interface
javax.xml.soap.SOAPElement

Adds an attribute with the specified name and value to this SOAPElement object.

**addBody()** - Method in interface
javax.xml.soap.SOAPEnvelope

Creates a SOAPBody object and sets it as the SOAPBody object for this SOAPEnvelope object.

**addBodyElement(Name)** - Method in interface
javax.xml.soap.SOAPBody

Creates a new SOAPBodyElement object with the specified name and adds it to this SOAPBody object.

**addBodyElement(QName)** - Method in interface
javax.xml.soap.SOAPBody

Creates a new SOAPBodyElement object with the specified QName and adds it to this SOAPBody object.

**addBodyPart(BodyPart)** - Method in class
javax.mail.internet.MimeMultipart

Adds a Part to the multipart.
**addBodyPart(BodyPart, int)** - Method in class javax.mail.internet.MimeMultipart

  Adds a BodyPart at position index.

**addBodyPart(BodyPart)** - Method in class javax.mail.Multipart

  Adds a Part to the multipart.

**addBodyPart(BodyPart, int)** - Method in class javax.mail.Multipart

  Adds a BodyPart at position index.

**addChild(UIComponent)** - Method in class javax.faces.webapp.UIComponentClassicTagBase

  Adds a child UIComponent.

**addChild(UIComponent)** - Method in class javax.faces.webapp.UIComponentTagBase

  Add the component identifier of the specified UIComponent to the list of component identifiers created or located by nested UIComponentTags processing this request.

**addChildConcept(Concept)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

  Adds a child Concept.

**addChildConcept(Concept)** - Method in interface javax.xml.registry.infomodel.Concept

  Adds a child Concept.

**addChildConcepts(Collection)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

  Adds a Collection of Concept children.

**addChildConcepts(Collection)** - Method in interface javax.xml.registry.infomodel.Concept

  Adds a Collection of Concept children.

**addChildElement(Name)** - Method in interface javax.xml.soap.SOAPElement

  Creates a new SOAPElement object initialized with the given Name object and adds the new element to this SOAPElement object.

**addChildElement(QName)** - Method in interface javax.xml.soap.SOAPElement

  Creates a new SOAPElement object initialized with the given QName object and adds the new element to this SOAPElement object.

**addChildElement(String)** - Method in interface javax.xml.soap.SOAPElement

  Creates a new SOAPElement object initialized with the specified local name and adds the new element to this SOAPElement object.
**addChildElement(String, String)** - Method in interface `javax.xml.soap.SOAPElement`  
Creates a new `SOAPElement` object initialized with the specified local name and prefix and adds the new element to this `SOAPElement` object.

**addChildElement(String, String, String)** - Method in interface `javax.xml.soap.SOAPElement`  
Creates a new `SOAPElement` object initialized with the specified local name, prefix, and URI and adds the new element to this `SOAPElement` object.

**addChildElement(SOAPElement)** - Method in interface `javax.xml.soap.SOAPElement`  
Add a `SOAPElement` as a child of this `SOAPElement` instance.

**addChildOrganization(Organization)** - Method in interface `javax.xml.registry.infomodel.Organization`  
Adds a child Organization.

**addChildOrganizations(Collection)** - Method in interface `javax.xml.registry.infomodel.Organization`  
Adds a Collection of Organization children.

**addClassification(Classification)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`  
Adds specified Classification to this object.

**addClassifications(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`  
Adds specified Classifications to this object.

**addComponent(String, String)** - Method in class `javax.faces.application.Application`  
Register a new mapping of component type to the name of the corresponding `UIComponent` class.

**addConnectionEventListener(ConnectionEventListener)** - Method in interface `javax.resource.spi.ManagedConnection`  
Adds a connection event listener to the ManagedConnection instance.

**addConnectionListener(ConnectionListener)** - Method in class `javax.mail.Folder`  
Add a listener for Connection events on this Folder.

**addConnectionListener(ConnectionListener)** - Method in class `javax.mail.Service`  
Add a listener for Connection events on this service.
**addConverter(String, String)** - Method in class
javax.faces.application.Application

Register a new mapping of converter id to the name of the corresponding Converter class.

**addConverter(Class, String)** - Method in class
javax.faces.application.Application

Register a new converter class that is capable of performing conversions for the specified target class.

**addCookie(Cookie)** - Method in interface
javax.servlet.http.HttpServletResponse

Adds the specified cookie to the response.

**addCookie(Cookie)** - Method in class
javax.servlet.http.HttpServletResponseWrapper

The default behavior of this method is to call addCookie(Cookie cookie) on the wrapped response object.

**addDataModelListener(DataModelListener)** - Method in class
javax.faces.model.DataModel

Add a new DataModelListener to the set interested in notifications from this DataModel.

**addDateHeader(String, long)** - Method in interface
javax.servlet.http.HttpServletResponse

Adds a response header with the given name and date-value.

**addDateHeader(String, long)** - Method in class
javax.servlet.http.HttpServletResponseWrapper

The default behavior of this method is to call addDateHeader(String name, long date) on the wrapped response object.

**addDetail()** - Method in interface javax.xml.soap.SOAPFault

Creates an optional Detail object and sets it as the Detail object for this SOAPFault object.

**addDetailEntry(Name)** - Method in interface javax.xml.soap.Detail

Creates a new DetailEntry object with the given name and adds it to this Detail object.

**addDetailEntry(QName)** - Method in interface javax.xml.soap.Detail

Creates a new DetailEntry object with the given QName and adds it to this Detail object.

**addDocument(Document)** - Method in interface
javax.xml.soap.SOAPBody

Adds the root node of the DOM Document to this SOAPBody object.

**ADDED** - Static variable in class javax.mail.event.MessageCountEvent
The messages were added to their folder.

**addELContextListener(ELContextListener)** - Method in class `javax.faces.application.Application`

Provide a way for Faces applications to register an `ELContextListener` that will be notified on creation of `ELContext` instances.


Registers a `ELContextListener` so that context objects can be added whenever a new `ELContext` is created.

**addELResolver(ELResolver)** - Method in class `javax.faces.application.Application`

Cause an the argument `resolver` to be added to the resolver chain as specified in section 5.5.1 of the JavaServer Faces Specification.

**addELResolver(ELResolver)** - Method in interface `javax.servlet.jsp.JspApplicationContext`

Adds an `ELResolver` to affect the way EL variables and properties are resolved for EL expressions appearing in JSP pages and tag files.

**addExternalIdentifier(ExternalIdentifier)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`

Adds specified `ExternalIdentifier` as an external identifier to this object.

**addExternalIdentifiers(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`

Adds specified `ExternalIdentifiers` as an external identifiers to this object.

**addExternalLink(ExternalLink)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`

Adds specified `ExternalLink` to this object.

**addExternalLinks(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`

Adds specified `ExternalLinks` to this object.

**addFacesListener(FacesListener)** - Method in class `javax.faces.component.UIComponent`

Add the specified `FacesListener` to the set of listeners registered to receive event notifications from this `UIComponent`.

**addFacesListener(FacesListener)** - Method in class `javax.faces.component.UIComponentBase`

Add the specified `FacesListener` to the set of listeners registered to
receive event notifications from this `UIComponent`

**addFacet(String)** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**

**addFacet(String)** - Method in class
javax.faces.webapp.**UIComponentTagBase**
Add the facet name of the specified facet to the list of facet names created or located by nested **UIComponentTag**s processing this request.

**addFault()** - Method in interface javax.xml.soap.**SOAPBody**
Creates a new SOAPFault object and adds it to this SOAPBody object.

**addFault(Name, String, Locale)** - Method in interface
javax.xml.soap.**SOAPBody**
Creates a new SOAPFault object and adds it to this SOAPBody object.

**addFault(QName, String, Locale)** - Method in interface
javax.xml.soap.**SOAPBody**
Creates a new SOAPFault object and adds it to this SOAPBody object.

**addFault(Name, String)** - Method in interface
javax.xml.soap.**SOAPBody**
Creates a new SOAPFault object and adds it to this SOAPBody object.

**addFault(QName, String)** - Method in interface
javax.xml.soap.**SOAPBody**
Creates a new SOAPFault object and adds it to this SOAPBody object.

**addFaultReasonText(String, Locale)** - Method in interface
javax.xml.soap.**SOAPFault**
Appends or replaces a Reason Text item containing the specified text message and an `xml:lang` derived from locale.

**addFolderListener(FolderListener)** - Method in class javax.mail.**Folder**
Add a listener for Folder events on this Folder.

**addFolderListener(FolderListener)** - Method in class javax.mail.**Store**
Add a listener for Folder events on any Folder object obtained from this Store.

**addFrom(Address[])** - Method in class
javax.mail.internet.**MimeMessage**
Add the specified addresses to the existing "From" field.

**addFrom(Address[])** - Method in class javax.mail.**Message**
Add these addresses to the existing "From" attribute

**addHeader(String, String)** - Method in class
javax.mail.internet.**InternetHeaders**
Add a header with the specified name and value to the header list.

**addHeader(String, String)** - Method in class javax.mail.internet.MimeBodyPart

Add this value to the existing values for this header_name.

**addHeader(String, String)** - Method in class javax.mail.internet.MimeMessage

Add this value to the existing values for this header_name.

**addHeader(String, String)** - Method in interface javax.mail.Part

Add this value to the existing values for this header_name.

**addHeader(String, String)** - Method in interface javax.servlet.http.HttpServletResponse

Adds a response header with the given name and value.

**addHeader(String, String)** - Method in class javax.servlet.http.HttpServletResponseWrapper

The default behavior of this method is to return addHeader(String name, String value) on the wrapped response object.

**addHeader(String, String)** - Method in class javax.xml.soap.MimeHeaders

Adds a MimeHeader object with the specified name and value to this MimeHeaders object's list of headers.

**addHeader()** - Method in interface javax.xml.soap.SOAPEnvelope

Creates a SOAPHeader object and sets it as the SOAPHeader object for this SOAPEnvelope object.

**addHeaderElement(Name)** - Method in interface javax.xml.soap.SOAPHeader

Creates a new SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object.

**addHeaderElement(QName)** - Method in interface javax.xml.soap.SOAPHeader

Creates a new SOAPHeaderElement object initialized with the specified qname and adds it to this SOAPHeader object.

**addHeaderLine(String)** - Method in class javax.mail.internet.InternetHeaders

Add an RFC822 header line to the header store.

**addHeaderLine(String)** - Method in class javax.mail.internet.MimeBodyPart

Add a header line to this body part.

**addHeaderLine(String)** - Method in class javax.mail.internet.MimeMessage
Add a raw RFC 822 header-line.

**addHeaderLine(String)** - Method in interface javax.mail.internet.MimePart

Add a raw RFC822 header-line.

**addIntHeader(String, int)** - Method in interface javax.servlet.http.HttpServletResponse

Adds a response header with the given name and integer value.

**addIntHeader(String, int)** - Method in class javax.servlet.http.HttpServletResponseWrapper

The default behavior of this method is to call addIntHeader(String name, int value) on the wrapped response object.

**addLifecycle(String, Lifecycle)** - Method in class javax.faces.lifecycle.LifecycleFactory

Register a new Lifecycle instance, associated with the specified lifecycleId, to be supported by this LifecycleFactory.

**addLocalizedString(LocalizedString)** - Method in interface javax.xml.registry.infomodel.InternationalString

Adds a LocalizedString to this object.

**addLocalizedString(String)** - Method in interface javax.xml.registry.infomodel.InternationalString

Adds many LocalizedStrings to this object.

**addMailcap(String)** - Method in class javax.activation.MailcapCommandMap

Add entries to the registry.

**addMessage(String, FacesMessage)** - Method in class javax.faces.context.FacesContext

Append a FacesMessage to the set of messages associated with the specified client identifier, if clientId is not null.

**addMessageChangedListener(MessageChangedListener)** - Method in class javax.mail.Folder

Add a listener for MessageChanged events on this Folder.

**addMessageCountListener(MessageCountListener)** - Method in class javax.mail.Folder

Add a listener for MessageCount events on this Folder.

**addMimeHeader(String, String)** - Method in class javax.xml.soap.AttachmentPart

Adds a MIME header with the specified name and value to this AttachmentPart object.

**addMimeHeader(String, String)** - Method in class
javax.xml.soap.SOAPPart
   Creates a MimeHeader object with the specified name and value and adds it to this SOAPPart object.

addMimeTypes(String) - Method in class java.activation.MimetypesFileTypeMap
   Prepends the MIME type values to the registry.

addMtomAttachment(DataHandler, String, String) - Method in class javax.xml.bind.attachment.AttachmentMarshaller
   Consider MIME content data for optimized binary storage as an attachment.

addMtomAttachment(byte[], int, int, String, String, String) - Method in class javax.xml.bind.attachment.AttachmentMarshaller
   Consider binary data for optimized binary storage as an attachment.

addNamespaceDeclaration(String, String) - Method in interface javax.xml.soap.SOAPElement
   Adds a namespace declaration with the specified prefix and URI to this SOAPElement object.

addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object) - Method in interface javax.management.j2ee.ListenerRegistration
   Add a listener to a registered managed object.

addNotUnderstoodHeaderElement(QName) - Method in interface javax.xml.soap.SOAPHeader
   Creates a new NotUnderstood SOAPHeaderElement object initialized with the specified name and adds it to this SOAPHeader object.

addParameter(String, QName, ParameterMode) - Method in interface javax.xml.rpc.Call
   Adds a parameter type and mode for a specific operation.

addParameter(String, QName, Class, ParameterMode) - Method in interface javax.xml.rpc.Call
   Adds a parameter type and mode for a specific operation.

addPhaseListener(PhaseListener) - Method in class javax.faces.component.UIViewRoot
   Register a new PhaseListener instance that is interested in being notified before and after the processing for standard phases of the request processing lifecycle.
addPort(QName, String, String) - Method in class javax.xml.ws.Service
   Creates a new port for the service.
addPort(QName, String, String) - Method in class javax.xml.ws.spi.ServiceDelegate
   Creates a new port for the service.
addProgressListener(ProgressListener) - Method in interface javax.enterprise.deploy.spi.status.ProgressObject
   Add a listener to receive Progress events on deployment actions.
addProgressListener(PropertyChangeListener) - Method in interface javax.enterprise.deploy.spi.DConfigBean
   Register a property listener for this bean.
addProvider(Provider) - Method in class javax.mail.Session
   Add a provider to the session.
addRecipient(Message.RecipientType, Address) - Method in class javax.mail.Message
   Add this recipient address to the existing ones of the given type.
addRecipients(Message.RecipientType, Address[]) - Method in class javax.mail.internet.MimeMessage
   Add the given addresses to the specified recipient type.
addRecipients(Message.RecipientType, String) - Method in class javax.mail.internet.MimeMessage
   Add the given addresses to the specified recipient type.
addRecipients(Message.RecipientType, Address[]) - Method in class javax.mail.Message
   Add these recipient addresses to the existing ones of the given type.
addRegistryObject(RegistryObject) - Method in interface javax.xml.registry.infomodel.RegistryPackage
   Adds a child RegistryObject as member.
addRegistryObjects(Collection) - Method in interface javax.xml.registry.infomodel.RegistryPackage
   Adds a Collection of RegistryObject children as members.
addRenderer(String, String, Renderer) - Method in class javax.faces.render.RenderKit
   Register the specified Renderer instance, associated with the specified component family and renderertype, to the set of Renderers registered with this RenderKit, replacing any previously registered Renderer for this combination of identifiers.
addRenderKit(String, RenderKit) - Method in class javax.faces.render.RenderKitFactory
Register the specified RenderKit instance, associated with the specified renderKitId, to be supported by this RenderKitFactory, replacing any previously registered RenderKit for this identifier.

**Address** - Class in javax.mail
This abstract class models the addresses in a message.

**Address()** - Constructor for class javax.mail.Address

**address** - Variable in class javax.mail.internet.InternetAddress

**address** - Variable in class javax.mail.search.AddressTerm
The address.

**ADDRESS_LINES_SLOT** - Static variable in interface javax.xml.registry.infomodel.Slot
Name for pre-defined Slot used in PostalAddress by JAXR UDDI provider.

**AddressException** - Exception in javax.mail.internet
The exception thrown when a wrongly formatted address is encountered.

**AddressException()** - Constructor for exception javax.mail.internet.AddressException
Constructs an AddressException with no detail message.

**AddressException(String)** - Constructor for exception javax.mail.internet.AddressException
Constructs an AddressException with the specified detail message.

**AddressException(String, String)** - Constructor for exception javax.mail.internet.AddressException
Constructs an AddressException with the specified detail message and reference info.

**AddressException(String, String, int)** - Constructor for exception javax.mail.internet.AddressException
Constructs an AddressException with the specified detail message and reference info.

**AddressStringTerm** - Class in javax.mail.search
This abstract class implements string comparisons for Message addresses.

**AddressStringTerm(String)** - Constructor for class javax.mail.search.AddressStringTerm
Constructor.

**AddressTerm** - Class in javax.mail.search
This class implements Message Address comparisons.

**AddressTerm(Address)** - Constructor for class javax.mail.search.AddressTerm

**addService(Service)** - Method in interface javax.xml.registry.infomodel.Organization
   Adds a child Service.

**addServiceBinding(ServiceBinding)** - Method in interface javax.xml.registry.infomodel.Service
   Adds a child ServiceBinding.

**addServiceBindings(Collection)** - Method in interface javax.xml.registry.infomodel.Service
   Adds a Collection of ServiceBinding children.

**addServices(Collection)** - Method in interface javax.xml.registry.infomodel.Organization
   Adds a Collection of Service children.

**addSlot(Slot)** - Method in interface javax.xml.registry.infomodel.ExtensibleObject
   Adds a Slot to this object.

**addSlots(Collection)** - Method in interface javax.xml.registry.infomodel.ExtensibleObject
   Adds more Slots to this object.

**addSpecificationLink(SpecificationLink)** - Method in interface javax.xml.registry.infomodel.ServiceBinding
   Adds a child SpecificationLink.

**addSpecificationLinks(Collection)** - Method in interface javax.xml.registry.infomodel.ServiceBinding
   Adds a Collection of SpecificationLink children.

**addStoreListener(StoreListener)** - Method in class javax.mail.Store
   Add a listener for StoreEvents on this Store.

**addSwaRefAttachment(DataHandler)** - Method in class javax.xml.bind.attachment.AttachmentMarshaller
   Add MIME data as an attachment and return attachment's content-id, cid.

**addTextNode(String)** - Method in interface javax.xml.soap.SOAPElement
   Creates a new Text object initialized with the given String and adds it to this SOAPElement object.

**addToExcludedPolicy(PermissionCollection)** - Method in interface
javax.security.jacc.PolicyConfiguration
Used to add excluded policy statements to this PolicyConfiguration.

addToExcludedPolicy(Permission) - Method in interface

addToRole(String, PermissionCollection) - Method in interface

addToRole(String, Permission) - Method in interface

addToUncheckedPolicy(PermissionCollection) - Method in interface

addToUncheckedPolicy(Permission) - Method in interface

addTransformer(ClassTransformer) - Method in interface
JavaMail's PersistenceUnitInfo
Add a transformer supplied by the provider that will be called for every new class definition or class redefinition that gets loaded by the loader returned by the PersistenceUnitInfo.getClassLoader() method.

addTransportListener(TransportListener) - Method in class
JavaMail's Transport
Add a listener for Transport events.

addUpgradeHeaderElement(Iterator) - Method in interface
JavaMail's SOAPHeader
Creates a new Upgrade SOAPHeaderElement object initialized with the specified List of supported SOAP URIs and adds it to this SOAPHeader object.

addUpgradeHeaderElement(String[]) - Method in interface
JavaMail's SOAPHeader
Creates a new Upgrade SOAPHeaderElement object initialized with the specified array of supported SOAP URIs and adds it to this
SOAPHeader object.

**addUpgradeHeaderElement(String)** - Method in interface javax.xml.soap.SOAPHeader

- Creates a new Upgrade SOAPHeaderElement object initialized with the specified supported SOAP URI and adds it to this SOAPHeader object.

**addUser(User)** - Method in interface javax.xml.registry.infomodel.Organization

- Adds a User.

**addUsers(Collection)** - Method in interface javax.xml.registry.infomodel.Organization

- Adds a Collection of Users.

**addValidator(String, String)** - Method in class javax.faces.application.Application

- Register a new mapping of validator id to the name of the corresponding Validator class.

**addValidator(Validator)** - Method in interface javax.faces.componentEditableValueHolder

- Add a Validator instance to the set associated with this component.

**addValidator(Validator)** - Method in class javax.faces.component.UIInput

- Add a Validator instance to the set associated with this UIInput.

**addValueChangeListener(ValueChangeListener)** - Method in interface javax.faces.componentEditableValueHolder

- Add a new ValueChangeListener to the set of listeners interested in being notified when ValueChangeEvent occurs.

**addValueChangeListener(ValueChangeListener)** - Method in class javax.faces.component.UIInput

- Add a new ValueChangeListener to the set of listeners interested in being notified when ValueChangeEvent occurs.

**addVerbatimAfterComponent(UIComponentClassicTagBase, UIComponent, UIComponent)** - Method in class javax.faces.webapp.UIComponentClassicTagBase

- Add verbatim as a sibling of component in component in the parent's child list.

**addVerbatimBeforeComponent(UIComponentClassicTagBase, UIComponent, UIComponent)** - Method in class javax.faces.webapp.UIComponentClassicTagBase

- Add verbatim as a sibling of component in component in the parent's child list.
addXpathListener(String, XpathListener) - Method in interface javax.enterprise.deploy.model.DDBean
Register a listener for a specific XPath.
addXpathListener(ModuleType, String, XpathListener) - Method in interface javax.enterprise.deploy.model.J2eeApplicationObject
Register a listener for changes in XPath that are related to this deployableObject.

afterBegin() - Method in interface javax.ejb.SessionSynchronization
The afterBegin method notifies a session Bean instance that a new transaction has started, and that the subsequent business methods on the instance will be invoked in the context of the transaction.

afterCompletion(boolean) - Method in interface javax.ejb.SessionSynchronization
The afterCompletion method notifies a session Bean instance that a transaction commit protocol has completed, and tells the instance whether the transaction has been committed or rolled back.

afterCompletion(int) - Method in interface javax.transaction.Synchronization
This method is called by the transaction manager after the transaction is committed or rolled back.

afterDelivery() - Method in interface javax.resource.spi.endpoint.MessageEndpoint
This is called by a resource adapter after a message is delivered.

afterMarshal(Object) - Method in class javax.xml.bind.Marshaller.Listener
Callback method invoked after marshalling source to XML.

afterPhase(PhaseEvent) - Method in interface javax.faces.event.PhaseListener
Handle a notification that the processing for a particular phase has just been completed.

afterUnmarshal(Object, Object) - Method in class javax.xml.bind.Unmarshaller.Listener
Callback method invoked after unmarshalling XML data into target.

ALERT - Static variable in class javax.mail.event.StoreEvent
Indicates that this message is an ALERT.
**ALL** - Static variable in class `javax.mail.internet.MimeUtility`

**allocate(XMLStreamReader)** - Method in interface `javax.xml.stream.util.XMLEventAllocator`
This method allocates an event given the current state of the XMLStreamReader.

**allocate(XMLStreamReader, XMLEventConsumer)** - Method in interface `javax.xml.stream.util.XMLEventAllocator`
This method allocates an event or set of events given the current state of the XMLStreamReader and adds the event or set of events to the consumer that was passed in.

**allocateConnection(ManagedConnectionFactory, ConnectionRequestInfo)** - Method in interface `javax.resource.spi.ConnectionManager`
The method `allocateConnection` gets called by the resource adapter's connection factory instance.

**ALLOCATOR** - Static variable in class `javax.xml.stream.XMLInputFactory`
The property used to set/get the implementation of the allocator

**AND_ALL_KEYS** - Static variable in interface `javax.xml.registry.FindQualifier`

**AndTerm** - Class in `javax.mail.search`
This class implements the logical AND operator on individual SearchTerms.

**AndTerm(SearchTerm, SearchTerm)** - Constructor for class `javax.mail.search.AndTerm`
Constructor that takes two terms.

**AndTerm(SearchTerm[])** - Constructor for class `javax.mail.search.AndTerm`
Constructor that takes an array of SearchTerms.

**ANSWERED** - Static variable in class `javax.mail.Flags.Flag`
This message has been answered.

**ANY_PHASE** - Static variable in class `javax.faces.event.PhaseId`
Identifier that indicates an interest in events, no matter which request processing phase is being performed.

**appendFaultSubcode(QName)** - Method in interface `javax.xml.soap.SOAPFault`
Adds a Subcode to the end of the sequence of Subcodes contained
by this SOAPFault.

**appendMessages(Message[])** - Method in class `javax.mail.Folder`
  Append given Messages to this folder.

**Application** - Class in `javax.faces.application`
  **Application** represents a per-web-application singleton object where applications based on JavaServer Faces (or implementations wishing to provide extended functionality) can register application-wide singletons that provide functionality required by JavaServer Faces.

**Application()** - Constructor for class `javax.faces.application.Application`

**APPLICATION** - Static variable in class `javax.servlet.jsp.PageContext`
  Name used to store ServletContext in PageContext name table.

**APPLICATION_FACTORY** - Static variable in class `javax.faces.FactoryFinder`
  The property name for the **ApplicationFactory** class name.

**APPLICATION_SCOPE** - Static variable in class `javax.servlet.jsp.PageContext`
  Application scope: named reference remains available in the ServletContext until it is reclaimed.

**ApplicationException** - Annotation Type in `javax.ejb`
  Applied to an exception to denote that it is an application exception and should be reported to the client directly (i.e., unwrapped).

**ApplicationFactory** - Class in `javax.faces.application`
  **ApplicationFactory** is a factory object that creates (if needed) and returns **Application** instances.

**ApplicationFactory()** - Constructor for class `javax.faces.application.ApplicationFactory`

**ApplicationServerInternalException** - Exception in `javax.resource.spi`
  **ApplicationServerInternalException** is thrown by an application server to indicate error conditions specific to an application server.

**ApplicationServerInternalException()** - Constructor for exception `javax.resource.spi.ApplicationServerInternalException`
  Constructs a new instance with null as its detail message.

**ApplicationServerInternalException(String)** - Constructor for exception `javax.resource.spi.ApplicationServerInternalException`
  Constructs a new instance with the specified detail message.

**ApplicationServerInternalException(Throwable)** - Constructor for exception `javax.resource.spi.ApplicationServerInternalException`
exception javax.resource.spi.ApplicationServerInternalException
Constructs a new throwable with the specified cause.

**ApplicationServerInternalException(String, Throwable)** - Constructor for exception javax.resource.spi.ApplicationServerInternalException
Constructs a new throwable with the specified detail message and cause.

**ApplicationServerInternalException(String, String)** - Constructor for exception javax.resource.spi.ApplicationServerInternalException
Constructs a new throwable with the specified detail message and an error code.

**APPLY_REQUEST_VALUES** - Static variable in class javax.faces.event.PhaseId
Identifier that indicates an interest in events queued for the Apply Request Values phase of the request processing lifecycle.

**AroundInvoke** - Annotation Type in javax.interceptor

**ArrayDataModel** - Class in javax.faces.model
ArrayDataModel is a convenience implementation of DataModel that wraps an array of Java objects.

**ArrayDataModel()** - Constructor for class javax.faces.model.ArrayDataModel
Construct a new ArrayDataModel with no specified wrapped data.

**ArrayDataModel(Object[])** - Constructor for class javax.faces.model.ArrayDataModel
Construct a new ArrayDataModel wrapping the specified array.

**ArrayELResolver** - Class in javax.el
Defines property resolution behavior on arrays.

**ArrayELResolver()** - Constructor for class javax.el.ArrayELResolver
Creates a new read/write ArrayELResolver.

**ArrayELResolver(boolean)** - Constructor for class javax.el.ArrayELResolver
Creates a new ArrayELResolver whose read-only status is determined by the given parameter.

**asCharacters()** - Method in interface javax.xml.stream.events.XMLEvent
Returns this event as Characters, may result in a class cast exception if this event is not Characters.

**asEndElement()** - Method in interface javax.xml.stream.events.XMLEvent
Returns this event as an end element event, may result in a class
cast exception if this event is not a end element.

`associateConnection(Object, ManagedConnectionFactory, ConnectionRequestInfo)` - Method in interface `javax.resource.spi.LazyAssociatableConnectionManager`

This method is called by a resource adapter (that is capable of lazy connection association optimization) in order to lazily associate a connection object with a `ManagedConnection` instance.

`associateConnection(Object)` - Method in interface `javax.resource.spi.ManagedConnection`

Used by the container to change the association of an application-level connection handle with a `ManagedConnection` instance.

**Association** - Interface in `javax.xml.registry.infomodel`

A `RegistryObject` instance may be associated with zero or more `RegistryObject` instances.

**ASSOCIATION** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**AssociationOverride** - Annotation Type in `javax.persistence`

This annotation is used to override a many-to-one or one-to-one mapping of property or field for an entity relationship.

**AssociationOverrides** - Annotation Type in `javax.persistence`

This annotation is used to override mappings of multiple many-to-one or one-to-one relationship properties or fields.

`asStartElement()` - Method in interface `javax.xml.stream.events.XMLEvent`

Returns this event as a start element event, may result in a class cast exception if this event is not a start element.

**AsyncHandler<T>** - Interface in `javax.xml.ws`

The `AsyncHandler` interface is implemented by clients that wish to receive callback notification of the completion of service endpoint operations invoked asynchronously.

**AT_BEGIN** - Static variable in class `javax.servlet.jsp.tagext.VariableInfo`

Scope information that scripting variable is visible after start tag.

**AT_END** - Static variable in class `javax.servlet.jsp.tagext.VariableInfo`

Scope information that scripting variable is visible after end tag.

**ATOM** - Static variable in class `javax.mail.internet.HeaderTokenizer.Token`

Token type indicating an ATOM.

`attachFile(File)` - Method in class `javax.mail.internet.MimeBodyPart`
Use the specified file to provide the data for this part.
attachFile(String) - Method in class javax.mail.internet.MimeBodyPart
Use the specified file to provide the data for this part.
ATTACHMENT - Static variable in interface javax.mail.Part
This part should be presented as an attachment.
AttachmentMarshaller - Class in javax.xml.bind.attachment
   Enable JAXB marshalling to optimize storage of binary data.
AttachmentMarshaller() - Constructor for class javax.xml.bind.attachment.AttachmentMarshaller

AttachmentPart - Class in javax.xml.soap
   A single attachment to a SOAPMessage object.
AttachmentPart() - Constructor for class javax.xml.soap.AttachmentPart

AttachmentUnmarshaller - Class in javax.xml.bind.attachment
   Enables JAXB unmarshalling of a root document containing optimized binary data formats.
AttachmentUnmarshaller() - Constructor for class javax.xml.bind.attachment.AttachmentUnmarshaller

Attribute - Interface in javax.xml.stream.events
   An interface that contains information about an attribute.
ATTRIBUTE - Static variable in interface javax.xml.stream.XMLStreamConstants
   Indicates an event is an attribute
attributeAdded(HttpSessionBindingEvent) - Method in interface javax.servlet.http.HttpSessionAttributeListener
   Notification that an attribute has been added to a session.
attributeAdded(ServletContextAttributeEvent) - Method in interface javax.servlet.ServletContextAttributeListener
   Notification that a new attribute was added to the servlet context.
attributeAdded(ServletRequestAttributeEvent) - Method in interface javax.servlet.ServletRequestAttributeListener
   Notification that a new attribute was added to the servlet request.
AttributeOverride - Annotation Type in javax.persistence
   The AttributeOverride annotation is used to override the mapping of a Basic (whether explicit or default) property or field or Id property or field.
AttributeOverrides - Annotation Type in javax.persistence
Is used to override mappings of multiple properties or fields. 

**attributeRemoved(HttpSessionBindingEvent)** - Method in interface javax.servlet.http.HttpSessionAttributeListener

Notification that an attribute has been removed from a session.

**attributeRemoved(ServletContextAttributeEvent)** - Method in interface javax.servlet.ServletContextAttributeListener

Notification that an existing attribute has been removed from the servlet context.

**attributeRemoved(ServletRequestAttributeEvent)** - Method in interface javax.servlet.ServletRequestAttributeListener

Notification that an existing attribute has been removed from the servlet request.

**attributeReplaced(HttpSessionBindingEvent)** - Method in interface javax.servlet.http.HttpSessionAttributeListener

Notification that an attribute has been replaced in a session.

**attributeReplaced(ServletContextAttributeEvent)** - Method in interface javax.servlet.ServletContextAttributeListener

Notification that an attribute on the servlet context has been replaced.

**attributeReplaced(ServletRequestAttributeEvent)** - Method in interface javax.servlet.ServletRequestAttributeListener

Notification that an attribute was replaced on the servlet request.

**AttributeTag** - Class in javax.faces.webapp

*Deprecated.* The Faces implementation must now provide the implementation for this class.

**AttributeTag()** - Constructor for class javax.faces.webapp.AttributeTag

*Deprecated.*

**AUDITABLE_EVENT** - Static variable in interface javax.xml.registry.LifeCycleManager

**AuditableEvent** - Interface in javax.xml.registry.infomodel

AuditableEvent instances provide a long term record of events that effect a change of state in a RegistryObject.

**AuthenticationFailedException** - Exception in javax.mail

This exception is thrown when the connect method on a Store or Transport object fails due to an authentication failure (e.g., bad user name or password).

**AuthenticationFailedException()** - Constructor for exception javax.mail.AuthenticationFailedException
Constructor

**AuthenticationFailedException(String)** - Constructor for exception `javax.mail.AuthenticationFailedException`

Constructor

**Authenticator** - Class in `javax.mail`

The class Authenticator represents an object that knows how to obtain authentication for a network connection.

**Authenticator()** - Constructor for class `javax.mail.Authenticator`

**AUTHORIZED_NAME SLOT** - Static variable in interface `javax.xml.registry.infomodel.Slot`

Name for pre-defined Slot used in Organization and ClassificationScheme by JAXR UDDI provider.

**AUTO_ACKNOWLEDGE** - Static variable in interface `javax.jms.Session`

With this acknowledgment mode, the session automatically acknowledges a client's receipt of a message either when the session has successfully returned from a call to `receive` or when the message listener the session has called to process the message successfully returns.

**autoFlush** - Variable in class `javax.servlet.jsp.JspWriter`

Whether the JspWriter is autoflushing.

**available()** - Method in class `javax.mail.util.SharedFileInputStream`

Returns the number of bytes that can be read from this input stream without blocking.
Basic - Annotation Type in javax.persistence
   The Basic annotation is the simplest type of mapping to a database column.

BASIC_AUTH - Static variable in class
javax.faces.context.ExternalContext
   String identifier for BASIC authentication.

BASIC_AUTH - Static variable in interface
javax.servlet.http.HttpServletRequest
   String identifier for Basic authentication.

BCC - Static variable in class javax.mail.Message.RecipientType
   The "Bcc" (blind carbon copy) recipients.

BEAN_ADDED - Static variable in class
javax.enterprise.deploy.model.XpathEvent
   Adding a DDBean

BEAN_CHANGED - Static variable in class
javax.enterprise.deploy.model.XpathEvent
   Changing a DDBean

BEAN_REMOVED - Static variable in class
javax.enterprise.deploy.model.XpathEvent
   Removing a DDBean

BeanELResolver - Class in javax.el
   Defines property resolution behavior on objects using the JavaBeans component architecture.

BeanELResolver() - Constructor for class java.el.BeanELResolver
   Creates a new read/write BeanELResolver.

BeanELResolver(boolean) - Constructor for class java.el.BeanELResolver
   Creates a new BeanELResolver whose read-only status is determined by the given parameter.

BeanELResolver.BeanProperties - Class in javax.el

BeanELResolver.BeanProperties(Class<?>) - Constructor for class
java.el.BeanELResolver.BeanProperties

BeanELResolver.BeanProperty - Class in javax.el
**BeanELResolver.BeanProperty(Class<?>, PropertyDescriptor)** - Constructor for class `javax.el.BeanELResolver.BeanProperty`

**BeanNotFoundException** - Exception in `javax.enterprise.deploy.spi.exceptions`
   This exception is to report that the bean is not a child of the parent bean.

**BeanNotFoundException(String)** - Constructor for exception `javax.enterprise.deploy.spi.exceptions.BeanNotFoundException`
   Creates a new BeanNotFoundException object.

**beforeCompletion()** - Method in interface `javax.ejb.SessionSynchronization`
   The beforeCompletion method notifies a session Bean instance that a transaction is about to be committed.

**beforeCompletion()** - Method in interface `javax.transaction.Synchronization`
   The beforeCompletion method is called by the transaction manager prior to the start of the two-phase transaction commit process.

**beforeDelivery(Method)** - Method in interface `javax.resource.spi.endpoint.MessageEndpoint`
   This is called by a resource adapter before a message is delivered.

**beforeMarshal(Object)** - Method in class `javax.xml.bind.Marshaller.Listener`
   Callback method invoked before marshalling from source to XML.

**beforePhase(PhaseEvent)** - Method in interface `javax.faces.event.PhaseListener`
   Handle a notification that the processing for a particular phase of the request processing lifecycle is about to begin.

**beforeUnmarshal(Object, Object)** - Method in class `javax.xml.bind.Unmarshaller.Listener`
   Callback method invoked before unmarshalling into target.

**begin()** - Method in interface `javax.persistence.EntityTransaction`
   Start the resource transaction.

**begin()** - Method in interface `javax.resource.cci.LocalTransaction`
Begins a local transaction on an EIS instance.

**begin()** - Method in interface `javax.resource.spi.LocalTransaction`
Begin a local transaction

**begin()** - Method in interface `javax.transaction.TransactionManager`
Create a new transaction and associate it with the current thread.

**begin()** - Method in interface `javax.transaction.UserTransaction`
Create a new transaction and associate it with the current thread.

**BigDecimalConverter** - Class in `javax.faces.convert` Converter implementation for `java.math.BigDecimal` values.

**BigDecimalConverter()** - Constructor for class `javax.faces.convert.BigDecimalConverter`

**BigDecimalHolder** - Class in `javax.xml.rpc.holders`

**BigDecimalHolder()** - Constructor for class `javax.xml.rpc.holders.BigDecimalHolder`

**BigDecimalHolder(BigDecimal)** - Constructor for class `javax.xml.rpc.holders.BigDecimalHolder`

**BIGINTEGER_ID** - Static variable in class `javax.faces.convert.BigIntegerConverter`
The message identifier of the `FacesMessage` to be created if the conversion to `BigInteger` fails.

**BigIntegerConverter** - Class in `javax.faces.convert` Converter implementation for `java.math.BigInteger` values.

**BigIntegerConverter()** - Constructor for class `javax.faces.convert.BigIntegerConverter`

**BigIntegerHolder** - Class in `javax.xml.rpc.holders`

**BigIntegerHolder()** - Constructor for class `javax.xml.rpc.holders.BigIntegerHolder`

**BigIntegerHolder(BigInteger)** - Constructor for class `javax.xml.rpc.holders.BigIntegerHolder`

**Binder<XmlNode>** - Class in `javax.xml.bind` Enable synchronization between XML infoset nodes and JAXB
objects representing same XML document.

**Binder()** - Constructor for class `javax.xml.bind.Binder`

**Binding** - Interface in `javax.xml.ws`
   The `Binding` interface is the base interface for JAX-WS protocol bindings.

**BindingProvider** - Interface in `javax.xml.ws`
   The `BindingProvider` interface provides access to the protocol binding and associated context objects for request and response message processing.

**bindings** - Variable in class `javax.faces.component.UICOMPONENT`

**BindingType** - Annotation Type in `javax.xml.ws`
   The `BindingType` annotation is used to specify the binding to use for a web service endpoint implementation class.

**BODY_CONTENT_EMPTY** - Static variable in class `javax.servlet.jsp.tagext.TagInfo`
   Static constant for `getBodyContent()` when it is empty.

**BODY_CONTENT_JSP** - Static variable in class `javax.servlet.jsp.tagext.TagInfo`
   Static constant for `getBodyContent()` when it is JSP.

**BODY_CONTENT_SCRIPTLESS** - Static variable in class `javax.servlet.jsp.tagext.TagInfo`
   Static constant for `getBodyContent()` when it is scriptless.

**BODYCONTENT** - Static variable in class `javax.servlet.jsp.tagext.TagInfo`
   Static constant for `getBodyContent()` when it is Tag dependent.

**bodyContent** - Variable in class `javax.faces.webapp.UICOMPONENTClassicTagBase`
   The `bodyContent` for this tag handler.

**BodyContent** - Class in `javax.servlet.jsp.tagext`
   An encapsulation of the evaluation of the body of an action so it is available to a tag handler.

**BodyContent(JspWriter)** - Constructor for class `javax.servlet.jsp.tagext.BodyContent`
   Protected constructor.

**bodyContent** - Variable in class `javax.servlet.jsp.tagext.BodyTagSupport`
   The current `BodyContent` for this `BodyTag`.

**BodyPart** - Class in `javax.mail`
This class models a Part that is contained within a Multipart. 
**BodyPart()** - Constructor for class `javax.mail.BodyPart`

**BodyTag** - Interface in `javax.servlet.jsp.tagext`
The BodyTag interface extends IterationTag by defining additional methods that let a tag handler manipulate the content of evaluating its body.

**BodyTagSupport** - Class in `javax.servlet.jsp.tagext`
A base class for defining tag handlers implementing BodyTag. 
**BodyTagSupport()** - Constructor for class `javax.servlet.jsp.tagext.BodyTagSupport`
Default constructor, all subclasses are required to only define a public constructor with the same signature, and to call the superclass constructor.

**BodyTerm** - Class in `javax.mail.search`
This class implements searches on a Message Body.
**BodyTerm(String)** - Constructor for class `javax.mail.search.BodyTerm`

**BOOLEAN_ID** - Static variable in class `javax.faces.convert.BooleanConverter`
The message identifier of the `FacesMessage` to be created if the conversion to `Boolean` fails.

**BooleanConverter** - Class in `javax.faces.convert`
`Converter` implementation for `java.lang.Boolean` (and boolean primitive) values.
**BooleanConverter()** - Constructor for class `javax.faces.convert.BooleanConverter`

**BooleanHolder** - Class in `javax.xml.rpc.holders`

**BooleanHolder()** - Constructor for class `javax.xml.rpc.holders.BooleanHolder`

**BooleanHolder(boolean)** - Constructor for class `javax.xml.rpc.holders.BooleanHolder`

**BooleanWrapperHolder** - Class in `javax.xml.rpc.holders`

**BooleanWrapperHolder()** - Constructor for class
**BooleanWrapperHolder** - Constructor for class javax.xml.rpc.holders.BooleanWrapperHolder

**BootstrapContext** - Interface in javax.resource.spi
This provides a mechanism to pass a bootstrap context to a resource adapter instance when it is bootstrapped.

**BoundaryStatistic** - Interface in javax.management.j2ee.statistics
Specifies standard measurements of the upper and lower limits of the value of an attribute.

**BoundedRangeStatistic** - Interface in javax.management.j2ee.statistics
The BoundedRangeStatistic model aggregates the attributes of RangeStatistic and BoundaryStatistic and provides standard measurements of a range that has fixed limits.

**broadcast(FacesEvent)** - Method in class javax.faces.component/UICommand
In addition to to the default
UIComponent.broadcast(javax.faces.event.FacesEvent) processing, pass the ActionEvent being broadcast to the method referenced by actionListener (if any), and to the default ActionListener registered on the Application.

**broadcast(FacesEvent)** - Method in class javax.faces.component/UIComponent
Broadcast the specified FacesEvent to all registered event listeners who have expressed an interest in events of this type.

**broadcast(FacesEvent)** - Method in class javax.faces.component/UIComponentBase
Override the default
UIComponentBase.broadcast(javax.faces.event.FacesEvent) processing to unwrap any wrapped FacesEvent and reset the current row index, before the event is actually broadcast.

**bufferSize** - Variable in class javax.servlet.jsp.JspWriter
The size of the buffer used by the JspWriter.

**bufpos** - Variable in class javax.mail.util.SharedFileInputStream
The file offset that corresponds to the first byte in the read buffer.
bufsize - Variable in class javax.mail.util.**SharedFileInputStream**
The normal size of the read buffer.

BulkResponse - Interface in **javax.xml.registry**
Contains the response of a method in the API that performs a bulk operation and returns a bulk response.

BusinessLifeCycleManager - Interface in **javax.xml.registry**
The BusinessLifeCycleManager interface, which is exposed by the Registry Service, implements the life cycle management functionality of the Registry as part of a business level API.

BusinessQueryManager - Interface in **javax.xml.registry**
The BusinessQueryManager interface, which is exposed by the Registry Service, implements the business style query interface.

BYTE_ID - Static variable in class javax.faces.convert.**ByteConverter**
The message identifier of the **FacesMessage** to be created if the conversion to **Byte** fails.

ByteArrayDataSource - Class in **javax.mail.util**
A DataSource backed by a byte array.

ByteArrayDataSource(InputStream, String) - Constructor for class javax.mail.util.**ByteArrayDataSource**
Create a ByteArrayDataSource with data from the specified InputStream and with the specified MIME type.

ByteArrayDataSource(byte[], String) - Constructor for class javax.mail.util.**ByteArrayDataSource**
Create a ByteArrayDataSource with data from the specified byte array and with the specified MIME type.

ByteArrayDataSource(String, String) - Constructor for class javax.mail.util.**ByteArrayDataSource**
Create a ByteArrayDataSource with data from the specified String and with the specified MIME type.

ByteArrayHolder - Class in **javax.xml.rpc.holders**

ByteArrayHolder() - Constructor for class javax.xml.rpc.holders.**ByteArrayHolder**

ByteArrayHolder(byte[]) - Constructor for class javax.xml.rpc.holders.**ByteArrayHolder**

ByteConverter - Class in **javax.faces.convert**
Converter implementation for java.lang.Byte (and byte primitive)
values.

**ByteConverter()** - Constructor for class java.util.ByteConverter

**ByteHolder** - Class in javax.xml.rpc.holders

**ByteHolder()** - Constructor for class javax.xml.rpc.holders.ByteHolder

**ByteHolder(byte)** - Constructor for class javax.xml.rpc.holders.ByteHolder

**BytesMessage** - Interface in javax.jms

A `BytesMessage` object is used to send a message containing a stream of uninterpreted bytes.

**ByteWrapperHolder** - Class in javax.xml.rpc.holders

**ByteWrapperHolder()** - Constructor for class javax.xml.rpc.holders.ByteWrapperHolder

**ByteWrapperHolder(Byte)** - Constructor for class javax.xml.rpc.holders.ByteWrapperHolder
calculateCharacterEncoding(FacesContext) - Method in class javax.faces.application.ViewHandler
Returns the correct character encoding to be used for this request.

calculateCharacterEncoding(FacesContext) - Method in class javax.faces.application.ViewHandlerWrapper
The default behavior of this method is to call ViewHandler.calculateCharacterEncoding(javax.faces.context.FacesContext) on the wrapped ViewHandler object.

calculateLocale(FacesContext) - Method in class javax.faces.application.ViewHandler
Returns an appropriate Locale to use for this and subsequent requests for the current client.

calculateLocale(FacesContext) - Method in class javax.faces.application.ViewHandlerWrapper
The default behavior of this method is to call ViewHandler.calculateLocale(javax.faces.context.FacesContext) on the wrapped ViewHandler object.

calculateRenderKitId(FacesContext) - Method in class javax.faces.application.ViewHandler
Return an appropriate renderKitId for this and subsequent requests from the current client.

calculateRenderKitId(FacesContext) - Method in class javax.faces.application.ViewHandlerWrapper
The default behavior of this method is to call ViewHandler.calculateRenderKitId(javax.faces.context.FacesContext) on the wrapped ViewHandler object.

CalendarHolder - Class in javax.xml.rpc.holders

CalendarHolder() - Constructor for class javax.xml.rpc.holders.CalendarHolder

CalendarHolder(Calendar) - Constructor for class javax.xml.rpc.holders.CalendarHolder

Call - Interface in javax.xml.rpc
The javax.xml.rpc.Call interface provides support for the dynamic
invocation of a service endpoint.

**call(SOAPMessage, Object)** - Method in class javax.xml.soap.SOAPConnection
Sends the given message to the specified endpoint and blocks until it has returned the response.

**canBeRequestTime()** - Method in class javax.servlet.jsp.tagext.TagAttributeInfo
Whether this attribute has been specified in the TLD as rtexprvalue.

**cancel()** - Method in interface javax.ejb.Timer
Cause the timer and all its associated expiration notifications to be cancelled.

**CANCEL** - Static variable in class javax.enterprise.deploy.shared.ActionType
A cancel operation is being preformed on the DeploymentManager action command.

**cancel()** - Method in interface javax.enterprise.deploy.spi.status.ProgressObject
(optional) A cancel request on an in-process operation stops all further processing of the operation and returns the environment to its original state before the operation was executed.

**CapabilityProfile** - Interface in javax.xml.registry
Provides information about the capabilities of a JAXR provider.

**CAR** - Static variable in class javax.enterprise.deploy.shared.ModuleType
The module is an Client Application archive.

**CascadeType** - Enum in javax.persistence
Defines the set of cascadable operations that are propagated to the associated entity.

**CASE_SENSITIVE_MATCH** - Static variable in interface javax.xml.registry.FindQualifier

**cause** - Variable in exception javax.xml.registry.JAXRException

**CC** - Static variable in class javax.mail.Message.RecipientType
The "Cc" (carbon copy) recipients.

**CDATA** - Static variable in interface javax.xml.stream.XMLStreamConstants
Indicates an event is a CDATA section

**CHARACTER_ENCODING_KEY** - Static variable in class javax.faces.application.ViewHandler
The key, in the session's attribute set, under which the response character encoding may be stored and retrieved.

**CHARACTER_ID** - Static variable in class javax.faces.convert.CharacterConverter
The message identifier of the FacesMessage to be created if the conversion to Character fails.

**CHARACTER_SET_ENCODING** - Static variable in class javax.xml.soap.SOAPMessage
Specifies the character type encoding for the SOAP Message.

**CharacterConverter** - Class in javax.faces.convert
Converter implementation for java.lang.Character (and char primitive) values.

**CharacterConverter()** - Constructor for class javax.faces.convert.CharacterConverter

**Characters** - Interface in javax.xml.stream.events
This describes the interface to Characters events.

**CHARACTERS** - Static variable in interface javax.xml.stream.XMLStreamConstants
Indicates an event is characters

**Classification** - Interface in javax.xml.registry.infomodel
The Classification interface is used to classify RegistryObject instances.

**CLASSIFICATION** - Static variable in interface javax.xml.registry.LifeCycleManager

**CLASSIFICATION_SCHEME** - Static variable in interface javax.xml.registry.LifeCycleManager

**ClassificationScheme** - Interface in javax.xml.registry.infomodel
A ClassificationScheme instance represents a taxonomy that may be used to classify or categorize RegistryObject instances.

**ClassTransformer** - Interface in javax.persistence.spi
A persistence provider supplies an instance of this interface to the PersistenceUnitInfo.addTransformer method.

**cleanup()** - Method in interface javax.resource.spi.ManagedConnection
Application server calls this method to force any cleanup on the ManagedConnection instance.

**clear()** - Method in interface javax.persistence.EntityManager
Clear the persistence context, causing all managed entities to become detached.

**clear()** - Method in class `javax.servlet.jsp.JspWriter`
Clear the contents of the buffer.

**clear()** - Method in interface `javax.xml.rpc.encoding.TypeMappingRegistry`
Removes all registered TypeMappings and encodingStyleURIs from this TypeMappingRegistry.

**clearBody()** - Method in interface `javax.jms.Message`
Clears out the message body.

**clearBody()** - Method in class `javax.servlet.jsp.tagext.BodyContent`
Clear the body without throwing any exceptions.

**clearBuffer()** - Method in class `javax.servlet.jsp.JspWriter`
Clears the current contents of the buffer.

**clearContent()** - Method in class `javax.xml.soap.AttachmentPart`
Clears out the content of this AttachmentPart object.

**clearProperties()** - Method in interface `javax.jms.Message`
Clears a message's properties.

**clearWarnings()** - Method in interface `javax.resource.cci.Interaction`
Clears all the warning reported by this Interaction instance.

**CLIENT_ACKNOWLEDGE** - Static variable in interface `javax.jms.Session`
With this acknowledgment mode, the client acknowledges a consumed message by calling the message's **acknowledge** method.

**CLIENT_CERT_AUTH** - Static variable in class `javax.faces.context.ExternalContext`
String identifier for **CLIENT_CERT** authentication.

**CLIENT_CERT_AUTH** - Static variable in interface `javax.servlet.http.HttpServletRequest`
String identifier for Client Certificate authentication.

**ClientConfiguration** - Interface in `javax.enterprise.deploy.spi.status`
The ClientConfiguration object installs, configures and executes an Application Client.

**ClientExecuteException** - Exception in `javax.enterprise.deploy.spi.exceptions`
This exception reports errors in setting up an application client for execution.

**ClientExecuteException()** - Constructor for exception
`javax.enterprise.deploy.spi.exceptions.ClientExecuteException`
Creates new ClientExecuteException without detail message.  

**ClientExecuteException(String)** - Constructor for exception

```
javax.enterprise.deploy.spi.exceptions.ClientExecuteException
```

Constructs an ClientExecuteException with the specified detail message.

**clone()** - Method in class `javax.mail.Flags`

```
Returns a clone of this Flags object.
```

**clone()** - Method in class `javax.mail.internet.InternetAddress`

```
Return a copy of this InternetAddress object.
```

**clone()** - Method in interface `javax.resource.cci.Record`

```
Creates and returns a copy of this object.
```

**clone()** - Method in class `javax.servlet.http.Cookie`

```
Overrides the standard java.lang.Object.clone method to return a copy of this cookie.
```

**cloneWithWriter(Writer)** - Method in class `javax.faces.context.ResponseWriter`

```
Create and return a new instance of this ResponseWriter, using the specified Writer as the output destination.
```

**cloneWithWriter(Writer)** - Method in class `javax.faces.context.ResponseWriterWrapper`

```
The default behavior of this method is to call ResponseWriter.cloneWithWriter(java.io.Writer) on the wrapped ResponseWriter object.
```

**close()** - Method in class `javax.faces.context.ResponseWriterWrapper`

```
The default behavior of this method is to call Writer.close() on the wrapped ResponseWriter object.
```

**close()** - Method in interface `javax.jms.Connection`

```
Closes the connection.
```

**close()** - Method in interface `javax.jms.ConnectionConsumer`

```
Closes the connection consumer.
```

**close()** - Method in interface `javax.jms.MessageConsumer`

```
Closes the message consumer.
```

**close()** - Method in interface `javax.jms.MessageProducer`

```
Closes the message producer.
```

**close()** - Method in interface `javax.jms.QueueBrowser`

```
Closes the QueueBrowser.
```

**close()** - Method in class `javax.jms.QueueRequestor`

```
Closes the QueueRequestor and its session.
```

**close()** - Method in interface `javax.jms.Session`
Closes the session.

**close()** - Method in class `javax.jms.TopicRequestor`
Closes the `TopicRequestor` and its session.

**close(boolean)** - Method in class `javax.mail.Folder`
Close this Folder.

**close()** - Method in class `javax.mail.Service`
Close this service and terminate its connection.

**close()** - Method in class `javax.mail.util.SharedFileInputStream`
Closes this input stream and releases any system resources associated with the stream.

**close()** - Method in interface `javax.persistence.EntityManager`
Close an application-managed `EntityManager`.

**close()** - Method in interface `javax.persistence.EntityManagerFactory`
Close the factory, releasing any resources that it holds.

**close()** - Method in interface `javax.resource.cci.Connection`
Initiates close of the connection handle at the application level.

**close()** - Method in interface `javax.resource.cci.Interaction`
Closes the current Interaction and release all the resources held for this instance by the resource adapter.

**close()** - Method in class `javax.servlet.jsp.JspWriter`
Close the stream, flushing it first.

**close()** - Method in interface `javax.xml.registry.Connection`
Closes a Connection when it is no longer needed.

**close()** - Method in class `javax.xml.soap.SOAPConnection`
Closes this SOAPConnection object.

**close()** - Method in class `javax.xml.stream.util.EventReaderDelegate`

**close()** - Method in class `javax.xml.stream.utilStreamReaderDelegate`

**close()** - Method in interface `javax.xml.stream.XMLEventReader`
Frees any resources associated with this Reader.

**close()** - Method in interface `javax.xml.stream.XMLEventWriter`
Frees any resources associated with this stream.

**close()** - Method in interface `javax.xml.stream.XMLStreamReader`
Frees any resources associated with this Reader.

**close()** - Method in interface `javax.xml.stream.XMLStreamWriter`
Close this writer and free any resources associated with the writer.

**close(MessageContext)** - Method in interface `javax.xml.ws.handler.Handler`
Called at the conclusion of a message exchange pattern just prior to
the JAX-WS runtime dispatching a message, fault or exception.

**closed(ConnectionEvent)** - Method in class
javax.mail.event.ConnectionAdapter

**CLOSED** - Static variable in class javax.mail.event.ConnectionEvent
A connection was closed.

**closed(ConnectionEvent)** - Method in interface
javax.mail.event.ConnectionListener
Invoked when a Store/Folder/Transport is closed.

**coerceToType(Object, Class<?>)** - Method in class
javax.el.ExpressionFactory
Coerces an object to a specific type according to the EL type
conversion rules.

**CollapsedStringAdapter** - Class in javax.xml.bind.annotation.adapters
Built-in XmlAdapter to handle xs:token and its derived types.

**CollapsedStringAdapter**() - Constructor for class
javax.xml.bind.annotation.adapters.CollapsedStringAdapter

**Column** - Annotation Type in javax.persistence
Is used to specify a mapped column for a persistent property or field.

**ColumnResult** - Annotation Type in javax.persistence
References name of a column in the SELECT clause of a SQL query
- i.e., column alias, if applicable.

**COMBINE_CLASSIFICATIONS** - Static variable in interface
javax.xml.registry.FindQualifier
Maps to UDDI combineCategoryBags.

**CommandInfo** - Class in javax.activation
The CommandInfo class is used by CommandMap implementations
to describe the results of command requests.

**CommandInfo(String, String)** - Constructor for class
javax.activation.CommandInfo
The Constructor for CommandInfo.

**CommandMap** - Class in javax.activation
The CommandMap class provides an interface to a registry of
command objects available in the system.

**CommandMap()** - Constructor for class javax.activation.CommandMap

**CommandObject** - Interface in javax.activation
JavaBeans components that are Activation Framework aware implement this interface to find out which command verb they're being asked to perform, and to obtain the DataHandler representing the data they should operate on.

**CommandType** - Class in javax.enterprise.deploy.shared
Class CommandTypes defines enumeration values for the DeploymentStatus object.

**CommandType(int)** - Constructor for class javax.enterprise.deploy.shared.CommandType
Construct a new enumeration value with the given integer value.

**COMMENT** - Static variable in class javax.mail.internet.HeaderTokenizer.Token
Token type indicating a comment.

**Comment** - Interface in javax.xml.stream.events
An interface for comment events

**COMMENT** - Static variable in interface javax.xml.stream.XMLStreamConstants
Indicates an event is a comment

**CommException** - Exception in javax.resource.spi
This indicates errors related to failed or interrupted communication with an EIS instance.

**CommException()** - Constructor for exception javax.resource.spi.CommException
Constructs a new instance with null as its detail message.

**CommException(String)** - Constructor for exception javax.resource.spi.CommException
Constructs a new instance with the specified detail message.

**CommException(Throwable)** - Constructor for exception javax.resource.spi.CommException
Constructs a new throwable with the specified cause.

**CommException(String, Throwable)** - Constructor for exception javax.resource.spi.CommException
Constructs a new throwable with the specified detail message and cause.

**CommException(String, String)** - Constructor for exception javax.resource.spi.CommException
Constructs a new throwable with the specified detail message and an error code.

**commit()** - Method in interface javax.jms.Session
Commits all messages done in this transaction and releases any locks currently held.

**commit()** - Method in interface javax.jms.XASession

Throws a TransactionInProgressException, since it should not be called for an XASession object.

**commit()** - Method in interface javax.persistence.EntityTransaction

Commit the current transaction, writing any unflushed changes to the database.

**commit()** - Method in interface javax.resource.cci.LocalTransaction

Commits the current local transaction and release all locks held by the underlying EIS instance.

**commit()** - Method in interface javax.resource.spi.LocalTransaction

Commit a local transaction

**commit(Xid, boolean)** - Method in interface javax.resource.spi.XATerminator

Commits the global transaction specified by xid.

**commit()** - Method in interface javax.security.jacc.PolicyConfiguration

This method is used to set to "inService" the state of the policy context whose interface is this PolicyConfiguration Object.

**commit()** - Method in interface javax.transaction.Transaction

Complete the transaction represented by this Transaction object.

**commit()** - Method in interface javax.transaction.TransactionManager

Complete the transaction associated with the current thread.

**commit()** - Method in interface javax.transaction.UserTransaction

Complete the transaction associated with the current thread.

**commit(Xid, boolean)** - Method in interface javax.transaction.xa.XAResource

Commits the global transaction specified by xid.

**compareTo(Object)** - Method in class javax.faces.application.FacesMessage.Severity

Compare this FacesMessage.Severity instance to the specified one.

**compareTo(Object)** - Method in class javax.faces.event.PhaseId

Compare this PhaseId instance to the specified one.

**compareValues(Object, Object)** - Method in class javax.faces.component.UInput

Return true if the new value is different from the previous value.

**compareValues(Object, Object)** - Method in class javax.faces.component.UISelectMany

Return true if the new value is different from the previous value.
**comparison** - Variable in class `javax.mail.search.ComparisonTerm`  
The comparison.

**ComparisonTerm** - Class in `javax.mail.search`  
This class models the comparison operator.

**ComparisonTerm()** - Constructor for class `javax.mail.search.ComparisonTerm`

**COMPLETED** - Static variable in class `javax.enterprise.deploy.shared.StateType`  
The action operation has completed normally.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIColumn`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UICommand`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIData`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIForm`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIGraphic`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIInput`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIMessage`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIMessages`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UINamingContainer`  
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class `javax.faces.component.UIOutput`  
The standard component family for this component.
The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UIPanel**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UIParameter**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UISelectBoolean**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UISelectItem**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UISelectItems**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UISelectMany**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UISelectOne**

The standard component family for this component.

**COMPONENT_FAMILY** - Static variable in class
javafx.faces.component.**UIViewRoot**

The standard component family for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.**HtmlColumn**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.**HtmlCommandButton**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.**HtmlCommandLink**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.**HtmlDataTableView**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.**HtmlForm**
The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlGraphicImage

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlInputHidden

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlInputSecret

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlInputText

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlInputTextarea

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlMessage

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlMessages

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlOutputFormat

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlOutputLabel

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlOutputLink

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlOutputText

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlPanelGrid

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javax.faces.component.html.HtmlPanelGroup
The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectBooleanCheckbox

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectManyCheckbox

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectManyListbox

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectManyMenu

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectOneListbox

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectOneMenu

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.html.HtmlSelectOneRadio

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UIColumn

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UICommand

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UIData

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UIForm

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UIGraphic

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class
javafx.faces.component.UIInput
The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIMessage**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIMessages**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UINamingContainer**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIOutput**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIPanel**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIParameter**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UISelectBoolean**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UISelectItem**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UISelectItems**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UISelectMany**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UISelectOne**

The standard component type for this component.

**COMPONENT_TYPE** - Static variable in class javax.faces.component.**UIViewRoot**

The standard component type for this component.

**CompositeELResolver** - Class in [javax.el](https://docs.oracle.com/en/java/javase/11/docs/api/javax/el/CompositeELResolver.html)

Maintains an ordered composite list of child **ELResolver**s.
**CompositeELResolver()** - Constructor for class javax.el.CompositeELResolver

**Concept** - Interface in javax.xml.registry.infomodel
   The Concept interface is used to represent taxonomy elements and their structural relationship with each other in order to describe an internal taxonomy.

**CONCEPT** - Static variable in interface javax.xml.registry.LifeCycleManager

**ConcurrentAccessException** - Exception in javax.ejb
   A ConcurrentAccessException indicates that the client has attempted an invocation on a stateful session bean while another invocation is in progress.

**ConcurrentAccessException()** - Constructor for exception javax.ejb.ConcurrentAccessException
   Constructs an ConcurrentAccessException with no detail message.

**ConcurrentAccessException(String)** - Constructor for exception javax.ejb.ConcurrentAccessException
   Constructs an ConcurrentAccessException with the specified detailed message.

**ConcurrentAccessException(String, Exception)** - Constructor for exception javax.ejb.ConcurrentAccessException
   Constructs an ConcurrentAccessException with the specified detail message and a nested exception.

**CONFIG** - Static variable in class javax.servlet.jsp.PageContext
   Name used to store ServletConfig in PageContext name table.

**CONFIG_FILES_ATTR** - Static variable in class javax.faces.webapp.FacesServlet
   Context initialization parameter name for a comma delimited list of context-relative resource paths (in addition to /WEB-INF/faces-config.xml which is loaded automatically if it exists) containing JavaServer Faces configuration information.

**ConfigurationException** - Exception in javax.enterprise.deploy.spi.exceptions
   This exception reports errors in generating a configuration bean.

**ConfigurationException()** - Constructor for exception javax.enterprise.deploy.spi.exceptionsConfigurationException
   Creates new ConfigurationException without detail message.
**ConfigurationException(String)** - Constructor for exception
java.enterprise.deploy.spi.exceptions.**ConfigurationException**
Constructs an `ConfigurationException` with the specified detail message.

**confirmAssociation(Association)** - Method in interface
javax.xml.registry.**BusinessLifeCycleManager**
Confirms this Association by the User associated with the caller.

**connect()** - Method in class javax.mail.**Service**
A generic connect method that takes no parameters.

**connect(String, String, String)** - Method in class javax.mail.**Service**
Connect to the current host using the specified username and password.

**connect(String, int, String, String)** - Method in class javax.mail.**Service**
Similar to `connect(host, user, password)` except a specific port can be specified.

**Connection** - Interface in **javax.jms**
A `Connection` object is a client's active connection to its JMS provider.

**Connection** - Interface in **javax.resource.cci**
A `Connection` represents an application-level handle that is used by a client to access the underlying physical connection.

**Connection** - Interface in **javax.xml.registry**
This class represents a connection between a JAXR client and a JAXR provider.

**CONNECTION_CLOSED** - Static variable in class
javax.resource.spi.**ConnectionEvent**
Event notification that an application component has closed the connection

**CONNECTION_ERROR_OCCURRED** - Static variable in class
javax.resource.spi.**ConnectionEvent**
Event notification that an error occurred on the connection.

**ConnectionAdapter** - Class in **javax.mail.event**
The adapter which receives connection events.

**ConnectionAdapter()** - Constructor for class
javax.mail.event.**ConnectionAdapter**

**connectionClosed(ConnectionEvent)** - Method in interface
javax.resource.spi.**ConnectionEventListener**
Notifies that an application component has closed the connection.

**ConnectionConsumer** - Interface in javax.jms
For application servers, **Connection** objects provide a special facility for creating a **ConnectionConsumer** (optional).

**connectionErrorOccurred(ConnectionEvent)** - Method in interface javax.resource.spi.**ConnectionEventListener**
Notifies a connection related error.

**ConnectionEvent** - Class in *javax.mail.event*
This class models Connection events.

**ConnectionEvent(Object, int)** - Constructor for class javax.mail.event.**ConnectionEvent**
Constructor

**ConnectionEvent** - Class in *javax.resource.spi*
The **ConnectionEvent** class provides information about the source of a connection related event. A **ConnectionEvent** instance contains the following information: Type of the connection event ManagedConnection instance that generated the connection event.

**ConnectionEvent(ManagedConnection, int)** - Constructor for class javax.resource.spi.**ConnectionEvent**
Construct a **ConnectionEvent** object.

**ConnectionEvent(ManagedConnection, int, Exception)** - Constructor for class javax.resource.spi.**ConnectionEvent**
Construct a **ConnectionEvent** object.

**ConnectionEventListener** - Interface in *javax.resource.spi*
The **ConnectionEventListener** interface provides an event callback mechanism to enable an application server to receive notifications from a **ManagedConnection** instance.

**ConnectionFactory** - Interface in *javax.jms*
A **ConnectionFactory** object encapsulates a set of connection configuration parameters that has been defined by an administrator.

**ConnectionFactory** - Interface in *javax.resource.cci*
**ConnectionFactory** provides an interface for getting connection to an EIS instance.

**ConnectionFactory** - Class in *javax.xml.registry*
This is the abstract base class for factory classes for creating a JAXR connection.

**ConnectionFactory()** - Constructor for class javax.xml.registry.**ConnectionFactory**
**ConnectionListener** - Interface in `javax.mail.event`
This is the Listener interface for Connection events.

**ConnectionManager** - Interface in `javax.resource.spi`
ConnectionManager interface provides a hook for the resource adapter to pass a connection request to the application server.

**ConnectionMetaData** - Interface in `javax.jms`
A ConnectionMetaData object provides information describing the Connection object.

**ConnectionMetaData** - Interface in `javax.resource.cci`
The interface ConnectionMetaData provides information about an EIS instance connected through a Connection instance.

**ConnectionRequestInfo** - Interface in `javax.resource.spi`
The ConnectionRequestInfo interface enables a resource adapter to pass its own request specific data structure across the connection request flow.

**ConnectionSpec** - Interface in `javax.resource.cci`
ConnectionSpec is used by an application component to pass connection request-specific properties to the ConnectionFactory.

**contains(FetchProfile.Item)** - Method in class `javax.mail.FetchProfile`
Returns true if the fetch profile contains given special item.

**contains(String)** - Method in class `javax.mail.FetchProfile`
Returns true if the fetch profile contains given header name.

**contains(Flags.Flag)** - Method in class `javax.mail.Flags`
Check whether the specified system flag is present in this Flags object.

**contains(String)** - Method in class `javax.mail.Flags`
Check whether the specified user flag is present in this Flags object.

**contains(Flags)** - Method in class `javax.mail.Flags`
Check whether all the flags in the specified Flags object are present in this Flags object.

**contains(Object)** - Method in interface `javax.persistence.EntityManager`
Check if the instance belongs to the current persistence context.

**containsHeader(String)** - Method in interface `javax.servlet.http.HttpServletResponse`
Returns a boolean indicating whether the named response header has already been set.

**containsHeader(String)** - Method in class `javax.servlet.http.HttpServletResponseWrapper`
The default behavior of this method is to call `containsHeader(String name)` on the wrapped response object.

- **containsProperty(String)**: Method in interface `javax.xml.rpc.handler.MessageContext`  
  Returns true if the `MessageContext` contains a property with the specified name.

- **content**: Variable in class `javax.mail.internet.MimeBodyPart`  
  Byte array that holds the bytes of the content of this Part.

- **content**: Variable in class `javax.mail.internet.MimeMessage`  
  Byte array that holds the bytes of this Message's content.

- **CONTENT_INFO**: Static variable in class `javax.mail.FetchProfile.Item`  
  This item is for fetching information about the content of the message.

- **ContentDisposition**: Class in `javax.mail.internet`  
  This class represents a MIME ContentDisposition value.

  - **ContentDisposition()**: Constructor for class `javax.mail.internet.ContentDisposition`  
    No-arg Constructor.

  - **ContentDisposition(String, ParameterList)**: Constructor for class `javax.mail.internet.ContentDisposition`  
    Constructor.

  - **ContentDisposition(String)**: Constructor for class `javax.mail.internet.ContentDisposition`  
    Constructor that takes a ContentDisposition string.

- **contentStream**: Variable in class `javax.mail.internet.MimeBodyPart`  
  If the data for this body part was supplied by an InputStream that implements the SharedInputStream interface, `contentStream` is another such stream representing the content of this body part.

- **contentStream**: Variable in class `javax.mail.internet.MimeMessage`  
  If the data for this message was supplied by an InputStream that implements the SharedInputStream interface, `contentStream` is another such stream representing the content of this message.

- **ContentType**: Class in `javax.mail.internet`  
  This class represents a MIME ContentType value.

  - **ContentType()**: Constructor for class `javax.mail.internet.ContentType`  
    No-arg Constructor.

  - **ContentType(String, String, ParameterList)**: Constructor for class `javax.mail.internet.ContentType`  
    Constructor.
**ContentType(String)** - Constructor for class `javax.mail.internet.ContentType`  
Constructors that takes a Content-Type string.

**contentType** - Variable in class `javax.mail.Multipart`  
This field specifies the content-type of this multipart object.

**ContextCallback** - Interface in `javax.faces.component`  
A simple callback interface that enables taking action on a specific `UIComponent` (either facet or child) in the view while preserving any contextual state for that component instance in the view.

**contextCreated(ELContextEvent)** - Method in interface `javax.el.ELContextListener`  
Invoked when a new `ELContext` has been created.

**contextDestroyed(ServletContextEvent)** - Method in interface `javax.servlet.ServletContextListener`  
Notification that the servlet context is about to be shut down.

**contextInitialized(ServletContextEvent)** - Method in interface `javax.servlet.ServletContextListener`  
Notification that the web application initialization process is starting.

**CONVERSION_MESSAGE_ID** - Static variable in class `javax.faces.component.UIInput`  
The message identifier of the `FacesMessage` to be created if a conversion error occurs, and neither the page author nor the `ConverterException` provides a message.

**convertClientId(FacesContext, String)** - Method in class `javax.faces.render.Renderer`  
Convert the component generated client id to a form suitable for transmission to the client.

**Converter** - Interface in `javax.faces.convert`  
**Converter** is an interface describing a Java class that can perform Object-to-String and String-to-Object conversions between model data objects and a String representation of those objects that is suitable for rendering.

**CONVERTER_ID** - Static variable in class `javax.faces.convert.BigDecimalConverter`  
The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class `javax.faces.convert.BigIntegerConverter`  
The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**BooleanConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**ByteConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**CharacterConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**DateTimeConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**DoubleConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**EnumConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**FloatConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**IntegerConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**LongConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**NumberConverter**
   The standard converter id for this converter.

**CONVERTER_ID** - Static variable in class
javax.faces.convert.**ShortConverter**
   The standard converter id for this converter.

**ConverterELTag** - Class in javax.faces.webapp
   **ConverterELTag** is a base class for all JSP custom actions that
create and register a Converter instance on the **ValueHolder**
associated with our most immediate surrounding instance of a tag
whose implementation class is a subclass of
**UIComponentClassicTagBase**.

**ConverterELTag()** - Constructor for class
**ConverterException** - Exception in *javax.faces.convert*

*ConverterException* is an exception thrown by the `getAsObject()` or `getAsText()` method of a *Converter*, to indicate that the requested conversion cannot be performed.

**ConverterException()** - Constructor for exception

`javax.faces.convert.ConverterException` - Constructor for exception

Construct a new exception with no detail message or root cause.

**ConverterException(String)** - Constructor for exception

`javax.faces.convert.ConverterException(String)` - Constructor for exception

Construct a new exception with the specified detail message and no root cause.

**ConverterException(Throwables)** - Constructor for exception

`javax.faces.convert.ConverterException(Throwables)` - Constructor for exception

Construct a new exception with the specified root cause.

**ConverterException(String, Throwable)** - Constructor for exception

`javax.faces.convert.ConverterException(String, Throwable)` - Constructor for exception

Construct a new exception with the specified detail message and root cause.

**ConverterException(FacesMessage)** - Constructor for exception

`javax.faces.convert.ConverterException(FacesMessage)` - Constructor for exception

Construct a new exception with the specified detail message and no root cause.

**ConverterException(FacesMessage, Throwable)** - Constructor for exception

`javax.faces.convert.ConverterException(FacesMessage, Throwable)` - Constructor for exception

Construct a new exception with the specified detail message and root cause.

**ConverterTag** - Class in *javax.faces.webapp*

*Deprecated.* *This has been partially replaced by ConverterELTag.*

The remainder of the functionality, namely, the binding facility and the implementation of the *ConverterTag.createConverter()* method, is now an implementation detail.

**ConverterTag()** - Constructor for class *javax.faces.webapp ConverterTag*

*Deprecated.*

**Cookie** - Class in *javax.servlet.http*

Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server.
**Cookie(String, String)** - Constructor for class javax.servlet.http.Cookie
Constructs a cookie with a specified name and value.

**copyMessages(Message[], Folder)** - Method in class javax.mail.Folder
Copy the specified Messages from this Folder into another Folder.

**countAttachments()** - Method in class javax.xml.soap.SOAPMessage
Gets a count of the number of attachments in this message.

**CountStatistic** - Interface in javax.management.j2ee.statistics
Specifies standard count measurements.

**create(int)** - Method in class javax.mail.Folder
Create this folder on the Store.

**create()** - Method in interface javax.management.j2ee.ManagementHome
Creates an MEJB session object which provides access to the J2EE Management Model.

**create(Name)** - Method in class javax.xml.soap.SOAPElementFactory
*Deprecated. Use*  
javax.xml.soap.SOAPFactory.createElement(javax.xml.soap.Name)  
*instead*

**create(String)** - Method in class javax.xml.soap.SOAPElementFactory
*Deprecated. Use*  
javax.xml.soap.SOAPFactory.createElement(String localName)  
*instead*

**create(String, String, String)** - Method in class javax.xml.soap.SOAPElementFactory
*Deprecated. Use*  
javax.xml.soap.SOAPFactory.createElement(String localName,  
String prefix, String uri)  
*instead*

**create(Object)** - Static method in class javax.xml.ws.Endpoint
Creates an endpoint with the specified implementor object.

**create(String, Object)** - Static method in class javax.xml.ws.Endpoint
Creates an endpoint with the specified binding type and implementor object.

**create(URL, QName)** - Static method in class javax.xml.ws.Service
Create a Service instance.

**create(QName)** - Static method in class javax.xml.ws.Service
Create a Service instance.

**createAndPublishEndpoint(String, Object)** - Method in class javax.xml.ws.spi.Provider
Creates and publishes an endpoint object with the specified address.
and implementation object.

**createAssociation(RegistryObject, Concept)** - Method in interface javax.xml.registry.LifecycleManager

  Creates an Association instance using the specified parameters.

**createAttachmentPart()** - Method in class javax.xml.soap.SOAPMessage

  Creates a new empty AttachmentPart object.

**createAttachmentPart(DataHandler)** - Method in class javax.xml.soap.SOAPMessage

  Creates an AttachmentPart object and populates it using the given DataHandler object.

**createAttachmentPart(Object, String)** - Method in class javax.xml.soap.SOAPMessage

  Creates an AttachmentPart object and populates it with the specified data of the specified content type.

**createAttribute(String, String, String, String)** - Method in class javax.xml.stream.XMLEventFactory

  Create a new Attribute

**createAttribute(String, String)** - Method in class javax.xml.stream.XMLEventFactory

  Create a new Attribute

**createAttribute(QName, String)** - Method in class javax.xml.stream.XMLEventFactory

  Create a new Attribute

**createBinder(Class<T>)** - Method in class javax.xml.bind.JAXBContext

  Creates a Binder object that can be used for associative/in-place unmarshalling/marshalling.

**createBinder()** - Method in class javax.xml.bind.JAXBContext

  Creates a Binder for W3C DOM.

**createBrowser(Queue)** - Method in interface javax.jms.QueueSession

  Creates a QueueBrowser object to peek at the messages on the specified queue.

**createBrowser(Queue, String)** - Method in interface javax.jms.QueueSession

  Creates a QueueBrowser object to peek at the messages on the specified queue using a message selector.

**createBrowser(Queue)** - Method in interface javax.jms.Session

  Creates a QueueBrowser object to peek at the messages on the specified queue.
**createBrowser(Queue, String)** - Method in interface `javax.jms.Session`
   Creates a `QueueBrowser` object to peek at the messages on the specified queue using a message selector.

**createBytesMessage()** - Method in interface `javax.jms.Session`
   Creates a `BytesMessage` object.

**createCall(QName)** - Method in interface `javax.xml.rpc.Service`
   Creates a `Call` instance.

**createCall(QName, QName)** - Method in interface `javax.xml.rpc.Service`
   Creates a `Call` instance.

**createCall(QName, String)** - Method in interface `javax.xml.rpc.Service`
   Creates a `Call` instance.

**createCall()** - Method in interface `javax.xml.rpc.Service`
   Creates a `Call` object not associated with specific operation or target service endpoint.

**createCData(String)** - Method in class `javax.xml.stream.XMLEventFactory`
   Create a Characters event with the CData flag set to true

**createCharacters(String)** - Method in class `javax.xml.stream.XMLEventFactory`
   Create a Characters event, this method does not check if the content is all whitespace.

**createClassification(ClassificationScheme, String, String)** - Method in interface `javax.xml.registry.LifeCycleManager`
   Creates a Classification instance for an external Classification using the specified String name and String value that identify a taxonomy element within the specified ClassificationScheme.

**createClassification(ClassificationScheme, InternationalString, String)** - Method in interface `javax.xml.registry.LifeCycleManager`
   Creates a Classification instance for an external Classification using the specified InternationalString name and String value that identify a taxonomy element within the specified ClassificationScheme.

**createClassification(Concept)** - Method in interface `javax.xml.registry.LifeCycleManager`
   Creates a Classification instance for an internal Classification using the specified Concept that identifies a taxonomy element within an internal ClassificationScheme.

**createClassificationScheme(String, String)** - Method in interface `javax.xml.registry.LifeCycleManager`
   Creates a ClassificationScheme given the specified String
createClassificationScheme(InternationalString, InternationalString) - Method in interface javax.xml.registry.LifeCycleManager
   Creates a ClassificationScheme given the specified InternationalString parameters.
createClassificationScheme(Concept) - Method in interface javax.xml.registry.LifeCycleManager
   Creates a ClassificationScheme from a Concept that has no ClassificationScheme or parent Concept.
createComment(String) - Method in class javax.xml.stream.XMLEventFactory
   Create a comment
createComponent(String) - Method in class javax.faces.application.Application
   Instantiate and return a new UIComponent instance of the class specified by a previous call to addComponent() for the specified component type.
createComponent(ValueBinding, FacesContext, String) - Method in class javax.faces.application.Application
   Deprecated. This has been replaced by Application.createComponent(javax.el.ValueExpression, javax.faces.FacesContext, String)
createComponent(ValueExpression, FacesContext, String) - Method in class javax.faces.application.Application
   Call the getValue() method on the specified ValueExpression.
createComponent(FacesContext, String) - Method in class javax.faces.webapp.UIComponentClassicTagBase
   Create and return a new child component of the type returned by calling getComponentType().
createComponent(FacesContext, String) - Method in class javax.faces.webapp.UIComponentELTag
   Create and return a new child component of the type returned by calling getComponentType().
createComponent(FacesContext, String) - Method in class javax.faces.webapp.UIComponentTag
   Deprecated. Implement createComponent using Faces 1.1 EL API.
createConcept(RegistryObject, String, String) - Method in interface javax.xml.registry.LifeCycleManager
   Creates a Concept instance using the specified parameters, where the name is a String.
createConcept(RegistryObject, InternationalString, String) - Method in interface javax.xml.registry.LifeCycleManager
Creates a Concept instance using the specified parameters, where the name is an InternationalString.

createConfiguration(DeployableObject) - Method in interface javax.enterprise.deploy.spi.DeploymentManager
Retrieve the object that provides server-specific deployment configuration information for the J2EE deployable component.

createConnection() - Method in interface javax.jms.ConnectionFactory
Creates a connection with the default user identity.

createConnection(String, String) - Method in interface javax.jms.ConnectionFactory
Creates a connection with the specified user identity.

createConnection() - Method in class javax.xml.registry.ConnectionFactory
Create a named connection.

createConnection() - Method in class javax.xml.soap.SOAPConnectionFactory
Create a new SOAPConnection.

createConnectionConsumer(Destination, String, ServerSessionPool, int) - Method in interface javax.jms.Connection
Creates a connection consumer for this connection (optional operation).

createConnectionConsumer(Queue, String, ServerSessionPool, int) - Method in interface javax.jms.QueueConnection
Creates a connection consumer for this connection (optional operation).

createConnectionConsumer(Topic, String, ServerSessionPool, int) - Method in interface javax.jms.TopicConnection
Creates a connection consumer for this connection (optional operation).

createConnectionFactory(ConnectionManager) - Method in interface javax.resource.spi.ManagedConnectionFactory
Creates a Connection Factory instance.

createConnectionFactory() - Method in interface javax.resource.spi.ManagedConnectionFactory
Creates a Connection Factory instance.

createConsumer(Destination) - Method in interface javax.jms.Session
Creates a MessageConsumer for the specified destination.
**createConsumer(Destination, String)** - Method in interface javax.jms.Session

Creates a MessageConsumer for the specified destination, using a message selector.

**createConsumer(Destination, String, boolean)** - Method in interface javax.jms.Session

Creates MessageConsumer for the specified destination, using a message selector.

**createContainerEntityManagerFactory(PersistenceUnitInfo, Map)** - Method in interface javax.persistence.spi.PersistenceProvider

Called by the container when an EntityManagerFactory is to be created.

**createConverter(String)** - Method in class javax.faces.application.Application

Instantiate and return a new Converter instance of the class specified by a previous call to addConverter() for the specified converter id.

**createConverter(Class)** - Method in class javax.faces.application.Application

Instantiate and return a new Converter instance of the class that has registered itself as capable of performing conversions for objects of the specified type.

**createConverter()** - Method in class javax.faces.webapp.ConverterELTag

Create and return a new Converter to be registered on our surrounding UICOMPONENT.

**createConverter()** - Method in class javax.faces.webapp.ConverterTag

Deprecated. Create and return a new Converter to be registered on our surrounding UICOMPONENT.

**CREATED** - Static variable in class javax.mail.event.FolderEvent

The folder was created.

**createDataContentHandler(String)** - Method in class javax.activation.CommandMap

Locate a DataContentHandler that corresponds to the MIME type.

**createDataContentHandler(String, DataSource)** - Method in class javax.activation.CommandMap

Locate a DataContentHandler that corresponds to the MIME type.

**createDataContentHandler(String)** - Method in interface javax.activation.DataContentHandlerFactory
Creates a new DataContentHandler object for the MIME type.

**createDataContentHandler(String)** - Method in class javax.activation.MailcapCommandMap

Return the DataContentHandler for the specified MIME type.

**createDetail()** - Method in class javax.xml.soap.SOAPFactory

Creates a new Detail object which serves as a container for DetailEntry objects.

**createDispatch(QName, Class<T>, Service.Mode)** - Method in class javax.xml.ws.Service

 Creates a Dispatch instance for use with objects of the users choosing.

**createDispatch(QName, JAXBContext, Service.Mode)** - Method in class javax.xml.ws.Service

 Creates a Dispatch instance for use with JAXB generated objects.

**createDispatch(QName, Class<T>, Service.Mode)** - Method in class javax.xml.ws.spi.ServiceDelegate

 Creates a Dispatch instance for use with objects of the users choosing.

**createDispatch(QName, JAXBContext, Service.Mode)** - Method in class javax.xml.ws.spi.ServiceDelegate

 Creates a Dispatch instance for use with JAXB generated objects.

**createDTD(String)** - Method in class javax.xml.stream.XMLEventFactory

Create a document type definition event This string contains the entire document type declaration that matches the doctypedecl in the XML 1.0 specification

**createDurableConnectionConsumer(Topic, String, String, ServerSessionPool, int)** - Method in interface javax.jms.Connection

Create a durable connection consumer for this connection (optional operation).

**createDurableConnectionConsumer(Topic, String, String, ServerSessionPool, int)** - Method in interface javax.jms.TopicConnection

Create a durable connection consumer for this connection (optional operation).

**createDurableSubscriber(Topic, String)** - Method in interface javax.jms.Session

Create a durable subscriber to the specified topic.

**createDurableSubscriber(Topic, String, String, boolean)** - Method in interface javax.jms.Session

Creates a durable subscriber to the specified topic, using a message selector and specifying whether messages published by its own connection should be delivered to it.

**createDurableSubscriber(Topic, String)** - Method in interface javax.jms.TopicSession
- Creates a durable subscriber to the specified topic.

**createDurableSubscriber(Topic, String, String, boolean)** - Method in interface javax.jms.TopicSession
- Creates a durable subscriber to the specified topic, using a message selector or specifying whether messages published by its own connection should be delivered to it.

**createElement(Element)** - Method in class javax.xml.soap.SOAPFactory
- Creates a SOAPElement object from an existing DOM Element.

**createElement(Name)** - Method in class javax.xml.soap.SOAPFactory
- Creates a SOAPElement object initialized with the given Name object.

**createElement(QName)** - Method in class javax.xml.soap.SOAPFactory
- Creates a SOAPElement object initialized with the given QName object.

**createElement(String)** - Method in class javax.xml.soap.SOAPFactory
- Creates a new SOAPElement object initialized with the given local name.

**createElement(String, String, String)** - Method in class javax.xml.soap.SOAPFactory
- Creates a new SOAPElement object with the given local name, prefix and uri.

**createEmailAddress(String)** - Method in interface javax.xml.registry.LifeCycleManager
- Creates an EmailAddress instance using an address as the parameter.

**createEmailAddress(String, String)** - Method in interface javax.xml.registry.LifeCycleManager
- Creates an EmailAddress instance using both an address and a type as parameters.

**createEndDocument()** - Method in class javax.xml.stream.XMLEventFactory
- Creates a new instance of an EndDocument event

**createEndElement(QName, Iterator)** - Method in class javax.xml.stream.XMLEventFactory
- Create a new EndElement

**createEndElement(String, String, String)** - Method in class javax.xml.stream.XMLEventFactory
Create a new EndElement
createEndElement(String, String, String, Iterator) - Method in class javax.xml.stream.XMLEventFactory
    Create a new EndElement
createEndpoint(XAResource) - Method in interface javax.resource.spi.endpoint.MessageEndpointFactory
    This is used to create a message endpoint.
createEndpoint(String, Object) - Method in class javax.xml.ws.spi.Provider
    Creates an endpoint object with the provided binding and implementation object.
createEntityManager() - Method in interface javax.persistence.EntityManagerFactory
    Create a new EntityManager.
createEntityManager(Map) - Method in interface javax.persistence.EntityManagerFactory
    Create a new EntityManager with the specified Map of properties.
createEntityManagerFactory(String) - Static method in class javax.persistence.Persistence
    Create and return an EntityManagerFactory for the named persistence unit.
createEntityManagerFactory(String, Map) - Static method in class javax.persistence.Persistence
    Create and return an EntityManagerFactory for the named persistence unit using the given properties.
createEntityManagerFactory(String, Map) - Method in interface javax.persistence.spi.PersistenceProvider
    Called by Persistence class when an EntityManagerFactory is to be created.
createEntityReference(String, EntityDeclaration) - Method in class javax.xml.stream.XMLEventFactory
    Creates a new instance of a EntityReference event
CreateException - Exception in javax.ejb
    The CreateException exception must be included in the throws clauses of all create methods defined in an enterprise Bean's home interface.
CreateException() - Constructor for exception javax.ejb.CreateException
    Constructs a CreateException with no detail message.
CreateException(String) - Constructor for exception
javax.ejb.**CreateException**
Constructs a CreateException with the specified detail message.

**createExternalIdentifier(ClassificationScheme, String, String)** - Method in interface javax.xml.registry.LifeCycleManager
Creates an ExternalIdentifier instance using the specified parameters, where the name is a String.

**createExternalIdentifier(ClassificationScheme, InternationalString, String)** - Method in interface javax.xml.registry.LifeCycleManager
Creates an ExternalIdentifier instance using the specified parameters, where the name is an InternationalString.

**createExternalLink(String, String)** - Method in interface javax.xml.registry.LifeCycleManager
Creates an ExternalLink instance using the specified parameters, where the description is a String.

**createExternalLink(String, InternationalString)** - Method in interface javax.xml.registry.LifeCycleManager
Creates an ExternalLink instance using the specified parameters, where the description is an InternationalString.

**createExtrinsicObject(DataHandler)** - Method in interface javax.xml.registry.LifeCycleManager
Creates an ExtrinsicObject instance using the specified parameters.

**createFault(String, QName)** - Method in class javax.xml.soap.SOAPFactory
Creates a new SOAPFault object initialized with the given reasonText and faultCode

**createFault()** - Method in class javax.xml.soap.SOAPFactory
Creates a new default SOAPFault object

**createFederatedConnection(Collection)** - Method in class javax.xml.registry.ConnectionFactory
Create a FederatedConnection.

**createFilteredReader(XMLStreamReader, StreamFilter)** - Method in class javax.xml.stream.XMLInputFactory
Create a filtered reader that wraps the filter around the reader

**createFilteredReader(XMLEventReader, EventFilter)** - Method in class javax.xml.stream.XMLInputFactory
Create a filtered event reader that wraps the filter around the event reader

**createIgnorableSpace(String)** - Method in class javax.xml.stream.XMLEventFactory
Create an ignorable space

**createIndexedRecord(String)** - Method in interface
javax.resource.cci.**RecordFactory**
creates a IndexedRecord.

**createInteraction()** - Method in interface javax.resource.cci.**Connection**
creates an Interaction associated with this Connection.

**createInternationalString()** - Method in interface
javax.xml.registry.**LifeCycleManager**
creates an empty InternationalString instance.

**createInternationalString(String)** - Method in interface
ejavax.xml.registry.**LifeCycleManager**
creates an InternationalString instance using a String parameter and the default Locale.

**createInternationalString(Locale, String)** - Method in interface
ejavax.xml.registry.**LifeCycleManager**
creates an InternationalString instance using the specified Locale and String parameters.

**createInternetHeaders(InputStream)** - Method in class
javax.mail.internet.**MimeMessage**
create and return an InternetHeaders object that loads the headers from the given InputStream.

**createInternetHeaders(InputStream)** - Method in class
javax.mail.internet.**MimeMultipart**
create and return an InternetHeaders object that loads the headers from the given InputStream.

**createJAXBIntrospector()** - Method in class
javax.xml.bind.**JAXBContext**
create a JAXBIntrospector object that can be used to introspect JAXB objects.

**createKey(String)** - Method in interface
javax.xml.registry.**LifeCycleManager**
creates a Key instance from an ID.

**createLocalizedString(Locale, String)** - Method in interface
javax.xml.registry.**LifeCycleManager**
creates aLocalizedString instance using the specified Locale and String parameters.

**createLocalizedString(Locale, String, String)** - Method in interface
javax.xml.registry.**LifeCycleManager**
creates aLocalizedString instance using the specified Locale,
String, and character set parameters.

**createManagedConnection(Subject, ConnectionRequestInfo)** - Method in interface javax.resource.spi.**ManagedConnectionFactory**

    Creates a new physical connection to the underlying EIS resource manager.

**createMapMessage()** - Method in interface javax.jms.**Session**

    Creates a MapMessage object.

**createMappedRecord(String)** - Method in interface javax.resource.cci.**RecordFactory**

    Creates a MappedRecord.

**createMarshaller()** - Method in class javax.xml.bind.**JAXBContext**

    Create a Marshaller object that can be used to convert a java content tree into XML data.

**createMessage()** - Method in interface javax.jms.**Session**

    Creates a Message object.

**createMessage()** - Method in class javax.xml.soap.**MessageFactory**

    Creates a new SOAPMessage object with the default SOAPPart, SOAPEnvelope, SOAPBody, and SOAPHeader objects.

**createMessage(MimeHeaders, InputStream)** - Method in class javax.xml.soap.**MessageFactory**

    Internalizes the contents of the given InputStream object into a new SOAPMessage object and returns the SOAPMessage object.

**createMethodBinding(String, Class[])** - Method in class javax.faces.application.**Application**

    **Deprecated.** *This has been replaced by calling*

        Application.getExpressionFactory() then
        ExpressionFactory.createMethodExpression(javax.el.ELContext, java.lang.String, java.lang.Class, java.lang.Class[]).

**createMethodExpression(ELContext, String, Class<?>, Class<?>[])** - Method in class javax.el.**ExpressionFactory**

    Parses an expression into a MethodExpression for later evaluation.

**createMimeBodyPart(InternetHeaders, byte[])** - Method in class javax.mail.internet.**MimeMultipart**

    Create and return a MimeBodyPart object to represent a body part parsed from the InputStream.

**createMimeBodyPart(InputStream)** - Method in class javax.mail.internet.**MimeMultipart**

    Create and return a MimeBodyPart object to represent a body part parsed from the InputStream.
**createMimeMessage(Session)** - Method in class javax.mail.internet.MimeMessage
Create and return a MimeMessage object.

**createName(String, String, String)** - Method in interface javax.xml.soap.SOAPEnvelope
Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.

**createName(String)** - Method in interface javax.xml.soap.SOAPEnvelope
Creates a new Name object initialized with the given local name.

**createName(String, String, String)** - Method in class javax.xml.soap.SOAPFactory
Creates a new Name object initialized with the given local name, namespace prefix, and namespace URI.

**createName(String)** - Method in class javax.xml.soap.SOAPFactory
Creates a new Name object initialized with the given local name.

**createNamedQuery(String)** - Method in interface javax.persistence.EntityManager
Create an instance of Query for executing a named query (in the Java Persistence query language or in native SQL).

**createNamespace(String)** - Method in class javax.xml.stream.XMLEventFactory
Create a new default Namespace

**createNamespace(String, String)** - Method in class javax.xml.stream.XMLEventFactory
Create a new Namespace

**createNativeQuery(String)** - Method in interface javax.persistence.EntityManager
Create an instance of Query for executing a native SQL statement, e.g., for update or delete.

**createNativeQuery(String, Class)** - Method in interface javax.persistence.EntityManager
Create an instance of Query for executing a native SQL query.

**createNativeQuery(String, String)** - Method in interface javax.persistence.EntityManager
Create an instance of Query for executing a native SQL query.

**createObject(String)** - Method in interface javax.xml.registry.LifecycleManager
Creates instances of information model interfaces (factory method).
**createObjectMessage()** - Method in interface `javax.jms.Session`
  Creates an `ObjectMessage` object.

**createObjectMessage(Serializable)** - Method in interface `javax.jms.Session`
  Creates an initialized `ObjectMessage` object.

**createOrganization(String)** - Method in interface `javax.xml.registry.LifeCycleManager`
  Creates an Organization instance using the specified name, where the name is a String.

**createOrganization(InternationalString)** - Method in interface `javax.xml.registry.LifeCycleManager`
  Creates an Organization instance using the specified name, where the name is an `InternationalString`.

**createOutput(String, String)** - Method in class `javax.xml.bind.SchemaOutputResolver`
  Decides where the schema file (of the given namespace URI) will be written, and return it as a `Result` object.

**createPersonName(String, String, String)** - Method in interface `javax.xml.registry.LifeCycleManager`
  Creates a `PersonName` instance using the specified first, middle, and last names.

**createPersonName(String)** - Method in interface `javax.xml.registry.LifeCycleManager`
  Creates a `PersonName` instance using the specified full name.

**createPostalAddress(String, String, String, String, String, String, String)** - Method in interface `javax.xml.registry.LifeCycleManager`
  Creates a `PostalAddress` instance using the specified parameters.

**createProcessingInstruction(String, String)** - Method in class `javax.xml.stream.XMLEventFactory`
  Create a processing instruction

**createProducer(Destination)** - Method in interface `javax.jms.Session`
  Creates a `MessageProducer` to send messages to the specified destination.

**createPublisher(Topic)** - Method in interface `javax.jms.TopicSession`
  Creates a publisher for the specified topic.

**createQName(String, String)** - Method in interface `javax.xml.soap.SOAPElement`
  Creates a `QName` whose namespace URI is the one associated with the parameter, prefix, in the context of this `SOAPElement`. 
**createQuery(String)** - Method in interface javax.persistence.EntityManager
Create an instance of Query for executing a Java Persistence query language statement.

**createQuery(int, String)** - Method in interface javax.xml.registry.DeclarativeQueryManager
Creates a Query object given a queryType (for example, QUERY_TYPE_SQL) and a String that represents a query in the syntax appropriate for queryType.

**createQueue(String)** - Method in interface javax.jms.QueueSession
Creates a queue identity given a queue name.

**createQueue(String)** - Method in interface javax.jms.Session
Creates a queue identity given a queue name.

**createQueueConnection()** - Method in interface javax.jms.QueueConnectionFactory
Creates a queue connection with the default user identity.

**createQueueConnection(String, String)** - Method in interface javax.jms.QueueConnectionFactory
Creates a queue connection with the specified user identity.

**createQueueSession(boolean, int)** - Method in interface javax.jms.QueueConnection
Creates a QueueSession object.

**createQueueSession(boolean, int)** - Method in interface javax.jms.XAQueueConnection
Creates an XAQueueSession object.

**createReceiver(Queue)** - Method in interface javax.jms.QueueSession
Creates a QueueReceiver object to receive messages from the specified queue.

**createReceiver(Queue, String)** - Method in interface javax.jms.QueueSession
Creates a QueueReceiver object to receive messages from the specified queue using a message selector.

**createRegistryPackage(String)** - Method in interface javax.xml.registry.LifecycleManager
Creates a RegistryPackage instance using the specified name, where the name is a String.

**createRegistryPackage(InternationalString)** - Method in interface javax.xml.registry.LifecycleManager
Creates a RegistryPackage instance using the specified name,
where the name is an InternationalString.

**createResponseStream(OutputStream)** - Method in class javax.faces.render.RenderKit

Use the provided OutputStream to create a new ResponseStream instance.

**createResponseWriter(Writer, String, String)** - Method in class javax.faces.render.RenderKit

Use the provided Writer to create a new ResponseWriter instance for the specified (optional) content type, and character encoding.

**createSender(Queue)** - Method in interface javax.jms.QueueSession

Creates a QueueSender object to send messages to the specified queue.

**createService(String)** - Method in interface javax.xml.registry.LifeCycleManager

Creates a Service instance using the specified name, where the name is a String.

**createService(InternationalString)** - Method in interface javax.xml.registry.LifeCycleManager

Creates a Service instance using the specified name, where the name is an InternationalString.

**createService(URL, QName)** - Method in class javax.xml.rpc.ServiceFactory

Create a Service instance.

**createService(QName)** - Method in class javax.xml.rpc.ServiceFactory

Create a Service instance.

**createServiceBinding()** - Method in interface javax.xml.registry.LifeCycleManager

Creates an empty ServiceBinding instance.

**createServiceDelegate(URL, QName, Class)** - Method in class javax.xml.ws.spi.Provider

Creates a service delegate object.

**createSession(boolean, int)** - Method in interface javax.jms.Connection

Creates a Session object.

**createSession(boolean, int)** - Method in interface javax.jms.XAConnection

Creates an Session object.

**createSlot(String, String, String)** - Method in interface javax.xml.registry.LifeCycleManager

Creates a Slot instance using the specified parameters, where the
value is a String.

**createSlot(String, Collection, String)** - Method in interface javax.xml.registry.LifeCycleManager

Creates a Slot instance using the specified parameters, where the value is a Collection of Strings.

**createSpace(String)** - Method in class javax.xml.stream.XMLEventFactory

Create a Characters event with the isSpace flag set to true

**createSpecificationLink()** - Method in interface javax.xml.registry.LifeCycleManager

Creates an empty SpecificationLink instance.

**createStartDocument()** - Method in class javax.xml.stream.XMLEventFactory

Creates a new instance of a StartDocument event

**createStartDocument(String, String, boolean)** - Method in class javax.xml.stream.XMLEventFactory

Creates a new instance of a StartDocument event

**createStartDocument(String, String)** - Method in class javax.xml.stream.XMLEventFactory

Creates a new instance of a StartDocument event

**createStartDocument(String)** - Method in class javax.xml.stream.XMLEventFactory

Creates a new instance of a StartDocument event

**createStartElement(QName, Iterator, Iterator)** - Method in class javax.xml.stream.XMLEventFactory

Create a new StartElement.

**createStartElement(String, String, String)** - Method in class javax.xml.stream.XMLEventFactory

Create a new StartElement.

**createStartElement(String, String, String, Iterator, Iterator)** - Method in class javax.xml.stream.XMLEventFactory

Create a new StartElement.

**createStartElement(String, String, String, Iterator, Iterator, NamespaceContext)** - Method in class javax.xml.stream.XMLEventFactory

Create a new StartElement.

**createStreamMessage()** - Method in interface javax.jms.Session

Creates a StreamMessage object.

**createSubscriber(Topic)** - Method in interface javax.jms.TopicSession
Creates a nondurable subscriber to the specified topic. 
**createSubscriber(Topic, String, boolean)** - Method in interface 
javax.jms.**TopicSession**
Creates a nondurable subscriber to the specified topic, using a 
message selector or specifying whether messages published by its 
own connection should be delivered to it.

**createTelephoneNumber()** - Method in interface 
javax.xml.registry.**LifeCycleManager**
Creates an empty TelephoneNumber instance.

**createTemporaryQueue()** - Method in interface javax.jms.**QueueSession**
Creates a TemporaryQueue object.

**createTemporaryQueue()** - Method in interface javax.jms.**Session**
Creates a TemporaryQueue object.

**createTemporaryTopic()** - Method in interface javax.jms.**Session**
Creates a TemporaryTopic object.

**createTemporaryTopic()** - Method in interface javax.jms.**TopicSession**
Creates a TemporaryTopic object.

**createTextMessage()** - Method in interface javax.jms.**Session**
Creates a TextMessage object.

**createTextMessage(String)** - Method in interface javax.jms.**Session**
Creates an initialized TextMessage object.

**createTimer(long, Serializable)** - Method in interface javax.ejb.**TimerService**
Create a single-action timer that expires after a specified duration.

**createTimer(long, long, Serializable)** - Method in interface javax.ejb.**TimerService**
Create an interval timer whose first expiration occurs after a 
specified duration, and whose subsequent expirations occur after a 
specified interval.

**createTimer(Date, Serializable)** - Method in interface javax.ejb.**TimerService**
Create a single-action timer that expires at a given point in time.

**createTimer(Date, long, Serializable)** - Method in interface javax.ejb.**TimerService**
Create an interval timer whose first expiration occurs at a given point 
in time and whose subsequent expirations occur after a specified 
interval.

**createTimer()** - Method in interface javax.resource.spi.**BootstrapContext**
Creates a new java.util.Timer instance.
**createTopic(String)** - Method in interface javax.jms.Session
Creates a topic identity given a Topic name.

**createTopic(String)** - Method in interface javax.jms.TopicSession
Creates a topic identity given a Topic name.

**createTopicConnection()** - Method in interface javax.jms.TopicConnectionFactory
Creates a topic connection with the default user identity.

**createTopicConnection(String, String)** - Method in interface javax.jms.TopicConnectionFactory
Creates a topic connection with the specified user identity.

**createTopicSession(boolean, int)** - Method in interface javax.jms.TopicConnection
Creates a TopicSession object.

**createTopicSession(boolean, int)** - Method in interface javax.jms.XATopicConnection
Creates an XATopicSession object.

**createTypeMapping()** - Method in interface javax.xml.rpc.encoding.TypeMappingRegistry
Creates a new empty TypeMapping object.

**createUniqueld()** - Method in class javax.faces.component.UIViewRoot
Generate an identifier for a component.

**createUnmarshalException(SAXException)** - Method in class javax.xml.bind.helpers.AbstractUnmarshallerImpl
Creates an UnmarshalException from a SAXException.

**createUnmarshaller(ValidationEventHandler)** - Method in interface javax.xml.bind.annotation.DomHandler
When a JAXB provider needs to unmarshal a part of a document into an infoset representation, it first calls this method to create a Result object.

**createUnmarshaller(ValidationEventHandler)** - Method in class javax.xml.bind.annotation.W3CDomHandler

**createUnmarshaller()** - Method in class javax.xml.bind.JAXBContext
Create an Unmarshaller object that can be used to convert XML data into a java content tree.

**createUser()** - Method in interface javax.xml.registry.LifeCycleManager
Creates an empty User instance.

**createValidator(String)** - Method in class javax.faces.application.Application
Instantiate and return a new `validator` instance of the class specified by a previous call to `addValidator()` for the specified validator id.

`createValidator()` - Method in class `javax.faces.webapp.ValidatorELTag`
Create and return a new `validator` to be registered on our surrounding `UIComponent`.

`createValidator()` - Method in class `javax.faces.webapp.ValidatorTag`
`Deprecated`. Create and return a new `validator` to be registered on our surrounding `UIComponent`.

`createValidator()` - Method in class `javax.xml.bind.JAXBContext`
`Deprecated. since JAXB2.0`

`createValueBinding(String)` - Method in class `javax.faces.application.Application`
`Deprecated. This has been replaced by calling
`Application.getExpressionFactory()` then`ExpressionFactory.createValueExpression(javax.el.ELContext, java.lang.String, java.lang.Class)`.

`createValueExpression(ELContext, String, Class<?>)` - Method in class `javax.el.ExpressionFactory`
Parses an expression into a `ValueExpression` for later evaluation.

`createValueExpression(Object, Class<?>)` - Method in class `javax.el.ExpressionFactory`
Creates a `ValueExpression` that wraps an object instance.

`createVerbatimComponent()` - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
Use the `Application` instance to create a new component with the following characteristics.

`createVerbatimComponentFromBodyContent()` - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
Create a transient UIOutput component from the body content, of this tag instance or return null if there is no body content, the body content is whitespace, or the body content is a comment.

`createView(FacesContext, String)` - Method in class `javax.faces.application.ViewHandler`
Create and return a new `UIViewRoot` instance initialized with information from the argument `FacesContext` and `viewId`.

`createView(FacesContext, String)` - Method in class `javax.faces.application.ViewHandlerWrapper`
The default behavior of this method is to call
ViewHandler.createView(javax.faces.context.FacesContext, String) on the wrapped ViewHandler object.

createXAConnection() - Method in interface javax.jms.XAConnectionFactory
  Creates an XAConnection with the default user identity.

createXAConnection(String, String) - Method in interface javax.jms.XAConnectionFactory
  Creates anXA connection with the specified user identity.

createXAQueueConnection() - Method in interface javax.jms.XAQueueConnectionFactory
  Creates anXA queue connection with the default user identity.

createXAQueueConnection(String, String) - Method in interface javax.jms.XAQueueConnectionFactory
  Creates anXA queue connection with the specified user identity.

createXAQueueSession() - Method in interface javax.jms.XAQueueConnection
  Creates anXAQueueSession object.

createXASession() - Method in interface java.jms.XAConnection
  Creates anXASession object.

createXATopicConnection() - Method in interface javax.jms.XATopicConnectionFactory
  Creates anXA topic connection with the default user identity.

createXATopicConnection(String, String) - Method in interface javax.jms.XATopicConnectionFactory
  Creates anXA topic connection with the specified user identity.

createXATopicSession() - Method in interface javax.jms.XATopicConnection
  Creates anXATopicSession object.

createXMLEventReader(Reader) - Method in class javax.xml.stream.XMLInputFactory
  Create a new XMLEventReader from a reader

createXMLEventReader(String, Reader) - Method in class javax.xml.stream.XMLInputFactory
  Create a new XMLEventReader from a reader

createXMLEventReader(XMLStreamReader) - Method in class javax.xml.stream.XMLInputFactory
  Create a new XMLEventReader from an XMLStreamReader.

createXMLEventReader(Source) - Method in class javax.xml.stream.XMLInputFactory
Create a new XMLEventReader from a JAXP source.

**createXMLEventReader(InputStream)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLEventReader from a java.io.InputStream

**createXMLEventReader(InputStream, String)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLEventReader from a java.io.InputStream

**createXMLEventReader(String, InputStream)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLEventReader from a java.io.InputStream

**createXMLEventWriter(Result)** - Method in class javax.xml.stream.XMLOutputFactory

Create a new XMLEventWriter that writes to a JAXP result.

**createXMLEventWriter(OutputStream)** - Method in class javax.xml.stream.XMLOutputFactory

Create a new XMLEventWriter that writes to a stream

**createXMLEventWriter(OutputStream, String)** - Method in class javax.xml.stream.XMLOutputFactory

Create a new XMLEventWriter that writes to a stream

**createXMLEventWriter(Writer)** - Method in class javax.xml.stream.XMLOutputFactory

Create a new XMLEventWriter that writes to a writer

**createXMLStreamReader(Reader)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLStreamReader from a reader

**createXMLStreamReader(Source)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLStreamReader from a JAXP source.

**createXMLStreamReader(InputStream)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLStreamReader from a java.io.InputStream

**createXMLStreamReader(InputStream, String)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLStreamReader from a java.io.InputStream

**createXMLStreamReader(String, InputStream)** - Method in class javax.xml.stream.XMLInputFactory

Create a new XMLStreamReader from a java.io.InputStream

**createXMLStreamReader(String, Reader)** - Method in class javax.xml.stream.XMLInputFactory
Create a new XMLStreamReader from a java.io.InputStream
createXMLStreamReader(Writer) - Method in class
javax.xml.stream.XMLOutputFactory
Create a new XMLStreamWriter that writes to a writer
createXMLStreamWriter(OutputStream) - Method in class
javax.xml.stream.XMLOutputFactory
Create a new XMLStreamWriter that writes to a stream
createXMLStreamWriter(OutputStream, String) - Method in class
javax.xml.stream.XMLOutputFactory
Create a new XMLStreamWriter that writes to a stream
createXMLStreamWriter(Result) - Method in class
javax.xml.stream.XMLOutputFactory
Create a new XMLStreamWriter that writes to a JAXP result.
CURRENCY_ID - Static variable in class
javax.faces.convert.NumberConverter
The message identifier of the FacesMessage to be created if the conversion to Number fails.
**DataContentHandler** - Interface in `javax.activation`
- The DataContentHandler interface is implemented by objects that can be used to extend the capabilities of the DataHandler's implementation of the Transferable interface.

**DataContentHandlerFactory** - Interface in `javax.activation`
- This interface defines a factory for DataContentHandlers.

**DataHandler** - Class in `javax.activation`
- The DataHandler class provides a consistent interface to data available in many different sources and formats.

**DataHandler(DataSource)** - Constructor for class `javax.activation.DataHandler`
- Create a DataHandler instance referencing the specified DataSource.

**DataHandler(Object, String)** - Constructor for class `javax.activation.DataHandler`
- Create a DataHandler instance representing an object of this MIME type.

**DataHandler(URL)** - Constructor for class `javax.activation.DataHandler`
- Create a DataHandler instance referencing a URL.

**datalen** - Variable in class `javax.mail.util.SharedFileInputStream`
- The amount of data in this subset of the file.

**DataModel** - Class in `javax.faces.model`
- DataModel is an abstraction around arbitrary data binding technologies that can be used to adapt a variety of data sources for use by JavaServer Faces components that support per-row processing for their child components (such as UIData).

**DataModel()** - Constructor for class `javax.faces.model.DataModel`

**DataModelEvent** - Class in `javax.faces.model`
- DataModelEvent represents an event of interest to registered listeners that occurred on the specified DataModel.

**DataModelEvent(DataModel, int, Object)** - Constructor for class `javax.faces.model.DataModelEvent`
- Construct an event object that is associated with the specified row index and associated data.

**DataModelListener** - Interface in `javax.faces.model`
**DataModelListener** represents an event listener that wishes to be notified of **DataModelEvent**s occurring on a particular **DataModel** instance.

**DataSource** - Interface in **javax.activation**
The DataSource interface provides the JavaBeans Activation Framework with an abstraction of an arbitrary collection of data.

**DatatypeConverter** - Class in **javax.xml.bind**
The javaType binding declaration can be used to customize the binding of an XML schema datatype to a Java datatype.

**DatatypeConverterInterface** - Interface in **javax.xml.bind**
The DatatypeConverterInterface is for JAXB provider use only.

**date** - Variable in class **javax.mail.search**.**DateTime**
The date.

**DATE_ID** - Static variable in class **javax.faces.convert**.**DateTimeConverter**
The message identifier of the **FacesMessage** to be created if the conversion to **Date** fails.

**DateTerm** - Class in **javax.mail.search**
This class implements comparisons for Dates

**DateTerm(int, Date)** - Constructor for class **javax.mail.search**.**DateTerm**
Constructor.

**DATETIME_ID** - Static variable in class **javax.faces.convert**.**DateTimeConverter**
The message identifier of the **FacesMessage** to be created if the conversion to **DateTime** fails.

**DateTimeConverter** - Class in **javax.faces.convert**
Converter implementation for **java.util.Date** values.

**DateTimeConverter()** - Constructor for class **javax.faces.convert**.**DateTimeConverter**

**DConfigBean** - Interface in **javax.enterprise.deploy.spi**
The DConfigBean is a deployment configuration bean (DConfigBean) that is associated with one or more deployment descriptor beans, (DDBean).

**DConfigBeanRoot** - Interface in **javax.enterprise.deploy.spi**
A J2EE component module consists of one or more deployment descriptor files and zero or more non-deployment descriptor XML instance documents.

**DConfigBeanVersionType** - Class in **javax.enterprise.deploy.shared**
Class DConfigBeanVersionTypes defines enumeration values for the J2EE Platform version number.

**DConfigBeanVersionType(int)** - Constructor for class javax.enterprise.deploy.shared.DConfigBeanVersionType

Construct a new enumeration value with the given integer value.

**DConfigBeanVersionUnsupportedException** - Exception in javax.enterprise.deploy.spi.exceptions

This exception is to report that there is no support for the DConfigBean version requested.

**DConfigBeanVersionUnsupportedException(String)** - Constructor for exception javax.enterprise.deploy.spi.exceptions.DConfigBeanVersionUnsupportedException

Creates an new DConfigBeanVersionUnsupportedException object.

**DDBean** - Interface in javax.enterprise.deploy.model

An interface for beans that represent a fragment of a standard deployment descriptor.

**DDBeanCreateException** - Exception in javax.enterprise.deploy.model.exceptions

This exception reports errors in generating a DDBean.

**DDBeanCreateException()** - Constructor for exception javax.enterprise.deploy.model.exceptions.DDBeanCreateException

Creates new DDBeanCreateException without detail message.

**DDBeanCreateException(String)** - Constructor for exception javax.enterprise.deploy.model.exceptions.DDBeanCreateException

Constructs an DDBeanCreateException with the specified detail message.

**DDBeanRoot** - Interface in javax.enterprise.deploy.model

An interface that represents the root of a standard deployment descriptor.

**debug** - Variable in class javax.mail.Service

Debug flag for this service.

**DECIMAL_ID** - Static variable in class javax.faces.convert.BigDecimalConverter

The message identifier of the FacesMessage to be created if the conversion to BigDecimal fails.

**DeclarativeQueryManager** - Interface in javax.xml.registry

This interface provides the ability to execute declarative queries (e.g. declaredType)

**declaredType** - Variable in class javax.xml.bind.JAXBElement

Java datatype binding for xml element declaration's type.
DeclareRoles - Annotation Type in javax.annotation.security
Used by application to declare roles.
declare(FacesContext) - Method in class
javax.faces.component.UIComponent
Decode any new state of this UIComponent from the request contained in the specified FacesContext, and store this state as needed.
declare(FacesContext) - Method in class
javax.faces.component.UIComponentBase
declare(FacesContext) - Method in class
javax.faces.component.UIInput
decode(FacesContext, UIComponent) - Method in class
javax.faces.render.Renderer
Decode any new state of the specified UIComponent from the request contained in the specified FacesContext, and store that state on the UIComponent.
declare(InputStream, String) - Static method in class
javax.mail.internet.MimeUtility
Decode the given input stream.
declareText(String) - Static method in class
javax.mail.internet.MimeUtility
Decode "unstructured" headers, that is, headers that are defined as "*text" as per RFC 822.
declareWord(String) - Static method in class
javax.mail.internet.MimeUtility
The string is parsed using the rules in RFC 2047 for parsing an "encoded-word".
DEFAULT BUFFER - Static variable in class javax.servlet.jsp.JspWriter
Constant indicating that the Writer is buffered and is using the implementation default buffer size.
DEFAULT_CHARSET_NAME - Static variable in interface
javax.xml.registry.infomodelLocalizedString
The default name returned by getCharsetName if no other name has explicitly been set.
DEFAULT_DELIVERY_MODE - Static variable in interface
javax.jms.Message
The message producer's default delivery mode is PERSISTENT.
**DEFAULT_LIFECYCLE** - Static variable in class `javax.faces.lifecycle.LifecycleFactory`
   The lifecycle identifier for the default `Lifecycle` instance for this JavaServer Faces implementation.

**DEFAULT_PRIORITY** - Static variable in interface `javax.jms.Message`
   The message producer's default priority is 4.

**DEFAULT_SOAP_PROTOCOL** - Static variable in interface `javax.xml.soap.SOAPConstants`
   The default protocol: SOAP 1.1 for backwards compatibility.

**DEFAULT_SUFFIX** - Static variable in class `javax.faces.application.ViewHandler`
   The value to use for the default extension if the webapp is using url extension mapping.

**DEFAULT_SUFFIX_PARAM_NAME** - Static variable in class `javax.faces.application.ViewHandler`
   Allow the web application to define an alternate suffix for pages containing JSF content.

**DEFAULT_TIME_TO_LIVE** - Static variable in interface `javax.jms.Message`
   The message producer's default time to live is unlimited; the message never expires.

**DefaultValidationEventHandler** - Class in `javax.xml.bind.helpers`
   JAXB 1.0 only default validation event handler.

**DefaultValidationEventHandler()** - Constructor for class `javax.xml.bind.helpers.DefaultValidationEventHandler`

**delete()** - Method in interface `javax.jms.TemporaryQueue`
   Deletes this temporary queue.

**delete()** - Method in interface `javax.jms.TemporaryTopic`
   Deletes this temporary topic.

**delete(boolean)** - Method in class `javax.mail.Folder`
   Delete this Folder.

**delete()** - Method in interface `javax.security.jacc.PolicyConfiguration`
   Causes all policy statements to be deleted from this PolicyConfiguration and sets its internal state such that calling any method, other than delete, getContextID, or inService on the PolicyConfiguration will be rejected and cause an UnsupportedOperationException to be thrown.

**deleteAssociations(Collection)** - Method in interface
javax.xml.registry.**BusinessLifeCycleManager**

Deletes the Associations corresponding to the specified Keys.

**deleteClassificationSchemes(Collection)** - Method in interface javax.xml.registry.**BusinessLifeCycleManager**

Deletes the ClassificationSchemes corresponding to the specified Keys.

**deleteConcepts(Collection)** - Method in interface javax.xml.registry.**BusinessLifeCycleManager**

Deletes the Concepts corresponding to the specified Keys.

**DELETED** - Static variable in class javax.mail.event.**FolderEvent**

The folder was deleted.

**DELETED** - Static variable in class javax.mail.**Flags.Flag**

This message is marked deleted.

**DeleteException** - Exception in javax.xml.registry

A RegistryException that occurs during a delete action.

**DeleteException()** - Constructor for exception javax.xml.registry.**DeleteException**

Constructs a JAXRException object with no reason or embedded Throwable.

**DeleteException(String)** - Constructor for exception javax.xml.registry.**DeleteException**

Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**DeleteException(String, Throwable)** - Constructor for exception javax.xml.registry.**DeleteException**

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

**DeleteException(Throwable)** - Constructor for exception javax.xml.registry.**DeleteException**

Constructs a JAXRException object initialized with the given Throwable object.

**deleteObjects(Collection)** - Method in interface javax.xml.registry.**LifeCycleManager**

Deletes one or more previously submitted objects from the registry using the object keys.

**deleteObjects(Collection, String)** - Method in interface javax.xml.registry.**LifeCycleManager**

Deletes one or more previously submitted objects from the registry using the object keys.
using the object keys and a specified objectType attribute.

**deleteOrganizations(Collection)** - Method in interface
javax.xml.registry.BusinessLifeCycleManager
Deletes the organizations corresponding to the specified Keys.

**deletesAreDetected(int)** - Method in interface
javax.resource.cci.ResultSetInfo

**deleteServiceBindings(Collection)** - Method in interface
javax.xml.registry.BusinessLifeCycleManager
Deletes the ServiceBindings corresponding to the specified Keys.

**deleteServices(Collection)** - Method in interface
javax.xml.registry.BusinessLifeCycleManager
Deletes the services corresponding to the specified Keys.

**delistResource(XAResource, int)** - Method in interface
javax.transaction.Transaction
Disassociate the resource specified from the transaction associated with the target Transaction object.

**DeliveryMode** - Interface in javax.jms
The delivery modes supported by the JMS API are PERSISTENT and NON_PERSISTENT.

**DenyAll** - Annotation Type in javax.annotation.security
Specifies that no security roles are allowed to invoke the specified method(s) - i.e that the methods are to be excluded from execution in the J2EE container.

**DeployableObject** - Interface in javax.enterprise.deploy.model
The DeployableObject interface is an abstract representation of a J2EE deployable module (JAR, WAR, RAR, EAR).

**DeploymentConfiguration** - Interface in javax.enterprise.deploy.spi
An interface that defines a container for all the server-specific configuration information for a single top-level J2EE module.

**DeploymentFactory** - Interface in javax.enterprise.deploy.spi.factories
The DeploymentFactory interface is a deployment driver for a J2EE plaform product.

**DeploymentFactoryManager** - Class in javax.enterprise.deploy.shared.factories
The DeploymentFactoryManager class is a central registry for J2EE DeploymentFactory objects.

**DeploymentManager** - Interface in javax.enterprise.deploy.spi
The DeploymentManager object provides the core set of functions a
J2EE platform must provide for J2EE application deployment.

**DeploymentManagerCreationException** - Exception in `javax.enterprise.deploy.spi.exceptions`
This exception is to report problems in returning a DeploymentManager object cause by such things as server down, unable to authenticate and the like.

**DeploymentManagerCreationException(String)** - Constructor for exception
`javax.enterprise.deploy.spi.exceptionsDeploymentManagerCreationException`
Creates an new DeploymentManagerCreationException object.

**DeploymentStatus** - Interface in `javax.enterprise.deploy.spi.status`
The DeploymentStatus interface provides information about the progress status of a deployment action.

**deprecateObjects(Collection)** - Method in interface `javax.xml.registry.LifeCycleManager`
Deprecates one or more previously submitted objects.

**DeserializationContext** - Interface in `javax.xml.rpc.encoding`
The javax.xml.rpc.encoding.DeserializationContext interface is implemented by the JAX-RPC runtime system in an XML processing mechanism specific manner.

**Deserializer** - Interface in `javax.xml.rpc.encoding`
The javax.xml.rpc.encoding.Deserializer interface defines a base interface for deserializers.

**DeserializerFactory** - Interface in `javax.xml.rpc.encoding`
The javax.xml.rpc.encoding.DeserializerFactory is a factory of deserializers.

**Destination** - Interface in `javax.jms`
A Destination object encapsulates a provider-specific address.

**destroy()** - Method in class `javax.faces.webapp.FacesServlet`
Release all resources acquired at startup time.

**destroy()** - Method in interface `javax.resource.spi.ManagedConnection`
Destroys the physical connection to the underlying resource manager.

**destroy()** - Method in interface `javax.servlet.Filter`
Called by the web container to indicate to a filter that it is being taken out of service.

**destroy()** - Method in class `javax.servlet.GenericServlet`
Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.
**destroy()** - Method in interface `javax.servlet.Servlet`
   Called by the servlet container to indicate to a servlet that the servlet is being taken out of service.

**destroy()** - Method in class `javax.xml.rpc.handler.GenericHandler`
   The `destroy` method indicates the end of lifecycle for a Handler instance.

**destroy()** - Method in interface `javax.xml.rpc.handler.Handler`
   The `destroy` method indicates the end of lifecycle for a Handler instance.

**destroy()** - Method in interface `javax.xml.rpc.handler.HandlerChain`
   Indicates the end of lifecycle for a HandlerChain.

**destroy()** - Method in interface `javax.xml.rpc.server.ServiceLifecycle`
   JAX-RPC runtime system ends the lifecycle of a service endpoint instance by invoking the destroy method.

**detachNode()** - Method in interface `javax.xml.soap.Node`
   Removes this `Node` object from the tree.

**Detail** - Interface in `javax.xml.soap`
   A container for `DetailEntry` objects.

**DetailEntry** - Interface in `javax.xml.soap`
   The content for a `Detail` object, giving details for a `SOAPFault` object.

**dh** - Variable in class `javax.mail.internet.MimeBodyPart`
   The DataHandler object representing this Part's content.

**dh** - Variable in class `javax.mail.internet.MimeMessage`
   The DataHandler object representing this Message's content.

**DIGEST_AUTH** - Static variable in class `javax.faces.context.ExternalContext`
   String identifier for DIGEST authentication.

**DIGEST_AUTH** - Static variable in interface `javax.servlet.http.HttpServletRequest`
   String identifier for Digest authentication.

**disconnected(ConnectionEvent)** - Method in class `javax.mail.event.ConnectionAdapter`

**DISCONNECTED** - Static variable in class `javax.mail.event.ConnectionEvent`
   A connection was disconnected (not currently used).

**disconnected(ConnectionEvent)** - Method in interface `javax.mail.event.ConnectionListener`
   Invoked when a Store is disconnected.
**DiscriminatorColumn** - Annotation Type in `javax.persistence`
Is used to define the discriminator column for the SINGLE_TABLE and JOINED inheritance mapping strategies.

**DiscriminatorType** - Enum in `javax.persistence`
Defines supported types of the discriminator column.

**DiscriminatorValue** - Annotation Type in `javax.persistence`
Is used to specify the value of the discriminator column for entities of the given type.

**dispatch(String)** - Method in class `javax.faces.context.ExternalContext`
Dispatch a request to the specified resource to create output for this response.

**dispatch(Object)** - Method in class `javax.mail.event.ConnectionEvent`
Invokes the appropriate ConnectionListener method.

**dispatch(Object)** - Method in class `javax.mail.event.FolderEvent`
Invokes the appropriate FolderListener method.

**dispatch(Object)** - Method in class `javax.mail.event.MailEvent`
This method invokes the appropriate method on a listener for this event.

**dispatch(Object)** - Method in class `javax.mail.event.MessageChangedEvent`
Invokes the appropriate MessageChangedListener method.

**dispatch(Object)** - Method in class `javax.mail.event.MessageCountEvent`
Invokes the appropriate MessageCountListener method.

**dispatch(Object)** - Method in class `javax.mail.event.StoreEvent`
Invokes the appropriate StoreListener method.

**dispatch(Object)** - Method in class `javax.mail.event.TransportEvent`
Invokes the appropriate TransportListener method.

**Dispatch**<T> - Interface in `javax.xml.ws`
The `dispatch` interface provides support for the dynamic invocation of a service endpoint operations.

**DissociatableManagedConnection** - Interface in `javax.resource.spi`
This is a mix-in interface that may be optionally implemented by a ManagedConnection implementation.

**dissociateConnections()** - Method in interface `javax.resource.spi.DissociatableManagedConnection`
This method is called by an application server (that is capable of lazy connection association optimization) in order to dissociate a ManagedConnection instance from all of its connection handles.
**DISTRIBUTE** - Static variable in class
javax.enterprise.deploy.shared.**CommandType**
The DeploymentManager action operation being processed is distribute.

**distribute(Target[], File, File)** - Method in interface
javax.enterprise.deploy.spi.**DeploymentManager**
The distribute method performs three tasks; it validates the deployment configuration data, generates all container specific classes and interfaces, and moves the fully baked archive to the designated deployment targets.

**distribute(Target[], InputStream, InputStream)** - Method in interface
javax.enterprise.deploy.spi.**DeploymentManager**
Deprecated. as of Java EE 5, replaced with
**DeploymentManager.distribute(Target[], ModuleType, InputStream, InputStream)**

**distribute(Target[], ModuleType, InputStream, InputStream)** - Method in interface
javax.enterprise.deploy.spi.**DeploymentManager**
The distribute method performs three tasks; it validates the deployment configuration data, generates all container specific classes and interfaces, and moves the fully baked archive to the designated deployment targets.

**doAfterBody()** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**
Perform any processing necessary to handle the content implications of CASE 4 in the class description.

**doAfterBody()** - Method in class
javax.servlet.jsp.tagext.**BodyTagSupport**
After the body evaluation: do not reevaluate and continue with the page.

**doAfterBody()** - Method in interface javax.servlet.jsp.tagext.**IterationTag**
Process body (re)evaluation.

**doAfterBody()** - Method in class javax.servlet.jsp.tagext.**TagSupport**
Default processing for a body.

**doCatch(Throwable)** - Method in interface javax.servlet.jsp.tagext.**TryCatchFinally**
Invoked if a Throwable occurs while evaluating the BODY inside a tag or in any of the following methods: Tag.doStartTag(), Tag.doEndTag(), IterationTag.doAfterBody() and BodyTag.doInitBody.
**doDelete(HttpServletResponse)** - Method in class `javax.servlet.http.HttpServlet`
   Called by the server (via the `service` method) to allow a servlet to handle a DELETE request.

**doEndTag()** - Method in class `javax.faces.webapp.AttributeTag`
   Deprecated.

**doEndTag()** - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
   Perform any processing necessary to handle the content implications of CASE 3 in the class description.

**doEndTag()** - Method in class `javax.servlet.jsp.tagext.BodyTagSupport`
   Default processing of the end tag returning EVAL_PAGE.

**doEndTag()** - Method in interface `javax.servlet.jsp.tagext.Tag`
   Process the end tag for this instance.

**doEndTag()** - Method in class `javax.servlet.jsp.tagext.TagAdapter`
   Must not be called.

**doEndTag()** - Method in class `javax.servlet.jsp.tagext.TagSupport`
   Default processing of the end tag returning EVAL_PAGE.

**doFilter(ServletRequest, ServletResponse, FilterChain)** - Method in interface `javax.servlet.Filter`
   The `doFilter` method of the Filter is called by the container each time a request/response pair is passed through the chain due to a client request for a resource at the end of the chain.

**doFilter(ServletRequest, ServletResponse)** - Method in interface `javax.servlet.FilterChain`
   Causes the next filter in the chain to be invoked, or if the calling filter is the last filter in the chain, causes the resource at the end of the chain to be invoked.

**doFinally()** - Method in interface `javax.servlet.jsp.tagext.TryCatchFinally`
   Invoked in all cases after `doEndTag()` for any class implementing Tag, IterationTag or BodyTag.

   Called by the server (via the `service` method) to allow a servlet to handle a GET request.

**doHead(HttpServletRequest, HttpServletResponse)** - Method in class `javax.servlet.http.HttpServlet`
   Receives an HTTP HEAD request from the protected `service` method and handles the request.
doInitBody() - Method in class javax.faces.webapp.UICOMPONENTCLASSICTAGBASE
Prepare for evaluation of the body.
doInitBody() - Method in interface javax.servlet.jsp.tagext.BODYTAG
Prepare for evaluation of the body.
doInitBody() - Method in class javax.servlet.jsp.tagext.BODYTAGSUPPORT
Prepare for evaluation of the body just before the first body evaluation: no action.

DomHandler<ELEMENTT,RESULTT> extends Result> - Interface in javax.xml.bind.annotation
Converts an element (and its descendants) from/to DOM (or similar) representation.

doOptions(HttpServletRequest, HttpServletResponse) - Method in class javax.servlet.http.HttpServletRequest
Called by the server (via the service method) to allow a servlet to handle a OPTIONS request.
doPost(HttpServletRequest, HttpServletResponse) - Method in class javax.servlet.http.HttpServletRequest
Called by the server (via the service method) to allow a servlet to handle a POST request.
doPut(HttpServletRequest, HttpServletResponse) - Method in class javax.servlet.http.HttpServletRequest
Called by the server (via the service method) to allow a servlet to handle a PUT request.
doStartTag() - Method in class javax.faces.webapp.AttributeTag
Deprecated. Register the specified attribute name and value with the UICOMPONENT instance associated with our most immediately surrounding UICOMPONENT instance, if this UICOMPONENT does not already have a value for the specified attribute name.
doStartTag() - Method in class javax.faces.webapp.ConverterELTag
Create a new instance of the specified Converter class, and register it with the UICOMPONENT instance associated with our most immediately surrounding UICOMPONENTCLASSICTAGBASE instance, if the UICOMPONENT instance was created by this execution of the containing JSP page.
doStartTag() - Method in class javax.faces.webapp.ConverterTag
Deprecated. Create a new instance of the specified Converter class, and register it with the UICOMPONENT instance associated with our most immediately surrounding UICOMPONENT instance, if the
**UIComponent** instance was created by this execution of the containing JSP page.

**doStartTag()** - Method in class `javax.faces.webapp.FacetTag`
Return `EVAL_BODY_INCLUDE` to cause nested body content to be evaluated.

**doStartTag()** - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
Perform any processing necessary to find (or create) the **UIComponent** instance in the view corresponding to this tag instance in the page and, if and only if a component was created, insert it into the tree at the proper location as expected by the page author.

**doStartTag()** - Method in class `javax.faces.webapp.ValidatorELTag`
Create a new instance of the specified **Validator** class, and register it with the **UIComponent** instance associated with our most immediately surrounding **UIComponentTagBase** instance, if the **UIComponent** instance was created by this execution of the containing JSP page.

**doStartTag()** - Method in class `javax.faces.webapp.ValidatorTag`
Deprecated. Create a new instance of the specified **Validator** class, and register it with the **UIComponent** instance associated with our most immediately surrounding **UIComponentTag** instance, if the **UIComponent** instance was created by this execution of the containing JSP page.

**doStartTag()** - Method in class `javax.servlet.jsp.tagext.BodyTagSupport`
Default processing of the start tag returning `EVAL_BODY_BUFFERED`.

**doStartTag()** - Method in class `javax.servlet.jsp.tagext.Tag`
Process the start tag for this instance.

**doStartTag()** - Method in class `javax.servlet.jsp.tagext.TagAdapter`
Must not be called.

**doStartTag()** - Method in class `javax.servlet.jsp.tagext.TagSupport`
Default processing of the start tag, returning `SKIP_BODY`.

**doTag()** - Method in interface `javax.servlet.jsp.tagext.SimpleTag`
Called by the container to invoke this tag.

**doTag()** - Method in class `javax.servlet.jsp.tagext.SimpleTagSupport`
Default processing of the tag does nothing.

**doTrace(HttpServletRequest, HttpServletResponse)** - Method in class `javax.servlet.http.HttpServlet`
Called by the server (via the **service** method) to allow a servlet to
handle a TRACE request.

**DOUBLE_ID** - Static variable in class javax.faces.convert.DoubleConverter

The message identifier of the FacesMessage to be created if the conversion to Double fails.

**DoubleConverter** - Class in javax.faces.convert Converter implementation for java.lang.Double (and double primitive) values.

**DoubleConverter()** - Constructor for class javax.faces.convert.DoubleConverter

**DoubleHolder** - Class in javax.xml.rpc.holders

**DoubleHolder()** - Constructor for class javax.xml.rpc.holders.DoubleHolder

**DoubleHolder(double)** - Constructor for class javax.xml.rpc.holders.DoubleHolder

**DoubleRangeValidator** - Class in javax.faces.validator

DoubleRangeValidator is a Validator that checks the value of the corresponding component against specified minimum and maximum values.

**DoubleRangeValidator()** - Constructor for class javax.faces.validator.DoubleRangeValidator

Construct a Validator with no preconfigured limits.

**DoubleRangeValidator(double)** - Constructor for class javax.faces.validator.DoubleRangeValidator

Construct a Validator with the specified preconfigured limit.

**DoubleRangeValidator(double, double)** - Constructor for class javax.faces.validator.DoubleRangeValidator

Construct a Validator with the specified preconfigured limits.

**DoubleWrapperHolder** - Class in javax.xml.rpc.holders

**DoubleWrapperHolder()** - Constructor for class javax.xml.rpc.holders.DoubleWrapperHolder

**DoubleWrapperHolder(Double)** - Constructor for class javax.xml.rpc.holders.DoubleWrapperHolder
**doWork(Work)** - Method in interface javax.resource.spi.work.WorkManager
  Accepts a Work instance for processing.

**doWork(Work, long, ExecutionContext, WorkListener)** - Method in interface javax.resource.spi.work.WorkManager
  Accepts a Work instance for processing.

**DRAFT** - Static variable in class javax.mail.Flags.Flag
  This message is a draft.

**ds** - Variable in class javax.mail.internet.MimeMultipart
  The DataSource supplying our InputStream.

**DTD** - Interface in javax.xml.stream.events
  This is the top level interface for events dealing with DTDs

**DTD** - Static variable in interface javax.xml.stream.XMLStreamConstants
  Indicates an event is a DTD

**DuplicateKeyException** - Exception in javax.ejb
  The DuplicateKeyException exception is thrown if an entity EJB object cannot be created because an object with the same key already exists.

**DuplicateKeyException()** - Constructor for exception javax.ejb.DuplicateKeyException
  Constructs a DuplicateKeyException with no detail message.

**DuplicateKeyException(String)** - Constructor for exception javax.ejb.DuplicateKeyException
  Constructs a DuplicateKeyException with the specified detail message.

**DUPS_OK_ACKNOWLEDGE** - Static variable in interface javax.jms.Session
  This acknowledgment mode instructs the session to lazily acknowledge the delivery of messages.

**DYNAMIC_SOAP_PROTOCOL** - Static variable in interface javax.xml.soap.SOAPConstants
  Used to create MessageFactory instances that create SOAPMessages whose concrete type is based on the Content-Type MIME header passed to the createMessage method.

**DynamicAttributes** - Interface in javax.servlet.jsp.tagext
  For a tag to declare that it accepts dynamic attributes, it must implement this interface.
EAR - Static variable in class javax.enterprise.deploy.shared.ModuleType
The module is an EAR archive.

EditableValueHolder - Interface in javax.faces.component
EditableValueHolder is an extension of ValueHolder that describes additional features supported by editable components, including ValueChangeEvent and Validators.

EISSystemException - Exception in javax.resource.spi
An EISSystemException is used to indicate any EIS specific system-level error conditions.

EISSystemException() - Constructor for exception javax.resource.spi.EISSystemException
Constructs a new instance with null as its detail message.

EISSystemException(String) - Constructor for exception javax.resource.spi.EISSystemException
Constructs a new instance with the specified detail message.

EISSystemException(Throwables) - Constructor for exception javax.resource.spi.EISSystemException
Constructs a new throwable with the specified cause.

EISSystemException(String, Throwable) - Constructor for exception javax.resource.spi.EISSystemException
Constructs a new throwable with the specified detail message and cause.

EISSystemException(String, String) - Constructor for exception javax.resource.spi.EISSystemException
Constructs a new throwable with the specified detail message and an error code.

EJB - Annotation Type in javax.ejb
Indicates a dependency on the local or remote view of an Enterprise Java Bean.

EJB - Static variable in class javax.enterprise.deploy.shared.ModuleType
The module is an Enterprise Java Bean archive.

EJBAccessException - Exception in javax.ejb
This exception indicates that client access to a business method was denied.

EJBAccessException() - Constructor for exception
javax.ejb.**EJBAccessException**

Constructs an EJBAccessException with no detail message.

**EJBAccessException(String)** - Constructor for exception

javax.ejb.**EJBAccessException**

Constructs an EJBAccessException with the specified detailed message.

**ejbActivate()** - Method in interface javax.ejb.**EntityBean**

A container invokes this method when the instance is taken out of the pool of available instances to become associated with a specific EJB object.

**ejbActivate()** - Method in interface javax.ejb.**SessionBean**

The activate method is called when the instance is activated from its "passive" state.

**EJBContext** - Interface in javax.ejb

The EJBContext interface provides an instance with access to the container-provided runtime context of an enterprise Bean instance.

**EJBException** - Exception in javax.ejb

The EJBException exception is thrown by an enterprise Bean instance to its container to report that the invoked business method or callback method could not be completed because of an unexpected error (e.g. the instance failed to open a database connection).

**EJBException()** - Constructor for exception javax.ejb.**EJBException**

Constructs an EJBException with no detail message.

**EJBException(String)** - Constructor for exception

javax.ejb.**EJBException**

Constructs an EJBException with the specified detailed message.

**EJBException(Substring)** - Constructor for exception

javax.ejb.**EJBException**

Constructs an EJBException that embeds the originally thrown exception.

**EJBException(String, Exception)** - Constructor for exception

javax.ejb.**EJBException**

Constructs an EJBException that embeds the originally thrown exception with the specified detail message.

**EJBHome** - Interface in javax.ejb

The EJBHome interface must be extended by all enterprise Beans' remote home interfaces.

**ejbLoad()** - Method in interface javax.ejb.**EntityBean**


A container invokes this method to instruct the instance to synchronize its state by loading it state from the underlying database.

**EJBLocalHome** - Interface in `javax.ejb`
The EJBLocalHome interface must be extended by all enterprise Beans' local home interfaces.

**EJBLocalObject** - Interface in `javax.ejb`
The EJBLocalObject interface must be extended by all enterprise Beans' local interfaces.

**EJBMetaData** - Interface in `javax.ejb`
The EJBMetaData interface allows a client to obtain the enterprise Bean's meta-data information.

**EJBMETHODPermission** - Class in `javax.security.jacc`
Class for EJB method permissions.

**EJBMETHODPermission(String, String)** - Constructor for class `javax.security.jacc.EJBMETHODPermission`
Creates a new EJBMETHODPermission with the specified name and actions.

**EJBMETHODPermission(String, String, String, String[])** - Constructor for class `javax.security.jacc.EJBMETHODPermission`
Creates a new EJBMETHODPermission with name corresponding to the EJBName and actions composed from methodName, methodInterface, and methodParams.

**EJBMETHODPermission(String, String, Method)** - Constructor for class `javax.security.jacc.EJBMETHODPermission`
Creates a new EJBMETHODPermission with name corresponding to the EJBName and actions composed from methodInterface, and the Method object.

**EJBObject** - Interface in `javax.ejb`
The EJBObject interface is extended by all enterprise Beans' remote interfaces.

**ejbPassivate()** - Method in interface `javax.ejb.EntityBean`
A container invokes this method on an instance before the instance becomes disassociated with a specific EJB object.

**ejbPassivate()** - Method in interface `javax.ejb.SessionBean`
The passivate method is called before the instance enters the "passive" state.

**ejbRemove()** - Method in interface `javax.ejb.EntityBean`
A container invokes this method before it removes the EJB object.
that is currently associated with the instance.

**ejbRemove()** - Method in interface javax.ejb.MessageDrivenBean
A container invokes this method before it ends the life of the message-driven object.

**ejbRemove()** - Method in interface javax.ejb.SessionBean
A container invokes this method before it ends the life of the session object.

**EJBRoleRefPermission** - Class in javax.security.jacc
Class for EJB isCallerInRole (String reference) permissions.

**EJBRoleRefPermission(String, String)** - Constructor for class javax.security.jacc.EJBRoleRefPermission
Creates a new EJBRoleRefPermission with the specified name and actions.

**EJBs** - Annotation Type in javax.ejb
Declares multiple TYPE-level @EJB annotations.

**EJBStats** - Interface in javax.management.j2ee.statistics
Specifies statistics provided by all EJB component types.

**ejbStore()** - Method in interface javax.ejb.EntityBean
A container invokes this method to instruct the instance to synchronize its state by storing it to the underlying database.

**ejbTimeout(Timer)** - Method in interface javax.ejb.TimedObject
Invoked by the EJB container upon timer expiration.

**EJBTransactionRequiredException** - Exception in javax.ejb
This exception indicates that a request carried a null transaction context, but the target object requires an active transaction.

**EJBTransactionRequiredException()** - Constructor for exception javax.ejb.EJBTransactionRequiredException
Constructs an EJBTransactionRequiredException with no detail message.

**EJBTransactionRequiredException(String)** - Constructor for exception javax.ejb.EJBTransactionRequiredException
Constructs an EJBTransactionRequiredException with the specified detailed message.

**EJBTransactionRolledbackException** - Exception in javax.ejb
This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back.

**EJBTransactionRolledbackException()** - Constructor for exception javax.ejb.EJBTransactionRolledbackException
Constructs an EJBTransactionRolledbackException with no detail message.

EJBTransactionRolledbackException(String) - Constructor for exception javax.ejb.EJBTransactionRolledbackException
   Constructs an EJBTransactionRolledbackException with the specified detailed message.

EJBTransactionRolledbackException(String, Exception) - Constructor for exception javax.ejb.EJBTransactionRolledbackException
   Constructs an EJBTransactionRolledbackException with the specified detail message and a nested exception.

ELContext - Class in javax.el
   Context information for expression evaluation.

ELContext() - Constructor for class javax.el.ELContext

ELContextEvent - Class in javax.el
   An event which indicates that an ELContext has been created.

ELContextEvent(ELContext) - Constructor for class javax.el.ELContextEvent
   Constructs an ELContextEvent object to indicate that an ELContext has been created.

ELContextListener - Interface in javax.el
   The listener interface for receiving notification when an ELContext is created.

Element - Interface in javax.xml.bind
   This is an element marker interface.

ELException - Exception in javax.el
   Represents any of the exception conditions that can arise during expression evaluation.

ELException() - Constructor for exception javax.el.ELException
   Creates an ELException with no detail message.

ELException(String) - Constructor for exception javax.el.ELException
   Creates an ELException with the provided detail message.

ELException(throwable) - Constructor for exception javax.el.ELException
   Creates an ELException with the given cause.

ELException(String, Throwable) - Constructor for exception javax.el.ELException
   Creates an ELException with the given detail message and root cause.
**ELException** - Exception in `javax.servlet.jsp.el`
   
   Deprecated. As of JSP 2.1, replaced by `ELException`

**ELException()** - Constructor for exception `javax.servlet.jsp.el.ELException`
   
   Deprecated. Creates an ELException with no detail message.

**ELException(String)** - Constructor for exception `javax.servlet.jsp.el.ELException`
   
   Deprecated. Creates an ELException with the provided detail message.

**ELException(Throwlable)** - Constructor for exception `javax.servlet.jsp.el.ELException`
   
   Deprecated. Creates an ELException with the given root cause.

**ELParseException()** - Constructor for exception `javax.servlet.jsp.el.ELParseException`
   
   Deprecated. Creates an ELParseException with no detail message.

**ELParseException(String)** - Constructor for exception `javax.servlet.jsp.el.ELParseException`
   
   Deprecated. Creates an ELParseException with the provided detail message.

**ELParseException(Throwable)** - Constructor for exception `javax.servlet.jsp.el.ELParseException`
   
   Deprecated. Creates an ELParseException with the given root cause.

**ELParseException(String, Throwable)** - Constructor for exception `javax.servlet.jsp.el.ELParseException`
   
   Deprecated. Creates an ELParseException with the given detail message and root cause.

**ELResolver** - Class in `javax.el`

   Enables customization of variable and property resolution behavior for EL expression evaluation.

**ELResolver()** - Constructor for class `javax.el.ELResolver`

**EMAIL_ADDRESS** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**EmailAddress** - Interface in `javax.xml.registry.infomodel`

   Represents an email address.

**Embeddable** - Annotation Type in `javax.persistence`

   Defines a class whose instances are stored as an intrinsic part of an owning entity and share the identity of the entity.

**Embedded** - Annotation Type in `javax.persistence`
Defines a persistent field or property of an entity whose value is an instance of an embeddable class.

**EmbeddedId** - Annotation Type in [javax.persistence](https://docs.oracle.com/javaee/8/api/javax/persistence/)

Is applied to a persistent field or property of an entity class or mapped superclass to denote a composite primary key that is an embeddable class.

**encode(OutputStream, String)** - Static method in class javax.mail.internet.MimeUtility

Wrap an encoder around the given output stream.

**encode(OutputStream, String, String)** - Static method in class javax.mail.internet.MimeUtility

Wrap an encoder around the given output stream.

**encodeActionURL(String)** - Method in class javax.faces.context.ExternalContext

Return the input URL, after performing any rewriting needed to ensure that it will correctly identify an addressable action in the current application.

**encodeAll(FacesContext)** - Method in class javax.faces.component.UIComponent

If this component returns true from UIComponent.isRendered(), render this component and all its children that return true from isRendered(), regardless of the value of the UIComponent.getRendersChildren() flag.

**encodeBegin(FacesContext)** - Method in class javax.faces.component.UIComponent

If our rendered property is true, render the beginning of the current state of this UIComponent to the response contained in the specified FacesContext.

**encodeBegin(FacesContext)** - Method in class javax.faces.component.UIComponentBase

**encodeBegin(FacesContext)** - Method in class javax.faces.component.UIData

In addition to the default behavior, ensure that any saved per-row state for our child input components is discarded unless it is needed to rerender the current page with errors.
UIComponentBase.encodeBegin(javax.faces.context.FacesContext) behavior.

encodeBegin(FacesContext, UIComponent) - Method in class javax.faces.render.Renderer
  Render the beginning specified UIComponent to the output stream or writer associated with the response we are creating.

encodeBegin() - Method in class javax.faces.webapp.UIComponentClassicTagBase
  Deprecated. No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.

encodeChildren(FacesContext) - Method in class javax.faces.component.UIComponent
  If our rendered property is true, render the child UIComponent's of this UIComponent.

encodeChildren(FacesContext) - Method in class javax.faces.component.UIComponentBase

encodeChildren(FacesContext, UIComponent) - Method in class javax.faces.render.Renderer
  Render the child components of this UIComponent, following the rules described for encodeBegin() to acquire the appropriate value to be rendered.

encodeChildren() - Method in class javax.faces.webapp.UIComponentClassicTagBase
  Deprecated. No encoding is done during JSP page execution. Encoding is deferred until the page has completed executing to allow the entire tree to be built before any encoding occurs.

encodedPersonal - Variable in class javax.mail.internet.InternetAddress
  The RFC 2047 encoded version of the personal name.

encodeEnd(FacesContext) - Method in class javax.faces.component.UIComponent
  If our rendered property is true, render the ending of the current state of this UIComponent.

encodeEnd(FacesContext) - Method in class javax.faces.component.UIComponentBase

encodeEnd(FacesContext) - Method in class javax.faces.component.UIViewRoot
Override the default
UIComponentBase.encodeEnd(javax.faces.context.FacesContext)
behavior.

**encodeEnd(FacesContext, UIComponent)** - Method in class
javax.faces.render.Renderer
Render the ending of the current state of the specified UIComponent,
following the rules described for encodeBegin() to acquire the
appropriate value to be rendered.

**encodeEnd()** - Method in class
javax.faces.webapp.UIComponentClassicTagBase

**Deprecated. No encoding is done during JSP page execution.**
Encoding is deferred until the page has completed executing to allow
the entire tree to be built before any encoding occurs.

**encodeNamespace(String)** - Method in class
javax.faces.context.ExternalContext
Return the specified name, after prefixing it with a namespace that
ensures that it will be unique within the context of a particular page.

**encodeRedirectURL(String)** - Method in interface
javax.servlet.http.HttpServletResponse
Encodes the specified URL for use in the sendRedirect method or, if
encoding is not needed, returns the URL unchanged.

**encodeRedirectUrl(String)** - Method in interface
javax.servlet.http.HttpServletResponse

**Deprecated. As of version 2.1, use encodeRedirectURL(String url) instead**

**encodeRedirectURL(String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to return
encodeRedirectURL(String url) on the wrapped response object.

**encodeRedirectUrl(String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to return
encodeRedirectUrl(String url) on the wrapped response object.

**encodeResourceURL(String)** - Method in class
javax.faces.context.ExternalContext
Return the input URL, after performing any rewriting needed to
ensure that it will correctly identify an addressable resource in the
current application.

**encodeText(String)** - Static method in class
javax.mail.internet.MimeUtility
   Encode a RFC 822 "text" token into mail-safe form as per RFC 2047.
   **encodeText(String, String, String)** - Static method in class
javax.mail.internet.MimeUtility
   Encode a RFC 822 "text" token into mail-safe form as per RFC 2047.
   **encodeURL(String)** - Method in interface
javax.servlet.http.HttpServletResponse
   Encodes the specified URL by including the session ID in it, or, if
   encoding is not needed, returns the URL unchanged.
   **encodeUrl(String)** - Method in interface
javax.servlet.http.HttpServletResponse
   Deprecated. As of version 2.1, use encodeURL(String url) instead
   **encodeURL(String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
   The default behavior of this method is to call encodeURL(String url)
   on the wrapped response object.
   **encodeUrl(String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
   The default behavior of this method is to call encodeUrl(String url) on
   the wrapped response object.
   **encodeWord(String)** - Static method in class
javax.mail.internet.MimeUtility
   Encode a RFC 822 "word" token into mail-safe form as per RFC
   2047.
   **encodeWord(String, String, String)** - Static method in class
javax.mail.internet.MimeUtility
   Encode a RFC 822 "word" token into mail-safe form as per RFC
   2047.
   **encodingSet()** - Method in interface
javax.xml.stream.events.StartDocument
   Returns true if CharacterEncodingScheme was set in the encoding
   declaration of the document
   **ENCODINGSTYLE_URI_PROPERTY** - Static variable in interface
javax.xml.rpc.Call
   Standard property for encoding Style: Encoding style specified as a
   namespace URI.
   **end(Xid, int)** - Method in interface javax.transaction.xa.XAResource
   Ends the work performed on behalf of a transaction branch.
   **END_DOCUMENT** - Static variable in interface
javax.xml.stream.XMLStreamConstants
Indicates an event is an end document
**END** **ELEMENT** - Static variable in interface
javax.xml.stream.XMLStreamConstants
Indicates an event is an end element
**endDocument()** - Method in class javax.faces.context.ResponseWriter
Write whatever text should end a response.
**endDocument()** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call
ResponseWriter.endDocument() on the wrapped ResponseWriter object.
**EndDocument** - Interface in javax.xml.stream.events
A marker interface for the end of the document
**endElement(String)** - Method in class javax.faces.context.ResponseWriter
Write the end of an element, after closing any open element created by a call to startElement().
**endElement(String)** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call
ResponseWriter.endElement(String) on the wrapped ResponseWriter object.
**EndElement** - Interface in javax.xml.stream.events
An interface for the end element event.
**Endpoint** - Class in javax.xml.ws
A Web service endpoint.
**Endpoint()** - Constructor for class javax.xml.ws.Endpoint
**ENDPOINT_ADDRESS_PROPERTY** - Static variable in interface
javax.xml.rpc.Stub
Standard property: Target service endpoint address.
**ENDPOINT_ADDRESS_PROPERTY** - Static variable in interface
javax.xml.ws.BindingProvider
Standard property: Target service endpoint address.
This is called during the activation of a message endpoint.
**endpointDeactivation(MessageEndpointFactory, ActivationSpec)** -
Method in interface javax.resource.spi.**ResourceAdapter**
   This is called when a message endpoint is deactivated.

**enlistResource(XAResource)** - Method in interface
java.transaction.**Transaction**
   Enlist the resource specified with the transaction associated with the
target Transaction object.

**EnterpriseBean** - Interface in **javax.ejb**
   The EnterpriseBean interface must be implemented by every
enterprise Bean class.

**Entity** - Annotation Type in **javax.persistence**
   Specifies that the class is an entity.

**ENTITY_DECLARATION** - Static variable in interface
javax.xml.stream.**XMLStreamConstants**
   Indicates a Entity Declaration

**ENTITY_REFERENCE** - Static variable in interface
javax.xml.stream.**XMLStreamConstants**
   Indicates an event is an entity reference

**EntityBean** - Interface in **javax.ejb**
   The EntityBean interface is implemented by every entity enterprise
Bean class.

**EntityBeanStats** - Interface in **javax.management.j2ee.statistics**
   Specifies statistics provided by entity beans.

**EntityContext** - Interface in **javax.ejb**
   The EntityContext interface provides an instance with access to the
container-provided runtime context of an entity enterprise Bean
instance.

**EntityDeclaration** - Interface in **javax.xml.stream.events**
   An interface for handling Entity Declarations This interface is used to
record and report unparsed entity declarations.

**EntityExistsException** - Exception in **javax.persistence**
   Thrown by the persistence provider when
   EntityManager.persist(Object) is called and the entity already
   exists.

**EntityExistsException()** - Constructor for exception
java.persistence.**EntityExistsException**
   Constructs a new EntityExistsException exception with null as its
detail message.

**EntityExistsException(String)** - Constructor for exception
java.persistence.**EntityExistsException**
Constructs a new `EntityExistsException` exception with the specified detail message.

**EntityExistsException(String, Throwable)** - Constructor for exception

Constructs a new `EntityExistsException` exception with the specified detail message and cause.

**EntityExistsException(Throwabla** - Constructor for exception

Constructs a new `EntityExistsException` exception with the specified cause.

**EntityListeners** - Annotation Type in `javax.persistence`

Specifies the callback listener classes to be used for an entity or mapped superclass.

**EntityManager** - Interface in `javax.persistence`

Interface used to interact with the persistence context.

**EntityManagerFactory** - Interface in `javax.persistence`

The `EntityManagerFactory` interface is used by the application to obtain an application-managed entity manager.

**EntityNotFoundException** - Exception in `javax.persistence`

Thrown by the persistence provider when an entity reference obtained by `EntityManager.getReference(Class, Object)` is accessed but the entity does not exist.

**EntityNotFoundException()** - Constructor for exception

Constructs a new `EntityNotFoundException` exception with null as its detail message.

**EntityNotFoundException(String)** - Constructor for exception

Constructs a new `EntityNotFoundException` exception with the specified detail message.

**EntityReference** - Interface in `javax.xml.stream.events`

An interface for handling Entity events.

**EntityResult** - Annotation Type in `javax.persistence`

References an entity in the SELECT clause of a SQL query.

**EntityTransaction** - Interface in `javax.persistence`

The `EntityTransaction` interface is used to control resource transactions on resource-local entity managers.

**entries()** - Method in interface

`javax.enterprise.deploy.model.DeployableObject`
Returns an enumeration of the module file entries.

**ENUM_ID** - Static variable in class `javax.faces.convert.EnumConverter`

The message identifier of the `FacesMessage` to be created if the conversion to `Enum` fails.

**ENUM_NO_CLASS_ID** - Static variable in class `javax.faces.convert.EnumConverter`

The message identifier of the `FacesMessage` to be created if the conversion to `Enum` fails and no target class has been provided.

**EnumConverter** - Class in `javax.faces.convert` implementation for `java.lang.Enum` (and enum primitive) values.

**EnumConverter()** - Constructor for class `javax.faces.convert.EnumConverter`

**EnumConverter(Class)** - Constructor for class `javax.faces.convert.EnumConverter`

**Enumerated** - Annotation Type in `javax.persistence`

Specifies that a persistent property or field should be persisted as a enumerated type.

**EnumType** - Enum in `javax.persistence`

Defines mapping for the enumerated types.

**ENVELOPE** - Static variable in class `javax.mail.FetchProfile.Item`

This is the Envelope item.

**ENVELOPE_CHANGED** - Static variable in class `javax.mail.event.MessageChangedEvent`

The message's envelope (headers, but not body) changed.

**EOF** - Static variable in class `javax.mail.internet.HeaderTokenizer.Token`

Token type indicating end of input.

**EQ** - Static variable in class `javax.mail.search.ComparisonTerm`

**equals(DataFlavor)** - Method in class `javax.activation.ActivationDataFlavor`

Compares the DataFlavor passed in with this DataFlavor; calls the `isMimeTypeEqual` method.

**equals(Object)** - Method in class `javax.el.Expression`

Determines whether the specified object is equal to this `Expression`.

**equals(Object)** - Method in class `javax.faces.validator.DoubleRangeValidator`
equals(Object) - Method in class javax.faces.validator.LengthValidator

equals(Object) - Method in class javax.faces.validator.LongRangeValidator

equals(Object) - Method in class javax.mail.Address
The equality operator.

equals(Object) - Method in class javax.mail.Flags
Check whether the two Flags objects are equal.

equals(Object) - Method in class javax.mail.internet.InternetAddress
The equality operator.

equals(Object) - Method in class javax.mail.internet.NewsAddress
The equality operator.

equals(Object) - Method in class javax.mail.search.AddressStringTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.AddressTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.AndTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.BodyTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.ComparisonTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.DateTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.FlagTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.FromStringTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.FromTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.HeaderTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.IntegerComparisonTerm
Equality comparison.

equals(Object) - Method in class javax.mail.search.MessageIDTerm
Equality comparison.
equals(Object) - Method in class javax.mail.search.MessageNumberTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.NotTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.OrTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.ReceivedDateTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.RecipientStringTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.RecipientTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.SentDateTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.SizeTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.StringTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.search.SubjectTerm
  Equality comparison.
equals(Object) - Method in class javax.mail.URLName
  Compares two URLNames.
equals(Object) - Method in class javax.resource.cci.Record
  Check if this instance is equal to another Record.
equals(Object) - Method in class javax.resource.spi.ConnectionRequestInfo
  Checks whether this instance is equal to another.
equals(Object) - Method in class javax.resource.spi.ManagedConnectionFactory
  Check if this ManagedConnectionFactory is equal to another ManagedConnectionFactory.
equals(Object) - Method in class javax.resource.spi.security.GenericCredential
  Deprecated. Tests if this GenericCredential instance refers to the same entity as the supplied object.
equals(Object) - Method in class javax.resource.spi.security.PasswordCredential
  Compares this PasswordCredential with the specified object for
equality.

**equals(Object)** - Method in class `javax.security.jacc.EJBMethodPermission`
Checks two EJBMethodPermission objects for equality.

**equals(Object)** - Method in class `javax.security.jacc.EJBRoleRefPermission`
Checks two EJBRoleRefPermission objects for equality.

**equals(Object)** - Method in class `javax.security.jacc.WebResourcePermission`
Checks two WebResourcePermission objects for equality.

**equals(Object)** - Method in class `javax.security.jacc.WebRoleRefPermission`
Checks two WebRoleRefPermission objects for equality.

**equals(Object)** - Method in class `javax.security.jacc.WebUserDataPermission`
Checks two WebUserDataPermission objects for equality.

**ERROR** - Static variable in interface `javax.xml.bind.ValidationEvent`
Conditions that correspond to the definition of "error" in section 1.2 of the W3C XML 1.0 Recommendation

**errorCode** - Variable in exception `javax.transaction.SystemException`
The error code with which to create the SystemException.

**errorCode** - Variable in exception `javax.transaction.xa.XAException`
The error code with which to create the SystemException.

**ErrorData** - Class in `javax.servlet.jsp`
Contains information about an error, for error pages.

**ErrorData(Throw, int, String, String)** - Constructor for class `javax.servlet.jsp.ErrorData`
Creates a new ErrorData object.

**EVAL_BODY_AGAIN** - Static variable in interface `javax.servlet.jsp.tagext.IterationTag`
Request the reevaluation of some body.

**EVAL_BODY_BUFFERED** - Static variable in interface `javax.servlet.jsp.tagext.BodyTag`
Request the creation of new buffer, a BodyContent on which to evaluate the body of this tag.

**EVAL_BODY_INCLUDE** - Static variable in interface `javax.servlet.jsp.tagext.Tag`
Evaluate body into existing out stream.

**EVAL_BODY_TAG** - Static variable in interface

javax.servlet.jsp.tagext.**BodyTag**

**Deprecated.** As of Java JSP API 1.2, use `BodyTag.EVAL_BODY_BUFFERED` or `IterationTag.EVAL_BODY_AGAIN`.

**EVAL_PAGE** - Static variable in interface `javax.servlet.jsp.tagext.Tag` Continue evaluating the page.

**evaluate(VariableResolver)** - Method in class `javax.servlet.jsp.el.Expression`

**Deprecated.** Evaluates an expression that was previously prepared.

**evaluate(String, Class, VariableResolver, FunctionMapper)** - Method in class `javax.servlet.jsp.el.ExpressionEvaluator`

**Deprecated.** Evaluates an expression.

**evaluateExpressionGet(FacesContext, String, Class)** - Method in class `javax.faces.application.Application`

Get a value by evaluating an expression.

**EvaluationException** - Exception in `javax.faces.el`

**Deprecated.** This has been replaced by `ELException`.

**EvaluationException()** - Constructor for exception `javax.faces.el.EvaluationException`

**Deprecated.** Construct a new exception with no detail message or root cause.

**EvaluationException(String)** - Constructor for exception `javax.faces.el.EvaluationException`

**Deprecated.** Construct a new exception with the specified detail message and no root cause.

**EvaluationException(Throwables)** - Constructor for exception `javax.faces.el.EvaluationException`

**Deprecated.** Construct a new exception with the specified root cause.

**EVENT_TYPE_CREATED** - Static variable in interface `javax.xml.registry.infomodel.AuditableEvent` An event where a RegistryObject is created.

**EVENT_TYPE_DELETED** - Static variable in interface `javax.xml.registry.infomodel.AuditableEvent` An event where a RegistryObject is deleted.
**EVENT_TYPE_DEPRECATED** - Static variable in interface javax.xml.registry.infomodel.AuditableEvent
   An event where a RegistryObject is deprecated.

**EVENT_TYPE_UNDEPRECATED** - Static variable in interface javax.xml.registry.infomodel.AuditableEvent
   An event where a RegistryObject isundeprecated.

**EVENT_TYPE_UPDATED** - Static variable in interface javax.xml.registry.infomodel.AuditableEvent
   An event where a RegistryObject is updated.

**EVENT_TYPE_VERSIONED** - Static variable in interface javax.xml.registry.infomodel.AuditableEvent
   An event where a RegistryObject is versioned.

**EventFilter** - Interface in javax.xml.stream
   This interface declares a simple filter interface that one can create to filter XMLEventReaders

**EventReaderDelegate** - Class in javax.xml.stream.util
   This is the base class for deriving an XMLEventReader filter.

**EventReaderDelegate()** - Constructor for class javax.xml.stream.util.EventReaderDelegate
   Construct an empty filter with no parent.

**EventReaderDelegate(XMLEventReader)** - Constructor for class javax.xml.stream.util.EventReaderDelegate
   Construct an filter with the specified parent.

**EXACT_NAME_MATCH** - Static variable in interface javax.xml.registry.FindQualifier

**examineAllHeaderElements()** - Method in interface javax.xml.soap.SOAPHeader
   Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object.

**examineHeaderElements(String)** - Method in interface javax.xml.soap.SOAPHeader
   Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified actor.

**examineMustUnderstandHeaderElements(String)** - Method in interface javax.xml.soap.SOAPHeader
   Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified actor and that have a MustUnderstand attribute whose value is equivalent to true.
**EXCEPTION** - Static variable in class `javax.servlet.jsp.PageContext`
Name used to store uncaught exception in ServletRequest attribute list and PageContext name table.

**ExceptionListener** - Interface in `javax.jms`
If a JMS provider detects a serious problem with a `Connection` object, it informs the `Connection` object's `ExceptionListener`, if one has been registered.

**ExcludeClassInterceptors** - Annotation Type in `javax.interceptor`
Used to exclude class-level interceptors for a business method.

**ExcludeDefaultInterceptors** - Annotation Type in `javax.interceptor`
Used to exclude default interceptors for a bean or a business method.

**ExcludeDefaultListeners** - Annotation Type in `javax.persistence`
Specifies that the invocation of default listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.

**ExcludeSuperclassListeners** - Annotation Type in `javax.persistence`
Specifies that the invocation of superclass listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.

**excludeUnlistedClasses()** - Method in interface `javax.persistence.spi.PersistenceUnitInfo`
Returns whether classes in the root of the persistence unit that have not been explicitly listed are to be included in the set of managed classes.

**EXECUTE** - Static variable in class `javax.enterprise.deploy.shared.ActionType`
The DeploymentManager action command is executing.

**execute()** - Method in interface `javax.enterprise.deploy.spi.status.ClientConfiguration`
This method performs an exec and starts the application client running in another process.

**execute(FacesContext)** - Method in class `javax.faces.lifecycle.Lifecycle`
Execute all of the phases of the request processing lifecycle, up to but not including the `Render Response` phase, as described in the JavaServer Faces Specification, in the specified order.

**execute(InteractionSpec, Record, Record)** - Method in interface `javax.resource.cci.Interaction`
Executes an interaction represented by the `InteractionSpec`.

**execute(InteractionSpec, Record)** - Method in interface `javax.resource.cci.Interaction`
Executes an interaction represented by the InteractionSpec.

**executeQuery(Query)** - Method in interface javax.xml.registry.DeclarativeQueryManager
Executes a query as specified by query parameter.

**executeUpdate()** - Method in interface javax.persistence.Query
Execute an update or delete statement.

**ExecutionContext** - Class in javax.resource.spi.work
This class models an execution context (transaction, security, etc) with which the Work instance must be executed.

**ExecutionContext()** - Constructor for class javax.resource.spi.work.ExecutionContext

**exists()** - Method in class javax.mail.Folder
Tests if this folder physically exists on the Store.

**Expression** - Class in javax.el
Base class for the expression subclasses ValueExpression and MethodExpression, implementing characteristics common to both.

**Expression()** - Constructor for class javax.el.Expression

**Expression** - Class in javax.servlet.jsp.el
Deprecated. As of JSP 2.1, replaced by ValueExpression

**Expression()** - Constructor for class javax.servlet.jsp.el.Expression
Deprecated.

**ExpressionEvaluator** - Class in javax.servlet.jsp.el
Deprecated. As of JSP 2.1, replaced by ExpressionFactory

**ExpressionEvaluator()** - Constructor for class javax.servlet.jsp.el.ExpressionEvaluator
Deprecated.

**ExpressionFactory** - Class in javax.el

**ExpressionFactory()** - Constructor for class javax.el.ExpressionFactory

**expunge()** - Method in class javax.mail.Folder
Expunge (permanently remove) messages marked DELETED.

**expunged** - Variable in class javax.mail.Message
True if this message has been expunged.

**ExtensibleObject** - Interface in javax.xml.registry.infomodel
An ExtensibleObject is one that allows itself to be extended by utilizing dynamically added Slots that add arbitrary attributes to the
object on a per instance basis.

**EXTERNAL_IDENTIFIER** - Static variable in interface
javax.xml.registry.LifeCycleManager

**EXTERNAL_LINK** - Static variable in interface
javax.xml.registry.LifeCycleManager

**ExternalContext** - Class in javax.faces.context
This class allows the Faces API to be unaware of the nature of its containing application environment.

**ExternalContext()** - Constructor for class
javax.faces.context.ExternalContext

**ExternalIdentifier** - Interface in javax.xml.registry.infomodel
ExternalIdentifier instances provide the additional identifier information to RegistryObjects such as DUNS number, Social Security Number, or an alias name of the organization.

**ExternalLink** - Interface in javax.xml.registry.infomodel
ExternalLink instances model a named URI to content that may reside outside the registry.

**extractAllHeaderElements()** - Method in interface
javax.xml.soap.SOAPHeader
Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object and detaches them from this SOAPHeader object.

**extractContentAsDocument()** - Method in interface
javax.xml.soap.SOAPBody
Creates a new DOM Document and sets the first child of this SOAPBody as its document element.

**extractHeaderElements(String)** - Method in interface
javax.xml.soap.SOAPHeader
Returns an Iterator over all the SOAPHeaderElement objects in this SOAPHeader object that have the specified actor and detaches them from this SOAPHeader object.

**EXTRINSIC_OBJECT** - Static variable in interface
javax.xml.registry.LifeCycleManager

**ExtrinsicObject** - Interface in javax.xml.registry.infomodel
ExtrinsicObjects provide metadata that describes submitted content whose type is not intrinsically known to the registry and therefore
must be described by means of additional attributes (e.g., mime type).
**FACES_CONTEXT_FACTORY** - Static variable in class `javax.faces.FactoryFinder`

The property name for the `FacesContextFactory` class name.

**FACES_MESSAGES** - Static variable in class `javax.faces.application.FacesMessage`

ResourceBundle identifier for messages whose message identifiers are defined in the JavaServer Faces specification.

**FacesContext** - Class in `javax.faces.context`

`FacesContext` contains all of the per-request state information related to the processing of a single JavaServer Faces request, and the rendering of the corresponding response.

**FacesContext()** - Constructor for class `javax.faces.context.FacesContext`

**FacesContextFactory** - Class in `javax.faces.context`

`FacesContextFactory` is a factory object that creates (if needed) and returns new `FacesContext` instances, initialized for the processing of the specified request and response objects.

**FacesContextFactory()** - Constructor for class `javax.faces.context.FacesContextFactory`

**FacesEvent** - Class in `javax.faces.event`

`FacesEvent` is the base class for user interface and application events that can be fired by `UIComponent`s.

**FacesEvent(UIComponent)** - Constructor for class `javax.faces.event.FacesEvent`

Construct a new event object from the specified source component.

**FacesException** - Exception in `javax.faces`

This class encapsulates general JavaServer Faces exceptions.

**FacesException()** - Constructor for exception `javax.faces.FacesException`

Construct a new exception with no detail message or root cause.

**FacesException(String)** - Constructor for exception `javax.faces.FacesException`

Construct a new exception with the specified detail message and no root cause.
**FacesException(Throwable)** - Constructor for exception
javax.faces.**FacesException**
   Construct a new exception with the specified root cause.

**FacesException(String, Throwable)** - Constructor for exception
javax.faces.**FacesException**
   Construct a new exception with the specified detail message and root cause.

**FacesListener** - Interface in **javax.faces.event**
   A generic base interface for event listeners for various types of FacesEventS.

**FacesMessage** - Class in **javax.faces.application**
   **FacesMessage** represents a single validation (or other) message, which is typically associated with a particular component in the view.

**FacesMessage()** - Constructor for class
class javax.faces.application.**FacesMessage**
   Construct a new **FacesMessage** with no initial values.

**FacesMessage(String)** - Constructor for class
class javax.faces.application.**FacesMessage**
   Construct a new **FacesMessage** with just a summary.

**FacesMessage(String, String)** - Constructor for class
class javax.faces.application.**FacesMessage**
   Construct a new **FacesMessage** with the specified initial values.

**FacesMessage(FacesMessage.Severity, String, String)** - Constructor for class
class javax.faces.application.**FacesMessage**
   Construct a new **FacesMessage** with the specified initial values.

**FacesMessage.Severity** - Class in **javax.faces.application**
   Class used to represent message severity levels in a typesafe enumeration.

**FacesServlet** - Class in **javax.faces.webapp**
   **FacesServlet** is a servlet that manages the request processing lifecycle for web applications that are utilizing JavaServer Faces to construct the user interface.

**FacesServlet()** - Constructor for class
class javax.faces.webapp.**FacesServlet**

**FacetTag** - Class in **javax.faces.webapp**
   **FacetTag** is the JSP mechanism for denoting a **UIComponent** is to be added as a **facet** to the component associated with its parent.

**FacetTag()** - Constructor for class
class javax.faces.webapp.**FacetTag**
**FactoryConfigurationError** - Error in `javax.xml.stream`
   An error class for reporting factory configuration errors.

**FactoryConfigurationError()** - Constructor for error
   `javax.xml.stream.FactoryConfigurationError`
   Default constructor

**FactoryConfigurationError( Exception )** - Constructor for error
   `javax.xml.stream.FactoryConfigurationError`
   Construct an exception with a nested inner exception

**FactoryConfigurationError( Exception, String )** - Constructor for error
   `javax.xml.stream.FactoryConfigurationError`
   Construct an exception with a nested inner exception and a message

**FactoryConfigurationError( String, Exception )** - Constructor for error
   `javax.xml.stream.FactoryConfigurationError`
   Construct an exception with a nested inner exception and a message

**FactoryConfigurationError( String )** - Constructor for error
   `javax.xml.stream.FactoryConfigurationError`
   Construct an exception with associated message

**FactoryFinder** - Class in `javax.faces`
   **FactoryFinder** implements the standard discovery algorithm for all factory objects specified in the JavaServer Faces APIs.

**FAILED** - Static variable in class
   `javax.enterprise.deploy.shared.StateType`
   The action operation has failed.

**FATAL_ERROR** - Static variable in interface
   `javax.xml.bind.ValidationEvent`
   Conditions that correspond to the definition of "fatal error" in section 1.2 of the W3C XML 1.0 Recommendation

**FederatedConnection** - Interface in `javax.xml.registry`
   Represents a single logical connection to a federation or group of registry providers.

**fetch( Message[], FetchProfile )** - Method in class `javax.mail.Folder`
   Prefetch the items specified in the FetchProfile for the given Messages.

**FetchProfile** - Class in `javax.mail`
   Clients use a FetchProfile to list the Message attributes that it wishes to prefetch from the server for a range of messages.

**FetchProfile()** - Constructor for class `javax.mail.FetchProfile`
Create an empty FetchProfile.

**FetchProfile.Item** - Class in [javax.mail](https://docs.oracle.com/javae/8/javamail/api/

This inner class is the base class of all items that can be requested in a FetchProfile.

**FetchProfile.Item(String)** - Constructor for class

This is a constructor for an item.

**FetchType** - Enum in [javax.persistence](https://docs.oracle.com/javae/8/persistence/

Defines strategies for fetching data from the database.

**FieldResult** - Annotation Type in [javax.persistence](https://docs.oracle.com/javae/8/persistence/

This is used to map the columns specified in the SELECT list of the query to the properties or fields of the entity class.

**FileDataSource** - Class in [javax.activation](https://docs.oracle.com/javae/8/activation/

This class implements a simple DataSource object that encapsulates a file.

**FileDataSource(File)** - Constructor for class

This is for creating a FileDataSource from a File object.

**FileDataSource(String)** - Constructor for class

This is for creating a FileDataSource from the specified path name.

**FileTypeMap** - Class in [javax.activation](https://docs.oracle.com/javae/8/activation/

This is an abstract class that provides a data typing interface for files.

**FileTypeMap()** - Constructor for class [javax.activation.FileTypeMap](https://docs.oracle.com/javae/8/activation/

This is the default constructor.

**Filter** - Interface in [javax.servlet](https://docs.oracle.com/javae/8/servlet/

This is an object that performs filtering tasks on either the request to a resource (a servlet or static content), or on the response from a resource, or both.

**FilterChain** - Interface in [javax.servlet](https://docs.oracle.com/javae/8/servlet/

This is an object provided by the servlet container to the developer giving a view into the invocation chain of a filtered request for a resource.

**FilterConfig** - Interface in [javax.servlet](https://docs.oracle.com/javae/8/servlet/

This is a filter configuration object used by a servlet container to pass information to a filter during initialization.

**finalize()** - Method in class [javax.mail.Folder](https://docs.oracle.com/javae/8/javamail/

**finalize()** - Method in class javax.mail.Service
Stop the event dispatcher thread so the queue can be garbage collected.

**finalize()** - Method in class javax.mail.util.SharedFileInputStream
Force this stream to close.

**find(Class<T>, Object)** - Method in interface javax.persistence.EntityManager
Find by primary key.

**findAncestorWithClass(JspTag, Class<?>)** - Static method in class javax.servlet.jsp.tagext.SimpleTagSupport
Find the instance of a given class type that is closest to a given instance.

**findAncestorWithClass(Tag, Class)** - Static method in class javax.servlet.jsp.tagext.TagSupport
Find the instance of a given class type that is closest to a given instance.

**findAssociations(Collection, String, String, Collection)** - Method in interface javax.xml.registry.BusinessQueryManager
Finds all Association objects that match all of the criteria specified by the parameters of this call.

**findAttribute(String)** - Method in class javax.servlet.jsp.JspContext
Searches for the named attribute in page, request, session (if valid), and application scope(s) in order and returns the value associated or null.

**findCallerAssociations(Collection, Boolean, Boolean, Collection)** - Method in interface javax.xml.registry.BusinessQueryManager
Finds all Association objects owned by the caller that match all of the criteria specified by the parameters of this call.

**findClassificationSchemeByName(Collection, String)** - Method in interface javax.xml.registry.BusinessQueryManager
Finds a ClassificationScheme by name based on the specified find qualifiers and name pattern.

**findClassificationSchemes(Collection, Collection, Collection, Collection)** - Method in interface javax.xml.registry.BusinessQueryManager
Finds all ClassificationScheme objects that match all of the criteria specified by the parameters of this call.

**findComponent(String)** - Method in class javax.faces.component.UIComponent
Search for and return the `UIComponent` with an id that matches the specified search expression (if any), according to the algorithm described below.

**findComponent(String)** - Method in class `javax.faces.component.UIComponentBase`

According to the algorithm described below.

**findComponent(FacesContext)** - Method in class `javax.faces.webapp.UIComponentClassicTagBase`

Find and return the `UIComponent`, from the component tree, that corresponds to this tag handler instance.

**findConceptByPath(String)** - Method in interface `javax.xml.registry.BusinessQueryManager`  
Finds a Concept based on the path specified.

**findConcepts(Collection, Collection, Collection, Collection, Collection)** - Method in interface `javax.xml.registry.BusinessQueryManager`  
Finds all Concept objects that match all of the criteria specified by the parameters of this call.

**FinderException** - Exception in `javax.ejb`

The FinderException exception must be included in the throws clause of every findMETHOD(...) method of an entity Bean's home interface.

**FinderException()** - Constructor for exception `javax.ejb.FinderException`

Constructs an FinderException with no detail message.

**FinderException(String)** - Constructor for exception `javax.ejb.FinderException`

Constructs an FinderException with the specified detail message.

**FindException** - Exception in `javax.xml.registry`

A RegistryException that occurs during a find action.

**FindException()** - Constructor for exception `javax.xml.registry.FindException`

Constructs a JAXRException object with no reason or embedded Throwable.

**FindException(String)** - Constructor for exception `javax.xml.registry.FindException`

Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**FindException(String, Throwable)** - Constructor for exception `javax.xml.registry.FindException`

Constructs a JAXRException object with the specified detail message.
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

**FindException(Throwables)** - Constructor for exception
```
javax.xml.registry.FindException
```
Constructs a JAXRException object initialized with the given Throwable object.

**findOrganizations(Collection, Collection, Collection, Collection, Collection)** - Method in interface
```
javax.xml.registry.BusinessQueryManager
```
Finds all Organization objects that match all of the criteria specified by the parameters of this call.

**FindQualifier** - Interface in **javax.xml.registry**
```
FindQualifier provides various constants that identify options that effect find method behavior.
```

**findRegistryPackages(Collection, Collection, Collection, Collection)** - Method in interface **javax.xml.registry.BusinessQueryManager**
```
Finds all RegistryPackage objects that match all of the criteria specified by the parameters of this call.
```

**findServiceBindings(Key, Collection, Collection, Collection)** - Method in interface **javax.xml.registry.BusinessQueryManager**
```
Finds all ServiceBinding objects that match all of the criteria specified by the parameters of this call.
```

**findServices(Key, Collection, Collection, Collection, Collection)** - Method in interface **javax.xml.registry.BusinessQueryManager**
```
Finds all Service objects that match all of the criteria specified by the parameters of this call.
```

**fireXpathEvent(XpathEvent)** - Method in interface
```
javax.enterprise.deploy.model.XpathListener
```

**FLAGGED** - Static variable in class **javax.mail.Flags.Flag**
```
This message is flagged.
```

**FLAGS** - Static variable in class **javax.mail.FetchProfile.Item**
```
This is the Flags item.
```

**Flags** - Class in **javax.mail**
```
The Flags class represents the set of flags on a Message.
```

**Flags()** - Constructor for class **javax.mail.Flags**
```
Construct an empty Flags object.
```

**Flags(Flags)** - Constructor for class **javax.mail.Flags**
Construct a Flags object initialized with the given flags.

**Flags(Flags.Flag)** - Constructor for class `javax.mail.Flags`  
Construct a Flags object initialized with the given system flag.

**Flags(String)** - Constructor for class `javax.mail.Flags`  
Construct a Flags object initialized with the given user flag.

**flags** - Variable in class `javax.mail.internet.MimeMessage`  
The Flags for this message.

**flags** - Variable in class `javax.mail.search.FlagTerm`  
Flags object containing the flags to test.

**Flags.Flag** - Class in `javax.mail`  
This inner class represents an individual system flag.

**FLAGS_CHANGED** - Static variable in class `javax.mail.event.MessageChangedEvent`  
The message's flags changed.

**FlagTerm** - Class in `javax.mail.search`  
This class implements comparisons for Message Flags.

**FlagTerm(Flags, boolean)** - Constructor for class `javax.mail.search.FlagTerm`  
Constructor.

**FLOAT_ID** - Static variable in class `javax.faces.convert.FloatConverter`  
The message identifier of the `FacesMessage` to be created if the conversion to `Float` fails.

**FloatConverter** - Class in `javax.faces.convert`  
Converter implementation for `java.lang.Float` (and float primitive) values.

**FloatConverter()** - Constructor for class `javax.faces.convert.FloatConverter`

**FloatHolder** - Class in `javax.xml.rpc.holders`  

**FloatHolder()** - Constructor for class `javax.xml.rpc.holders.FloatHolder`

**FloatHolder(float)** - Constructor for class `javax.xml.rpc.holders.FloatHolder`

**FloatWrapperHolder** - Class in `javax.xml.rpc.holders`  

**FloatWrapperHolder()** - Constructor for class `javax.xml.rpc.holders.FloatWrapperHolder`
**FloatWrapperHolder(Float)** - Constructor for class javax.xml.rpc.holders.FloatWrapperHolder

**flush()** - Method in class javax.faces.context.ResponseWriter
Flush any output buffered by the output method to the underlying Writer or OutputStream.

**flush()** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call ResponseWriter.flush() on the wrapped ResponseWriter object.

**flush()** - Method in interface javax.persistence.EntityManager
Synchronize the persistence context to the underlying database.

**flush()** - Method in class javax.servlet.jsp.JspWriter
Flush the stream.

**flush()** - Method in class javax.servlet.jsp.tagext.BodyContent
Redefined flush() so it is not legal.

**flush()** - Method in interface javax.xml.stream.XMLEventWriter
Writes any cached events to the underlying output mechanism

**flush()** - Method in interface javax.xml.stream.XMLStreamWriter
Write any cached data to the underlying output mechanism.

**flushBuffer()** - Method in interface javax.servlet.ServletResponse
Forces any content in the buffer to be written to the client.

**flushBuffer()** - Method in class javax.servlet.ServletResponseWrapper
The default behavior of this method is to call flushBuffer() on the wrapped response object.

**FlushModeType** - Enum in javax.persistence
Flush mode setting.

**fold(int, String)** - Static method in class javax.mail.internet.MimeUtility
Fold a string at linear whitespace so that each line is no longer than 76 characters, if possible.

**folder** - Variable in class javax.mail.event.FolderEvent
The folder the event occurred on.

**Folder** - Class in javax.mail
Folder is an abstract class that represents a folder for mail messages.

**Folder(Store)** - Constructor for class javax.mail.Folder
Constructor that takes a Store object.

**folder** - Variable in class javax.mail.Message
The containing folder, if this message is obtained from a folder
**FolderAdapter** - Class in *javax.mail.event*
   The adapter which receives Folder events.

**FolderAdapter()** - Constructor for class *javax.mail.event.FolderAdapter*

**FolderClosedException** - Exception in *javax.mail*
   This exception is thrown when a method is invoked on a Messaging object and the Folder that owns that object has died due to some reason.

**FolderClosedException(Folder)** - Constructor for exception *javax.mail.FolderClosedException*
   Constructor

**FolderClosedException(Folder, String)** - Constructor for exception *javax.mail.FolderClosedException*
   Constructor

**folderCreated(FolderEvent)** - Method in class *javax.mail.event.FolderEvent*
   Invoked when a Folder is created.

**folderDeleted(FolderEvent)** - Method in class *javax.mail.event.FolderEvent*
   Invoked when a folder is deleted.

**FolderEvent** - Class in *javax.mail.event*
   This class models Folder existence events.

**FolderEvent(Object, Folder, int)** - Constructor for class *javax.mail.event.FolderEvent*
   Constructor.

**FolderEvent(Object, Folder, Folder, int)** - Constructor for class *javax.mail.event.FolderEvent*
   Constructor.

**FolderListener** - Interface in *javax.mail.event*
   This is the Listener interface for Folder events.

**FolderNotFoundException** - Exception in *javax.mail*
   This exception is thrown by Folder methods, when those methods are invoked on a non existent folder.
**FolderNotFoundException()** - Constructor for exception
javax.mail.FolderNotFoundException
   Constructs a MessagingException with no detail message.
**FolderNotFoundException(Folder)** - Constructor for exception
javax.mail.FolderNotFoundException
   Constructs a MessagingException with the specified folder.
**FolderNotFoundException(Folder, String)** - Constructor for exception
javax.mail.FolderNotFoundException
   Constructs a MessagingException with the specified folder and the specified detail message.
**FolderNotFoundException(String, Folder)** - Constructor for exception
javax.mail.FolderNotFoundException
   Constructs a MessagingException with the specified detail message and the specified folder.
**folderRenamed(FolderEvent)** - Method in class
javax.mail.event.FolderAdapter

**folderRenamed(FolderEvent)** - Method in interface
javax.mail.event.FolderListener
   Invoked when a folder is renamed.
**forget(Xid)** - Method in interface javax.resource.spi.XATerminator
   Tells the resource manager to forget about a heuristically completed transaction branch.
**forget(Xid)** - Method in interface javax.transaction.xa.XAResource
   Tells the resource manager to forget about a heuristically completed transaction branch.
**FORM_AUTH** - Static variable in class
javax.faces.context.ExternalContext
   String identifier for FORM authentication.
**FORM_AUTH** - Static variable in interface
javax.servlet.http.HttpServletRequest
   String identifier for Form authentication.
**format(Date, StringBuffer, FieldPosition)** - Method in class
javax.mail.internet.MailDateFormat
   Formats the given date in the format specified by draft-ietf-drums-msg-fmt-08 in the current TimeZone.
**forward(String)** - Method in class javax.servlet.jsp.PageContext
   This method is used to re-direct, or "forward" the current ServletRequest and ServletResponse to another active component.
in the application.

**forward(ServletRequest, ServletResponse)** - Method in interface javax.servlet.RequestDispatcher
  Forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.

**FromStringTerm** - Class in javax.mail.search
  This class implements string comparisons for the From Address header.

**FromStringTerm(String)** - Constructor for class javax.mail.searchFromStringTerm Constructor.

**FromTerm** - Class in javax.mail.search
  This class implements comparisons for the From Address header.

**FromTerm(Address)** - Constructor for class javax.mail.search.FromTerm Constructor.

**fullURL** - Variable in class javax.mail.URLName
  The full version of the URL

**FunctionInfo** - Class in javax.servlet.jsp.tagext
  Information for a function in a Tag Library.

**FunctionInfo(String, String, String)** - Constructor for class javax.servlet.jsp.tagext.FunctionInfo Constructor for FunctionInfo.

**FunctionMapper** - Class in javax.el
  The interface to a map between EL function names and methods.

**FunctionMapper()** - Constructor for class javax.el.FunctionMapper

**FunctionMapper** - Interface in javax.servlet.jsp.el
  Deprecated. As of JSP 2.1, replaced by FunctionMapper

**functions** - Variable in class javax.servlet.jsp.tagext.TagLibraryInfo
  An array describing the functions that are defined in this tag library.
**GE** - Static variable in class javax.mail.search. *ComparisonTerm*

**Generated** - Annotation Type in *javax.annotation*

The Generated annotation is used to mark source code that has been generated.

**GeneratedValue** - Annotation Type in *javax.persistence*

Provides for the specification of generation strategies for the values of primary keys.

**generateSchema(SchemaOutputResolver)** - Method in class javax.xml.bind. *JAXBContext*

Generates the schema documents for this context.

**GenerationType** - Enum in *javax.persistence*

Defines the types of primary key generation.

**GenericCredential** - Interface in *javax.resource.spi.security*

*Deprecated.* The preferred way to represent generic credential information is via the org.ietf.jgss.GSSCredential interface in J2SE Version 1.4, which provides similar functionality.

**GenericHandler** - Class in *javax.xml.rpc.handler*

The javax.xml.rpc.handler.GenericHandler class implements the Handler interface.

**GenericHandler()** - Constructor for class javax.xml.rpc.handler. *GenericHandler*

Default constructor.

**GenericServlet** - Class in *javax.servlet*

Defines a generic, protocol-independent servlet.

**GenericServlet()** - Constructor for class javax.servlet. *GenericServlet*

Does nothing.

**get(String)** - Method in class javax.activation. *MimeTypeParameterList*

Retrieve the value associated with the given name, or null if there is no current association.

**get(String)** - Method in class javax.mail.internet. *ParameterList*

Returns the value of the specified parameter.

**get(Object)** - Method in class javax.xml.soap. *SOAPConnection*

Gets a message from a specific endpoint and blocks until it receives,

**getAccept()** - Method in class javax.faces.component.html. *HtmlForm*
Return the value of the accept property.

```java
getAcceptCharset() - Method in class
javax.faces.component.html.HtmlForm
```

Return the value of the acceptcharset property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlCommandButton
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlCommandLink
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlInputSecret
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlInputText
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlInputTextarea
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlOutputLabel
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlOutputLink
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
```

Return the value of the accesskey property.

```java
getAccesskey() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
```
Return the value of the accesskey property.

**getAccesskey()** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the accesskey property.

**getAccesskey()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the accesskey property.

**getAccessURI()** - Method in interface
javax.xml.registry.infomodel.ServiceBinding
Gets the URI that gives access to the service via this binding.

**getAcknowledgeMode()** - Method in interface javax.jms.Session
Returns the acknowledgement mode of the session.

**getAction()** - Method in interface
javax.enterprise.deploy.spi.status.DeploymentStatus
Retrieve the deployment ActionType for this event.

**getAction()** - Method in class javax.faces.component.ActionSource
Deprecated. This has been replaced by
ActionSource2.getActionExpression().

**getAction()** - Method in class javax.faces.component.UICommand
Deprecated. This has been replaced by
UICommand.getActionExpression().

**getActionExpression()** - Method in interface
javax.faces.component.ActionSource2
Return the MethodExpression pointing at the application action to be
invoked, if this UIComponent is activated by the user, during the Apply
Request Values or Invoke Application phase of the request
processing lifecycle, depending on the value of the immediate
property.

**getActionExpression()** - Method in class
javax.faces.component.UICommand

**getActionListener()** - Method in class javax.faces.application.Application
Return the default ActionListener to be registered for all
ActionSource components in this application.

**getActionListener()** - Method in interface
javax.faces.component.ActionSource
Deprecated. Use ActionSource.getActionListeners() instead.

**getActionListener()** - Method in class
javax.faces.component.UICommand
Deprecated. Use `UICommand.getActionListeners()` instead.

**getActionListeners()** - Method in interface
javax.faces.component.ActionSource
   Return the set of registered `ActionListener`s for this `ActionSource` instance.

**getActionListeners()** - Method in class
javax.faces.component.UICommand

**getActions()** - Method in class
javax.security.jacc.EJBMethodPermission
   Returns a String containing a canonical representation of the actions of this EJBMMethodPermission.

**getActions()** - Method in class
javax.security.jacc.EJBRoleRefPermission
   Returns a canonical String representation of the actions of this EJBRoleRefPermission.

**getActions()** - Method in class
javax.security.jacc.WebResourcePermission
   Returns a canonical String representation of the actions of this WebResourcePermission.

**getActions()** - Method in class
javax.security.jacc.WebRoleRefPermission
   Returns a canonical String representation of the actions of this WebRoleRefPermission.

**getActions()** - Method in class
javax.security.jacc.WebUserDataPermission
   Returns a canonical String representation of the actions of this WebUserDataPermission.

**getActionType(int)** - Static method in class
javax.enterprise.deploy.shared.ActionType
   Return an object of the specified value.

**getActionURL(FacesContext, String)** - Method in class
javax.faces.application.ViewHandler
   Return a URL suitable for rendering (after optional encoding performed by the `encodeActionURL()` method of `ExternalContext`) that selects the specified view identifier.

**getActionURL(FacesContext, String)** - Method in class
javax.faces.application.ViewHandlerWrapper
   The default behavior of this method is to call
   `ViewHandler.getActionURL(javax.faces.context.FacesContext,`
on the wrapped `ViewHandler` object.

**getActiveCount()** - Method in interface `javax.management.j2ee.statistics.JTASstats`
Number of active transactions.

**getActor()** - Method in interface `javax.xml.soap.SOAPHeaderElement`
Returns the uri of the `actor` attribute of this `SOAPHeaderElement`.

**getAdaptee()** - Method in class `javax.servlet.jsp.tagext.TagAdapter`
Gets the tag that is being adapted to the Tag interface.

**getAdapter(Class<A>)** - Method in class `javax.xml.bind.helpers.AbstractMarshallerImpl`
Gets the adapter associated with the specified type.

**getAdapter(Class<A>)** - Method in class `javax.xml.bind.helpers.AbstractUnmarshallerImpl`
Gets the adapter associated with the specified type.

**getAdapter(Class<A>)** - Method in interface `javax.xml.bind.Marshaller`
Gets the adapter associated with the specified type.

**getAdapter(Class<A>)** - Method in interface `javax.xml.bind.Unmarshaller`
Gets the adapter associated with the specified type.

**getAdapterName()** - Method in interface `javax.resource.cci.ResourceAdapterMetaData`
Gets a tool displayable name of the resource adapter.

**getAdapterShortDescription()** - Method in interface `javax.resource.cci.ResourceAdapterMetaData`
Gets a tool displayable short description of the resource adapter.

**getAdapterVendorName()** - Method in interface `javax.resource.cci.ResourceAdapterMetaData`
Gets the name of the vendor that has provided the resource adapter.

**getAdapterVersion()** - Method in interface `javax.resource.cci.ResourceAdapterMetaData`
Gets the version of the resource adapter.

**getAddress()** - Method in class `javax.mail.internet.InternetAddress`
Get the email address.

**getAddress()** - Method in class `javax.mail.search.AddressTerm`
Return the address to match with.

**getAddress()** - Method in interface `javax.xml.registry.infomodel.EmailAddress`
Returns the email address for this object.

**getAfterPhaseListener()** - Method in class
javax.faces.component.**UIViewRoot**

**getAllAttributes()** - Method in interface javax.xml.soap.**SOAPElement**
Returns an **Iterator** over all of the attribute **Name** objects in this **SOAPElement** object.

**getAllAttributesAsQNames()** - Method in interface javax.xml.soap.**SOAPElement**
Returns an **Iterator** over all of the attributes in this **SOAPElement** as QName objects.

**getAllCommands(String)** - Method in class javax.activation.**CommandMap**
Get all the available commands for this type.

**getAllCommands(String, DataSource)** - Method in class javax.activation.**CommandMap**
Get all the available commands for this type.

**getAllCommands()** - Method in class javax.activation.**DataHandler**
Return all the commands for this type of data.

**getAllCommands(String)** - Method in class javax.activation.**MailcapCommandMap**
Get all the available commands in all mailcap files known to this instance of MailcapCommandMap for this MIME type.

**getAllHeaderLines()** - Method in class javax.mail.internet.**InternetHeaders**
Return all the header lines as an Enumeration of Strings.

**getAllHeaderLines()** - Method in class javax.mail.internet.**MimeBodyPart**
Get all header lines as an Enumeration of Strings.

**getAllHeaderLines()** - Method in class javax.mail.internet.**MimeMessage**
Get all header lines as an Enumeration of Strings.

**getAllHeaderLines()** - Method in interface javax.mail.internet.**MimePart**
Get all header lines as an Enumeration of Strings.

**getAllHeaders()** - Method in class javax.mail.internet.**InternetHeaders**
Return all the headers as an Enumeration of Header objects.

**getAllHeaders()** - Method in class javax.mail.internet.**MimeBodyPart**
Return all the headers from this Message as an enumeration of Header objects.

**getAllHeaders()** - Method in class javax.mail.internet.**MimeMessage**
Return all the headers from this Message as an enumeration of Header objects.

**getAllHeaders()** - Method in interface javax.mail.**Part**
Return all the headers from this part as an Enumeration of Header objects.

**getAllHeaders()** - Method in class `javax.xml.soap.MimeHeaders`
Returns all the `MimeHeaders` in this `MimeHeaders` object.

**getAllMimeHeaders()** - Method in class `javax.xml.soap.AttachmentPart`
Retrieves all the headers for this `AttachmentPart` object as an iterator over the `MimeHeader` objects.

**getAllMimeHeaders()** - Method in class `javax.xml.soap.SOAPPart`
Retrieves all the headers for this `SOAPPart` object as an iterator over the `MimeHeader` objects.

**getAllRecipients()** - Method in class `javax.mail.internet.MimeMessage`
Get all the recipient addresses for the message.

**getAllRecipients()** - Method in class `javax.mail.Message`
Get all the recipient addresses for the message.

**getAlt()** - Method in class `javax.faces.component.html.HtmlCommandButton`
Return the value of the `alt` property.

**getAlt()** - Method in class `javax.faces.component.html.HtmlGraphicImage`
Return the value of the `alt` property.

**getAlt()** - Method in class `javax.faces.component.html.HtmlInputSecret`
Return the value of the `alt` property.

**getAlt()** - Method in class `javax.faces.component.html.HtmlInputText`
Return the value of the `alt` property.

**getApplication()** - Method in class `javax.faces.application.ApplicationFactory`
Create (if needed) and return an `Application` instance for this web application.

**getApplication()** - Method in class `javax.faces.context.FacesContext`
Return the `Application` instance associated with this web application.

**getApplicationMap()** - Method in class `javax.faces.context.ExternalContext`
Return a mutable `Map` representing the application scope attributes for the current application.

**getAreaCode()** - Method in interface `javax.xml.registry.infomodel.TelephoneNumber`
Gets the area code.

**getAsObject(FacesContext, UIComponent, String)** - Method in class
javax.faces.convert.BigDecimalConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.BigIntegerConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.BooleanConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.ByteConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.CharacterConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.Converter
  Convert the specified string value, which is associated with the specified UIComponent, into a model data object that is appropriate for being stored during the Apply Request Values phase of the request processing lifecycle.

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.DateTimeConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.DoubleConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.EnumConverter
  Convert the value argument to one of the enum constants of the class provided in our constructor.

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.FloatConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.IntegerConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.LongConverter
getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.NumberConverter

getAsObject(FacesContext, UIComponent, String) - Method in class javax.faces.convert.ShortConverter

getAssociatedObjects() - Method in interface javax.xml.registry.infomodel.RegistryObject

    Returns the collection of RegistryObject instances associated with this object.

getAssociations() - Method in interface javax.xml.registry.infomodel.RegistryObject

    Gets all Associations where this object is source.

getAssociationType() - Method in interface javax.xml.registry.infomodel.Association

    Gets the association type for this Association.

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.BigDecimalConverter

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.BigIntegerConverter

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.BooleanConverter

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.ByteConverter

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.CharacterConverter

getAsString(FacesContext, UIComponent, Object) - Method in interface javax.faces.convert.Converter

    Convert the specified model object value, which is associated with the specified UIComponent, into a String that is suitable for being included in the response generated during the Render Response phase of the request processing lifecycle.

getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.DateTimeConverter
getAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.DoubleConverter

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.EnumConverter
   Convert the enum constant given by the value argument into a String.

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.FloatConverter

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.IntegerConverter

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.LongConverter

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.NumberConverter

gAsString(FacesContext, UIComponent, Object) - Method in class javax.faces.convert.ShortConverter

gGetAttachment(SOAPElement) - Method in class javax.xml.soap.SOAPMessage
   Returns an AttachmentPart object that is associated with an attachment that is referenced by this SOAPElement or null if no such attachment exists.

gGetAttachmentAsByteArray(String) - Method in class javax.xml.bind.attachment.AttachmentUnmarshaller
   Retrieve the attachment identified by content-id, cid, as a byte[].

gGetAttachmentAsDataHandler(String) - Method in class javax.xml.bind.attachment.AttachmentUnmarshaller
   Lookup MIME content by content-id, cid, and return as a DataHandler.

gGetAttachmentMarshaller() - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl

gGetAttachmentMarshaller() - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl
javax.xml.bind.**Marshaller**

**getAttachments()** - Method in class javax.xml.soap.SOAPMessage
Retrieves all the AttachmentPart objects that are part of this SOAPMessage object.

**getAttachments(MimeHeaders)** - Method in class javax.xml.soap.SOAPMessage
Retrieves all the AttachmentPart objects that have header entries that match the specified headers.

**getAttachmentUnmarshaller()** - Method in class javax.xml.bind.helpers.AbstractUnmarshallerImpl

**getAttachmentUnmarshaller()** - Method in interface javax.xml.bind.Unmarshaller

**getAttribute(ObjectName, String)** - Method in interface javax.management.j2ee.Management
Gets the value of a specific attribute of a named managed object.

**getAttribute(String)** - Method in interface javax.servlet.http.HttpSession
Returns the object bound with the specified name in this session, or null if no object is bound under the name.

**getAttribute(String)** - Method in class javax.servlet.jsp.JspContext
Returns the object associated with the name in the page scope or null if not found.

**getAttribute(String, int)** - Method in class javax.servlet.jsp.JspContext
Return the object associated with the name in the specified scope or null if not found.

**getAttribute(String)** - Method in class javax.servlet.jsp.tagext.TagData
The value of the attribute.

**getAttribute(String)** - Method in interface javax.servlet.ServletContext
Returns the servlet container attribute with the given name, or null if there is no attribute by that name.

**getAttribute(String)** - Method in interface javax.servlet.ServletRequest
Returns the value of the named attribute as an object, or null if no attribute of the given name exists.

**getAttribute(String)** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to call getAttribute(String name) on the wrapped request object.
**getAttributeByQName(QName)** - Method in interface
javax.xml.stream.eventsStartElement
  Returns the attribute referred to by this name

**getAttributeCount()** - Method in class
javax.xml.stream.util.StreamReaderDelegate

**getAttributeCount()** - Method in interface
javax.xml.stream.XMLStreamReader
  Returns the count of attributes on this START_ELEMENT, this method is only valid on a START_ELEMENT or ATTRIBUTE.

**getAttributeLocalName(int)** - Method in class
javax.xml.stream.util.StreamReaderDelegate

**getAttributeLocalName(int)** - Method in interface
javax.xml.stream.XMLStreamReader
  Returns the localName of the attribute at the provided index

**getAttributeName(int)** - Method in class
javax.xml.stream.util.StreamReaderDelegate

**getAttributeName(int)** - Method in interface
javax.xml.stream.XMLStreamReader
  Returns the qname of the attribute at the provided index

**getAttributeNames()** - Method in interface
javax.enterprise.deploy.model.DDBean
  Returns the list of attribute names associated with the XML element.

**getAttributeNames()** - Method in interface javax.servlet.http.HttpSession
  Returns an Enumeration of String objects containing the names of all the objects bound to this session.

**getAttributeNames()** - Method in interface javax.servlet.ServletContext
  Returns an Enumeration containing the attribute names available within this servlet context.

**getAttributeNames()** - Method in interface javax.servlet.ServletRequest
  Returns an Enumeration containing the names of the attributes available to this request.

**getAttributeNames()** - Method in class
javax.servlet.ServletRequestWrapper
  The default behavior of this method is to return getAttributeNames() on the wrapped request object.

**getAttributeNamesInScope(int)** - Method in class
enumerate all the attributes in a given scope.

**getAttributeNamespace(int)** - Method in class
javax.xml.stream.util.StreamReaderDelegate

Returns the namespace of the attribute at the provided index

**getAttributePrefix(int)** - Method in class
javax.xml.stream.util.StreamReaderDelegate

Returns the prefix of this attribute at the provided index

**getAttributes()** - Method in class javax.faces.component.UIComponent

Return a mutable Map representing the attributes (and properties, see below) associated with this UIComponent, keyed by attribute name (which must be a String).

**getAttributes()** - Method in class
javax.faces.component.UIComponentBase

**getAttributes(ObjectName, String[])** - Method in interface
javax.management.j2ee.Management

Enables the values of several attributes of a named managed object.

**getAttributes()** - Method in class javax.servlet.jsp.tagext.TagData

Enumerates the attributes.

**getAttributes()** - Method in class javax.servlet.jsp.tagext.TagInfo

Attribute information (in the TLD) on this tag.

**getAttributes()** - Method in interface
javax.xml.stream.events.StartElement

Returns an Iterator of non-namespace declared attributes declared on this START_ELEMENT, returns an empty iterator if there are no attributes.

**getAttributesScope(String)** - Method in class
javax.servlet.jsp.JspContext

Get the scope where a given attribute is defined.

**getAttributeString(String)** - Method in class
javax.servlet.jsp.tagext.TagData

Get the value for a given attribute.
**getAttributeType(int)** - Method in class
javax.xml.stream.util.StreamReaderDelegate

g**etAttributeType(int)** - Method in interface
javax.xml.stream.XMLStreamReader
    Returns the XML type of the attribute at the provided index

**getAttributeValue(String)** - Method in interface
javax.enterprise.deploy.model.DDBean
    Returns the string value of the named attribute.

**getAttributeValue(Name)** - Method in interface
javax.xml.soap.SOAPElement
    Returns the value of the attribute with the specified name.

**getAttributeValue(QName)** - Method in interface
javax.xml.soap.SOAPElement
    Returns the value of the attribute with the specified qname.

**getAttributeValue(String, String)** - Method in class
javax.xml.stream.util.StreamReaderDelegate
    Returns the normalized attribute value of the attribute with the namespace and localName. If the namespaceURI is null the namespace is not checked for equality.

**getAttributeValue(int)** - Method in interface
javax.xml.stream.XMLStreamReader
    Returns the value of the attribute at the index

**getAuditTrail()** - Method in interface
javax.xml.registry.infomodel.RegistryObject
    Returns the complete audit trail of all requests that effected a state change in this object as an ordered Collection of AuditableEvent objects.

**getAuthType()** - Method in class
javax.faces.context.ExternalContext
    Return the name of the authentication scheme used to authenticate the current user, if any; otherwise, return null.

**getAuthType()** - Method in interface
javax.servlet.http.HttpServletRequest
Returns the name of the authentication scheme used to protect the servlet.

**getAuthType()** - Method in class
javax.servlet.http.HttpServletRequestWrapper

The default behavior of this method is to return getAuthType() on the wrapped request object.

**getAutocomplete()** - Method in class
javax.faces.component.html(HtmlInputSecret)

Return the value of the autocomplete property.

**getAutocomplete()** - Method in class
javax.faces.component.html.HtmlInputText

Return the value of the autocomplete property.

**getAvailableModules(ModuleType[], Target[])** - Method in interface
javax.enterprise.deploy.spi.DeploymentManager

Retrieve the list of all J2EE application modules running or not running on the identified targets.

**getBase64Content()** - Method in class javax.xml.soap.AttachmentPart

Returns an InputStream which can be used to obtain the content of AttachmentPart as Base64 encoded character data, this method would base64 encode the raw bytes of the attachment and return.

**getBaseType()** - Method in class javax.activation.MimeType

Return a String representation of this object without the parameter list.

**getBaseType()** - Method in class javax.mail.internet.ContentType

Return the MIME type string, without the parameters.

**getBaseURI()** - Method in interface
javax.xml.stream.events.EntityDeclaration

Get the base URI for this reference or null if this information is not available

**getBean(CommandInfo)** - Method in class javax.activation.DataHandler

A convenience method that takes a CommandInfo object and instantiates the corresponding command, usually a JavaBean component.

**getBean()** - Method in class javax.enterprise.deploy.model.XpathEvent

The bean being added/removed/changed.

**getBeanProperty(String)** - Method in class
javax.el.BeanELResolver.BeanProperties

**getBeforePhaseListener()** - Method in class
javax.faces.component.**UIViewRoot**

**getBgcolor()** - Method in class
javax.faces.component.html.**HtmlDataTable**
Return the value of the bgcolor property.
**getBgcolor()** - Method in class
javax.faces.component.html.**HtmlPanelGrid**
Return the value of the bgcolor property.

**getBinding()** - Method in interface javax.xml.ws.**BindingProvider**
Get the Binding for this binding provider.

**getBinding()** - Method in class javax.xml.ws.**Endpoint**
Returns the binding for this endpoint.

**getBindingID()** - Method in interface javax.xml.ws.handler.**PortInfo**
Gets the URI identifying the binding used by the port being accessed.

**getBody()** - Method in interface javax.xml.soap.**SOAPEnvelope**
Returns the SOAPBody object associated with this SOAPEnvelope object.

**getBodyContent()** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**
**getBodyContent()** - Method in class
javax.servlet.jsp.tagext.**BodyTagSupport**
Get current bodyContent.

**getBodyContent()** - Method in class
javax.servlet.jsp.tagext.**TagInfo**
The bodycontent information for this tag.

**getBodyLength()** - Method in interface javax.jms.**BytesMessage**
Gets the number of bytes of the message body when the message is in read-only mode.

**getBodyPart(int)** - Method in class javax.mail.internet.**MimeMultipart**
Get the specified BodyPart.

**getBodyPart(String)** - Method in class javax.mail.internet.**MimeMultipart**
Get the MimeBodyPart referred to by the given ContentID (CID).

**getBodyPart(int)** - Method in class javax.mail.**Multipart**
Get the specified Part.

**getBodyPart(int)** - Method in class javax.mail.**MultipartDataSource**
Get the specified Part.

**getBoolean(String)** - Method in interface javax.jms.**MapMessage**
Returns the boolean value with the specified name.

**getBooleanProperty(String)** - Method in interface javax.jms.**Message**
Returns the value of the boolean property with the specified name.

**getBorder()** - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the border property.

**getBorder()** - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the border property.

**getBorder()** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the border property.

**getBorder()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the border property.

**getBranchQualifier()** - Method in interface javax.transaction.xa.Xid
Obtain the transaction branch identifier part of XID as an array of bytes.

**getBufferSize()** - Method in class javax.servlet.jsp.JspWriter
This method returns the size of the buffer used by the JspWriter.

**getBufferSize()** - Method in interface javax.servlet.ServletResponse
Returns the actual buffer size used for the response.

**getBufferSize()** - Method in class javax.servlet.ServletResponseWrapper
The default behavior of this method is to return getBufferSize() on the wrapped response object.

**getBuilder()** - Method in class
javax.xml.bind.annotation.W3CDomHandler

**getBulkResponse(String)** - Method in interface
javax.xml.registry.RegistryService
Returns the BulkResponse associated with specified requestId.

**getBusinessLifeCycleManager()** - Method in interface
javax.xml.registry.RegistryService
Returns the BusinessLifeCycleManager object implemented by the JAXR provider.

**getBusinessObject(Class<T>)** - Method in interface
javax.ejb.SessionContext
Obtain an object that can be used to invoke the current bean through the given business interface.

**getBusinessQueryManager()** - Method in interface
javax.xml.registry.RegistryService
Returns the BusinessQueryManager object implemented by the JAXR provider.

**getByte(String)** - Method in interface javax.jms.MapMessage

Returns the byte value with the specified name.

**getByteProperty(String)** - Method in interface javax.jms.Message

Returns the value of the byte property with the specified name.

**getBytes(String)** - Method in interface javax.jms.MapMessage

Returns the byte array value with the specified name.

**getCallerIdentity()** - Method in interface javax.ejb.EJBContext

*Deprecated. Use Principal getCallerPrincipal() instead.*

**getCallerPrincipal()** - Method in interface javax.ejb.EJBContext

Obtain the java.security.Principal that identifies the caller.

**getCalls(QName)** - Method in interface javax.xml.rpc.Service

Gets an array of preconfigured call objects for invoking operations on the specified port.

**getCapabilityLevel()** - Method in interface javax.xml.registry.CapabilityProfile

Gets the capability level supported by the JAXR provider.

**getCapabilityProfile()** - Method in interface javax.xml.registry.RegistryService

Returns the CapabilityProfile for the JAXR provider.

**getCaptionClass()** - Method in class javax.faces.component.html.HtmlDataTable

Return the value of the captionClass property.

**getCaptionClass()** - Method in class javax.faces.component.html.HtmlPanelGrid

Return the value of the captionClass property.

**getCaptionStyle()** - Method in class javax.faces.component.html.HtmlDataTable

Return the value of the captionStyle property.

**getCaptionStyle()** - Method in class javax.faces.component.html.HtmlPanelGrid

Return the value of the captionStyle property.

**getCause()** - Method in exception javax.faces.FacesException

Return the cause of this exception, or null if the cause is nonexistent or unknown.

**getCause()** - Method in exception javax.mail.MessagingException

Overrides the getCause method of Throwable to return the next exception in the chain of nested exceptions.
**get Cause()** - Method in exception javax.xml.bind.JAXBException

Returns the Throwable embedded in this JAXBException if there is one.

**get Cause()** - Method in exception javax.xml.registry.JAXRException

Returns the Throwable object embedded in this JAXRException if there is one.

**get Caused By Exception()** - Method in exception javax.ejb.EJBException

Obtain the exception that caused the EJBException being thrown.

**getCellpadding()** - Method in class javax.faces.component.html.HtmlDataTable

Return the value of the cellpadding property.

**getCellpadding()** - Method in class javax.faces.component.html.HtmlPanelGrid

Return the value of the cellpadding property.

**getCellspacing()** - Method in class javax.faces.component.html.HtmlDataTable

Return the value of the cellspacing property.

**getCellspacing()** - Method in class javax.faces.component.html.HtmlPanelGrid

Return the value of the cellspacing property.

**get Change Event()** - Method in class javax.enterprise.deploy.model.XpathEvent

**getChar(String)** - Method in interface javax.jms.MapMessage

Returns the Unicode character value with the specified name.

**get Character Encoding()** - Method in class javax.faces.context.ResponseWriter

Return the character encoding (such as "ISO-8859-1") for this ResponseWriter.

**get Character Encoding()** - Method in class javax.faces.context.ResponseWriterWrapper

The default behavior of this method is to call ResponseWriter.getCharacterEncoding() on the wrapped ResponseWriter object.

**get Character Encoding()** - Method in interface javax.servlet.ServletRequest

Returns the name of the character encoding used in the body of this
request.


The default behavior of this method is to return `getCharacterEncoding()` on the wrapped request object.


Returns the name of the character encoding (MIME charset) used for the body sent in this response.


The default behavior of this method is to return `getCharacterEncoding()` on the wrapped response object.


Returns the encoding style of the XML data.

**getCharacterEncodingScheme()** - Method in class [javax.xml.stream.utilStreamReaderDelegate](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/stream/javax/xml/stream/util/StreamReaderDelegate.html)


Returns the character encoding declared on the xml declaration
Returns null if none was declared.

**getCharacterOffset()** - Method in interface [javax.xml.stream.Location](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml/stream/javax/xml/stream/Location.html)

Returns the byte or character offset into the input source this location is pointing to.


Return the value of the *charset* property.


Return the value of the *charset* property.


Gets the canonical name for the charset for this object.

**getChildBean(String)** - Method in interface [javax.enterprise.deploy.model/DDBean](https://docs.oracle.com/en/java/javase/11/docs/api/java.xml.registry.infomodel/javax/xml/registry/infomodel/DDBean.html)

Return a list of DDBeans based upon the XPath.
javax.enterprise.deploy.model.\texttt{DeployableObject}

Return an array of standard beans representing the XML content returned based upon the XPath.

\texttt{getChildBean(ModuleType, String)} - Method in interface javax.enterprise.deploy.model.\texttt{J2eeApplicationObject}

Return a list of \texttt{DDBean} based upon an XPath; all deployment descriptors of the specified type are searched.

\texttt{getChildConceptCount()} - Method in interface javax.xml.registry.infomodel.\texttt{ClassificationScheme}

Gets number of children.

\texttt{getChildConceptCount()} - Method in interface javax.xml.registry.infomodel.\texttt{Concept}

Gets number of children.

\texttt{getChildCount()} - Method in class javax.faces.component.\texttt{UIComponent}

Return the number of child \texttt{UIComponent}s that are associated with this \texttt{UIComponent}.

\texttt{getChildren()} - Method in class javax.faces.component.\texttt{UIComponentBase}

Return a mutable List representing the child \texttt{UIComponent}s associated with this component.
**getChildren()** - Method in class
javax.faces.component.UIComponentBase

**getChildrenConcepts()** - Method in interface
javax.xml.registry.infomodel.ClassificationScheme
   Gets all immediate children Concepts.

**getChildrenConcepts()** - Method in interface
javax.xml.registry.infomodel.Concept
   Gets all immediate children Concepts.

**getChildTargetModuleID()** - Method in interface
javax.enterprise.deploy.spi.TargetModuleID
   Retrieve a list of identifiers of the children of this deployed module.

**getCity()** - Method in interface
javax.xml.registry.infomodel.PostalAddress
   Returns the city.

**getClassFromScope(String)** - Method in interface
javax.enterprise.deploy.model.DeployableObject
   Retrieve the specified class from this deployable module.

**getClassifications()** - Method in interface
javax.xml.registry.infomodel.RegistryObject
   Gets the Classification instances that classify this object.

**getClassificationScheme()** - Method in interface
javax.xml.registry.infomodel.Classification
   Gets the ClassificationScheme that is used in classifying the object.

**getClassificationScheme()** - Method in interface
javax.xml.registry.infomodel.Concept
   Gets the ClassificationScheme that this Concept is a descendent of.

**getClassifiedObject()** - Method in interface
javax.xml.registry.infomodel.Classification
   Gets the Object that is being classified.

**getClassLoader()** - Method in interface
javax.persistence.spi.PersistenceUnitInfo
   Returns ClassLoader that the provider may use to load any classes, resources, or open URLs.

**getClassName()** - Method in class
javax.mail.Provider
   Returns name of the class that implements the protocol.

**getClassName()** - Method in class
javax.servlet.jsp.tagext.TagVariableInfo
   The body of the <variable-class> element.
**getClassName()** - Method in class `javax.servlet.jsp.tagext.VariableInfo`
Returns the type of this variable.

**getClientConfiguration(TargetModuleID)** - Method in interface `javax.enterprise.deploy.spi.status.ProgressObject`
Return the ClientConfiguration object associated with the TargetModuleID.

**getClientId(FacesContext)** - Method in class `javax.faces.component.UIComponent`
Return a client-side identifier for this component, generating one if necessary.

**getClientId(FacesContext)** - Method in class `javax.faces.component.UIComponentBase`

**getClientId(FacesContext)** - Method in class `javax.faces.component.UIData`
Return a client identifier for this component that includes the current value of the `rowIndex` property, if it is not set to -1.

**getClientID()** - Method in interface `javax.jms.Connection`
Gets the client identifier for this connection.

**getClientIdsWithMessages()** - Method in class `javax.faces.context.FacesContext`
Return an Iterator over the client identifiers for which at least one `FacesMessage` has been queued.

**getCloseCount()** - Method in interface `javax.management.j2ee.statistics.JCAConnectionPoolStats`
The number of connections closed

**getCloseCount()** - Method in interface `javax.management.j2ee.statistics.JDBCConnectionPoolStats`
Number of connections closed.

**getCollection()** - Method in interface `javax.xml.registry.BulkResponse`
Get the Collection of objects returned as a response of a bulk operation.

**getCols()** - Method in class `javax.faces.component.html.HtmlInputTextarea`
Return the value of the `cols` property.

**getColumnClasses()** - Method in class `javax.faces.component.html.HtmlDataTable`
Return the value of the `columnClasses` property.

**getColumnClasses()** - Method in class
javax.faces.component.html.HtmlPanelGrid
  Return the value of the columnClasses property.

getColumnNumber() - Method in class
gavax.xml.bind.helpers.ValidationEventLocatorImpl
  getColumnNumber() - Method in interface
gavax.xml.bindhelpers.ValidationEventLocator
    Return the column number if available

columnNumber() - Method in interface
gavax.xml.stream.Location
  Return the column number where the current event ends, returns -1 if none is available.

columns() - Method in class
gavax.faces.component.html.HtmlPanelGrid
  Get the value of the columns property.

getCommand(String, String) - Method in class
gavax.activation.CommandMap
  Get the default command corresponding to the MIME type.

getCommand(String, String, DataSource) - Method in class
gavax.activation.CommandMap
  Get the default command corresponding to the MIME type.

getCommand(String) - Method in class
gavax.activation.DataHandler
  Get the command cmdName.

getCommand(String, String) - Method in class
gavax.activation.MailcapCommandMap
  Get the command corresponding to cmdName for the MIME type.

getCommand() - Method in interface
gavax.enterprise.deploy.spi.status.DeploymentStatus
  Retrieve the deployment CommandType of this event.

getCommandClass() - Method in class
gavax.activation.CommandInfo
  Return the command's class name.

getCommandName() - Method in class
gavax.activation.CommandInfo
  Return the command verb.

getCommandObject(DataHandler, ClassLoader) - Method in class
gavax.activation.CommandInfo
  Return the instantiated JavaBean component.

getCommandType(int) - Static method in class
gavax.enterprise.deploy.shared.CommandType
  Return an object of the specified value.

getComment() - Method in class
gavax.servlet.http.Cookie
Returns the comment describing the purpose of this cookie, or null if the cookie has no comment.

`getCommittedCount()` - Method in interface `javax.management.j2ee.statistics.JTASstats`
Number of committed transactions.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.ArrayELResolver`
If the base object is a Java language array, returns the most general type that this resolver accepts for the property argument.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.BeanELResolver`
If the base object is not null, returns the most general type that this resolver accepts for the property argument.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.CompositeELResolver`
Returns the most general type that this resolver accepts for the property argument, given a base object.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.ELResolver`
Returns the most general type that this resolver accepts for the property argument, given a base object.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.ListELResolver`
If the base object is a list, returns the most general type that this resolver accepts for the property argument.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.MapELResolver`
If the base object is a map, returns the most general type that this resolver accepts for the property argument.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.el.ResourceBundleELResolver`
If the base object is a ResourceBundle, returns the most general type that this resolver accepts for the property argument.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.servlet.jsp.el.ImplicitObjectELResolver`
If the base object is null, returns String.class.

`getCommonPropertyType(ELContext, Object)` - Method in class `javax.servlet.jsp.el.ScopedAttributeELResolver`
If the base object is null, returns String.class.
**getComparison()** - Method in class `javax.mail.search.DateTerm`
  Return the type of comparison.

**getComparison()** - Method in class `javax.mail.search.IntegerComparisonTerm`
  Return the type of comparison.

**getComponent()** - Method in class `javax.faces.event.FacesEvent`
  Return the source `UIComponent` that sent this event.

**getComponentInstance()** - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
  Return the `UIComponent` instance that is associated with this tag instance.

**getComponentInstance()** - Method in class `javax.faces.webapp.UIComponentTagBase`
  Return the `UIComponent` instance that is associated with this tag instance.

**getComponentStateToRestore(FacesContext)** - Method in class `javax.faces.render.ResponseStateManager`
  **Deprecated.** This method has been replaced by

**getComponentStateToSave(FacesContext)** - Method in class `javax.faces.application.StateManager`
  **Deprecated.** the distinction between tree structure and component state is now an implementation detail. The default implementation returns `null`.

**getComponentStateToSave(FacesContext)** - Method in class `javax.faces.applicationStateManagerWrapper`
  The default behavior of this method is to call
  `StateManager.getComponentStateToSave(javax.faces.context.FacesContext)` on the wrapped `StateManager` object.

**getComponentType()** - Method in class `javax.faces.webapp.UIComponentTagBase`
  Return the component type for the component that is or will be bound to this tag.

**getComponentTypes()** - Method in class `javax.faces.application.Application`
  Return an `Iterator` over the set of currently defined component types for this `Application`.

**getConcept()** - Method in interface
javax.xml.registry.infomodel.**Classification**

*Gets the Concept that is classifying the object.*

**getConnection()** - Method in interface
javax.resource.cci.**ConnectionFactory**

*Gets a connection to an EIS instance.*

**getConnection(ConnectionSpec)** - Method in interface
javax.resource.cci.**ConnectionFactory**

*Gets a connection to an EIS instance.*

**getConnection()** - Method in interface javax.resource.cci.**Interaction**

*Gets the Connection associated with the Interaction.*

**getConnection(Subject, ConnectionRequestInfo)** - Method in interface
javax.resource.spi.**ManagedConnection**

*Creates a new connection handle for the underlying physical connection represented by the ManagedConnection instance.*

**getConnectionFactory()** - Method in interface
javax.management.j2ee.statistics.**JCAConnectionStats**

*Returns the associated JCAConnectionFactory OBJECT_NAME*

**getConnectionHandle()** - Method in class
javax.resource.spi.**ConnectionEvent**

*Get the connection handle associated with the Managed Connection instance.*

**getConnectionPools()** - Method in interface
javax.management.j2ee.statistics.**JCAStats**

*Returns an array of JCAConnectionPool Stats that provide statistics about the connection pools associated with the referencing JCA resource statistics*

**getConnectionPools()** - Method in interface
javax.management.j2ee.statistics.**JDBCStats**

**getConnections()** - Method in interface
javax.management.j2ee.statistics.**JCAStats**

*Returns an array of JCAConnectionStats that provide statistics about the non-pooled connections associated with the referencing JCA resource statistics*

**getConnections()** - Method in interface
javax.management.j2ee.statistics.**JDBCStats**

**getConnections()** - Method in interface
javax.management.j2ee.statistics.**JMSStats**
Returns an array of JMSConnectionStats that provide statistics about the connections associated with the referencing JMS resource.

**getConsumers()** - Method in interface `javax.management.j2ee.statistics.JMSSessionStats`

Returns an array of JMSConsumerStats that provide statistics about the message consumers associated with the referencing JMS session statistics.

**getContainerClientId(FacesContext)** - Method in class `javax.faces.component.UIComponent`

Allow components that implement `NamingContainer` to selectively disable prepending their clientId to their descendent's clientIds by breaking the prepending logic into a separately callable method.

**getContainerClientId(FacesContext)** - Method in class `javax.faces.component.UIForm`

Override the `UIComponent.getContainerClientId(javax.faces.context.FacesContext)` to allow users to disable this form from prepending its clientId to its descendent's clientIds depending on the value of this form's `UIForm.isPrependId()` property.

**getContent(DataSource)** - Method in interface `javax.activation.DataContentHandler`

Return an object representing the data in its most preferred form.

**getContent()** - Method in class `javax.activation.DataHandler`

Return the data in its preferred Object form.

**getContent()** - Method in class `javax.mail.internet.MimeBodyPart`

Return the content as a Java object.

**getContent()** - Method in class `javax.mail.internet.MimeMessage`

Return the content as a Java object.

**getContent()** - Method in interface `javax.mail.Part`

Return the content as a Java object.

**getContent()** - Method in class `javax.xml.soap.AttachmentPart`

Gets the content of this `AttachmentPart` object as a Java object.

**getContent()** - Method in class `javax.xml.soap.SOAPPart`

Returns the content of the SOAPEnvelope as a JAXP `Source` object.

**getContentDescription()** - Method in class `javax.xml.soap.SOAPMessage`

Retrieves a description of this `SOAPMessage` object's content.

**getContentID()** - Method in class `javax.mail.internet.MimeBodyPart`
Returns the value of the "Content-ID" header field.

**getContentID()** - Method in class `javax.mail.internet.MimeMessage`
Returns the value of the "Content-ID" header field.

**getContentID()** - Method in interface `javax.mail.internet.MimePart`
Get the Content-ID of this part.

**getContentId()** - Method in class `javax.xml.soap.AttachmentPart`
Gets the value of the MIME header whose name is "Content-ID".

**getContentId()** - Method in class `javax.xml.soap.SOAPPart`
Retrieves the value of the MIME header whose name is "Content-Id".

**getContentLanguage()** - Method in class `javax.mail.internet.MimeBodyPart`
Get the languages specified in the Content-Language header of this MimePart.

**getContentLanguage()** - Method in class `javax.mail.internet.MimeMessage`
Get the languages specified in the "Content-Language" header field of this message.

**getContentLanguage()** - Method in interface `javax.mail.internet.MimePart`
Get the language tags specified in the Content-Language header of this MimePart.

**getContentLength()** - Method in interface `javax.servlet.ServletRequest`
Returns the length, in bytes, of the request body and made available by the input stream, or -1 if the length is not known.

**getContentLength()** - Method in class `javax.servlet.ServletRequestWrapper`
The default behavior of this method is to return getContentLength() on the wrapped request object.

**getContentLocation()** - Method in class `javax.xml.soap.AttachmentPart`
Gets the value of the MIME header whose name is "Content-Location".

**getContentLocation()** - Method in class `javax.xml.soap.SOAPPart`
Retrieves the value of the MIME header whose name is "Content-Location".

**getContentMD5()** - Method in class `javax.mail.internet.MimeBodyPart`
Return the value of the "Content-MD5" header field.

**getContentMD5()** - Method in class `javax.mail.internet.MimeMessage`
Return the value of the "Content-MD5" header field.

**getContentMD5()** - Method in interface `javax.mail.internet.MimePart`
Get the Content-MD5 digest of this part.

**getContentType()** - Method in class javax.mail.internet.MimeBodyPart
Produce the raw bytes of the content.

**getContentType()** - Method in class javax.mail.internet.MimeMessage
Produce the raw bytes of the content.

**getContentType()** - Method in class javax.activation.DataHandler
Return the MIME type of this object as retrieved from the source object.

**getContentType()** - Method in interface javax.activation.DataSource
This method returns the MIME type of the data in the form of a string.

**getContentType()** - Method in class javax.activation.FileDataSource
This method returns the MIME type of the data in the form of a string.

**getContentType(File)** - Method in class javax.activation.FileTypeMap
Return the type of the file object.

**getContentType(String)** - Method in class javax.activation.FileTypeMap
Return the type of the file passed in.

**getContentType(File)** - Method in class javax.activation.MimetypesFileTypeMap
Return the MIME type of the file object.

**getContentType(String)** - Method in class javax.activation.MimetypesFileTypeMap
Return the MIME type based on the specified file name.

**getContentType()** - Method in class javax.activation.URLDataSource
Returns the value of the URL content-type header field.

**getContentType()** - Method in class javax.faces.context.ResponseWriter
Return the content type (such as "text/html") for this ResponseWriter.

**getContentType()** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call
*ResponseWriter.getContentType()* on the wrapped *ResponseWriter* object.

**getContentType()** - Method in class javax.mail.internet.MimeBodyPart
Returns the value of the RFC 822 "Content-Type" header field.

**getContentType()** - Method in class javax.mail.internet.MimeMessage
Returns the value of the RFC 822 "Content-Type" header field.

**getContentType()** - Method in class javax.mail.internet.MimePartDataSource
Returns the content-type of this DataSource.

**getContentType()** - Method in class **javax.mail.Multipart**
Return the content-type of this Multipart.

**getContentType()** - Method in interface **javax.mail.Part**
Returns the Content-Type of the content of this part.

**getContentType()** - Method in class **javax.mail.util.BytesArrayDataSource**
Get the MIME content type of the data.

**getContentType()** - Method in interface **javax.servlet.ServletRequest**
Returns the MIME type of the body of the request, or null if the type is not known.

**getContentType()** - Method in class **javax.servlet.ServletRequestWrapper**
The default behavior of this method is to return getContentType() on the wrapped request object.

**getContentType()** - Method in interface **javax.servlet.ServletResponse**
Returns the content type used for the MIME body sent in this response.

**getContentType()** - Method in class **javax.servlet.ServletResponseWrapper**
The default behavior of this method is to return getContentType() on the wrapped response object.

**getContentType()** - Method in class **javax.xml.soap.AttachmentPart**
Gets the value of the MIME header whose name is "Content-Type".

**getContext(Class)** - Method in class **javax.el.ELContext**
Returns the context object associated with the given key.

**getContext()** - Method in class **javax.faces.context.ExternalContext**
Return the application environment object instance for the current application.

**getContext(String)** - Static method in class **javax.security.jacc.PolicyContext**
This method may be used by a Policy provider to activate the PolicyContextHandler registered to the context object key and cause it to return the corresponding policy context object from the container.

**getContext(String, Object)** - Method in interface **javax.security.jacc.PolicyContextHandler**
This public method is used by the PolicyContext class to activate the handler and obtain from it the context object identified by the (case-sensitive) key.
**getContext(String)** - Method in interface javax.servlet.**ServletContext**
Returns a **ServletContext** object that corresponds to a specified URL on the server.

**getContext()** - Method in interface javax.xml.ws.**Response**
Gets the contained response context.

**getContextData()** - Method in interface javax.interceptor.**InvocationContext**
Returns the context data associated with this invocation or lifecycle callback.

**getContextID()** - Method in interface javax.security.jacc.**PolicyConfiguration**
This method returns this object's policy context identifier.

**getContextID()** - Static method in class javax.security.jacc.**PolicyContext**
This static method returns the value of the policy context identifier associated with the thread on which the accessor is called.

**getContextPath()** - Method in interface javax.servlet.http.**HttpServletRequest**
Returns the portion of the request URI that indicates the context of the request.

**getContextPath()** - Method in class javax.servlet.http.**HttpServletRequestWrapper**
The default behavior of this method is to return getContextPath() on the wrapped request object.

**getContextPath()** - Method in interface javax.servlet.**ServletContext**
Returns the context path of the web application.

**getConvertedValue(FacesContext, Object)** - Method in class javax.faces.component.**UIInput**
Convert the submitted value into a "local value" of the appropriate data type, if necessary.

**getConvertedValue(FacesContext, UIComponent, Object)** - Method in class javax.faces.render.**Renderer**
Attempt to convert previously stored state information into an object of the type required for this component (optionally using the registered **Converter** for this component, if there is one).

**getConverter()** - Method in class javax.faces.component.**UIOutput**

**getConverter()** - Method in interface javax.faces.component.**ValueHolder**
Return the **Converter** (if any) that is registered for this **UIComponent**

**getConverterIds()** - Method in class javax.faces.application.**Application**
Return an iterator over the set of currently registered converter ids for this application.

`getConverterMessage()` - Method in class `javax.faces.component.UIInput`
If there has been a call to `UIInput.setConverterMessage(java.lang.String)` on this instance, return the message.

`getConverterTypes()` - Method in class `javax.faces.application.Application`
Return an iterator over the set of class instances for which Converter classes have been explicitly registered.

Returns an array containing all of the Cookie objects the client sent with this request.

`getCookies()` - Method in class `javax.servlet.http.HttpServletRequestWrapper`
The default behavior of this method is to return getCookies() on the wrapped request object.

`getCoords()` - Method in class `javax.faces.component.html.HtmlCommandLink`
Return the value of the `coords` property.

`getCoords()` - Method in class `javax.faces.component.html.HtmlOutputLink`
Return the value of the `coords` property.

`getCount()` - Method in class `javax.mail.internet.MimeMultipart`
Return the number of enclosed BodyPart objects.

`getCount()` - Method in class `javax.mail.Multipart`
Return the number of enclosed BodyPart objects.

`getCount()` - Method in interface `javax.mail.MultipartDataSource`
Return the number of enclosed BodyPart objects.

`getCount()` - Method in interface `javax.management.j2ee.statistics.CountStatistic`
The count since the last reset.

`getCount()` - Method in interface `javax.management.j2ee.statistics.TimeStatistic`
Number of times the operation was invoked since the beginning of this measurement.

`getCountry()` - Method in interface `javax.xml.registry.infomodel.PostalAddress`
Returns the country.

**getCountryCode()** - Method in interface javax.xml.registry.infomodel.**TelephoneNumber**  
Gets the country code.

**getCreateCount()** - Method in interface javax.management.j2ee.statistics.**EJBStats**  
Number of times create was called.

**getCreateCount()** - Method in interface javax.management.j2ee.statistics.**JCAConnectionPoolStats**  
The number of connections created

**getCreateCount()** - Method in interface javax.management.j2ee.statistics.**JDBCConnectionPoolStats**  
The number of connections created.

**getCreated()** - Method in class javax.faces.webapp.**UIComponentClassicTagBase**  
Return true if we dynamically created a new component instance during execution of this tag.

**getCreated()** - Method in class javax.faces.webapp.**UIComponentTagBase**  
Return true if we dynamically created a new component instance during execution of this tag.

**getCreatedComponents()** - Method in class javax.faces.webapp.**UIComponentClassicTagBase**  
Returns the List of **UIComponent** ids created or located by nested **UIComponentTag**s while processing the current request.

**getCreationTime()** - Method in interface javax.servlet.http.**HttpSession**  
Returns the time when this session was created, measured in milliseconds since midnight January 1, 1970 GMT.

**getCredentialData()** - Method in interface javax.resource.spi.security.**GenericCredential**  
*Deprecated.* Gets security data for a specific security mechanism represented by the GenericCredential.

**getCredentials()** - Method in interface javax.xml.registry.**Connection**  
Gets the credentials associated with this client.

**getCurrencyCode()** - Method in class javax.faces.convert.**NumberConverter**  
Return the ISO 4217 currency code used by getAsString() with a type of currency.

**getCurrencySymbol()** - Method in class
javax.faces.convert.NumberConverter

Return the currency symbol used by getAsString() with a type of currency.

**getCurrent()** - Method in interface
javax.management.j2ee.statistics.RangeStatistic

The current value of this attribute.

**getCurrentInstance()** - Static method in class
javax.faces.context.FacesContext

Return the FacesContext instance for the request that is being processed by the current thread, if any.

**getCurrentLocale()** - Method in interface
javax.enterprise.deploy.spi.DeploymentManager

Returns the active locale this implementation of javax.enterprise.deploy.spi subpackages is running.

**getData()** - Method in interface javax.xml.stream.events.Characters

Get the character data of this event.

**getData()** - Method in interface
javax.xml.stream.events.ProcessingInstruction

The data section of the processing instruction.

**getDataHandler()** - Method in class javax.mail.internet.MimeBodyPart

Return a DataHandler for this body part's content.

**getDataHandler()** - Method in class javax.mail.internet.MimeMessage

Return a DataHandler for this Message's content.

**getDataHandler()** - Method in interface javax.mail.Part

Return a DataHandler for the content within this part.

**getDataHandler()** - Method in class javax.xml.soap.AttachmentPart

Gets the DataHandler object for this AttachmentPart object.

**getDataModel()** - Method in class javax.faces.component.UIData

Return the internal DataModel object representing the data objects that we will iterate over in this component's rendering.

**getDataModel()** - Method in class javax.faces.model.DataModelEvent

Return the DataModel that fired this event.

**getDataModelListeners()** - Method in class
javax.faces.model.DataModel

Return the set of DataModelListeners interested in notifications from this DataModel.

**getSource()** - Method in class javax.activation.DataHandler

Return the DataSource associated with this instance of DataHandler.

**getDate()** - Method in class javax.mail.search.DateTerm
Return the Date to compare with.

**getDateHeader(String)** - Method in interface `javax.servlet.http.HttpServletRequest`  
Returns the value of the specified request header as a `long` value that represents a `Date` object.

**getDateHeader(String)** - Method in class `javax.servlet.http.HttpServletRequestWrapper`  
The default behavior of this method is to return `getDateHeader(String name)` on the wrapped request object.

**getDConfigBean(DDBean)** - Method in interface `javax.enterprise.deploy.spi.DConfigBean`  
Return the JavaBean containing the server-specific deployment configuration information based upon the XML data provided by the DBean.

**getDConfigBean(DDBeanRoot)** - Method in interface `javax.enterprise.deploy.spi.DConfigBeanRoot`  
Return a DConfigBean for a deployment descriptor that is not the module's primary deployment descriptor.

**getDConfigBeanRoot(DDBeanRoot)** - Method in interface `javax.enterprise.deploy.spi.DeploymentConfiguration`  
Returns the top level configuration bean, DConfigBeanRoot, associated with the deployment descriptor represented by the designated DBeanRoot bean.

**getDConfigBeanVersion()** - Method in interface `javax.enterprise.deploy.spi.DeploymentManager`  
Returns the J2EE platform version number for which the configuration beans are provided.

**getDConfigBeanVersionType(int)** - Static method in class `javax.enterprise.deploy.shared.DConfigBeanVersionType`  
Return an object of the specified value.

**getDDBean()** - Method in interface `javax.enterprise.deploy.spi.DConfigBean`  
Return the JavaBean containing the deployment descriptor XML text associated with this DConfigBean.

**getDDBeanRoot()** - Method in interface `javax.enterprise.deploy.model.DeployableObject`  
Return the top level standard bean representing the root of the
getDDBeanRoot(String) - Method in interface javax.enterprise.deploy.model.DeployableObject
  Returns a DDBeanRoot object for the XML instance document named.

getDDBeanRootVersion() - Method in interface javax.enterprise.deploy.model.DDBeanRoot
  Returns the version number of an XML instance document.

getDebug() - Method in class javax.mail.Session
  Get the debug setting for this Session.

getDebugOut() - Method in class javax.mail.Session
  Returns the stream to be used for debugging output.

getDeclaration() - Method in interface javax.xml.stream.events.EntityReference
  Return the declaration of this entity.

getDeclarativeQueryManager() - Method in interface javax.xml.registry.RegistryService
  Returns the DeclarativeQueryManager object implemented by the JAXR provider.

getDeclare() - Method in class javax.servlet.jsp.tagext.TagVariableInfo
  The body of the <declare> element.

getDeclare() - Method in class javax.servlet.jsp.tagext.VariableInfo
  Returns whether this is a new variable.

getDeclaredType() - Method in class javax.xml.bind.JAXBElement
  Returns the Java binding of the xml element declaration's type attribute.

getDefaultCommandMap() - Static method in class javax.activation.CommandMap
  Get the default CommandMap.

defaultDomain() - Method in interface javax.management.j2ee.Management
  Returns the default domain name of this MEJB.

defaultFactory() - Static method in class javax.servlet.jsp.JspFactory
  Returns the default factory for this implementation.

getDefaultFileTypeMap() - Static method in class javax.activation.FileTypeMap
  Return the default FileTypeMap for the system.

defaultFolder() - Method in class javax.mail.Store
  Returns a Folder object that represents the 'root' of the default
namespace presented to the user by the Store.

getDefaultInstance(Properties, Authenticator) - Static method in class javax.mail.Session
  Get the default Session object.

getDefaultJavaCharsetName - Static method in class javax.mail.Session
  Get the default Session object.

getDefaultJavaCharsetName - Static method in class javax.mail.internet.MimeUtility
  Get the default charset corresponding to the system's current default locale.

getDefaultLocale() - Method in interface javax.enterprise.deploy.spi.DeploymentManager
  Returns the default locale supported by this implementation of javax.enterprise.deploy.spi subpackages.

getDefaultLocale() - Method in class javax.faces.application.Application
  Return the default Locale for this application.

getDefaultPostalScheme() - Method in interface javax.xml.registry.RegistryService
  Gets the default user-defined postal scheme for codifying the attributes of PostalAddress.

getDefaultRenderKitId() - Method in class javax.faces.application.Application
  Return the renderKitId to be used for rendering this application.

getDefaultTypeMapping() - Method in interface javax.xml.rpc.encoding.TypeMappingRegistry
  Gets the registered default TypeMapping instance.

getDefaultUserName() - Method in class javax.mail.Authenticator

getDelegate() - Method in interface javax.persistence.EntityManager
  Return the underlying provider object for the EntityManager, if available.

getDeletedMessageCount() - Method in class javax.mail.Folder
  Get the number of deleted messages in this Folder.

getDeliveryMode() - Method in interface javax.jms.MessageProducer
  Gets the producer's default delivery mode.

getDeployableObject() - Method in interface javax.enterprise.deploy.model.DDBeanRoot
  Return the containing DeployableObject
**getDeployableObject(String)** - Method in interface
javax.enterprise.deploy.model.J2eeApplicationObject
   Return the DeployableObject of the specified URI designator.

**getDeployableObject()** - Method in interface
javax.enterprise.deploy.spi.DeploymentConfiguration
   Returns an object that provides access to the deployment descriptor
data and classes of a J2EE module.

**getDeployableObjects(ModuleType)** - Method in interface
javax.enterprise.deploy.model.J2eeApplicationObject
   Return the all DeployableObjects of the specified type.

**getDeployableObjects()** - Method in interface
javax.enterprise.deploy.model.J2eeApplicationObject
   Return the all DeployableObjects in this application.

**getDeploymentFactories()** - Method in class
javax.enterprise.deploy.sharedfactories.DeploymentFactoryManager
   Retrieve the lists of currently registered DeploymentFactories.

**getDeploymentManager(String, String, String)** - Method in class
javax.enterprise.deploy.sharedfactories.DeploymentFactoryManager
   Retrieves a DeploymentManager instance to use for deployment.

**getDeploymentManager(String, String, String)** - Method in interface
javax.enterprise.deploy.spi.factories.DeploymentFactory
   Return a connected DeploymentManager instance.

**getDeploymentStatus()** - Method in class
javax.enterprise.deploy.spi.status.ProgressEvent
   Retrieve the status information.

**getDeploymentStatus()** - Method in interface
javax.enterprise.deploy.spi.status.ProgressObject
   Retrieve the status of this activity.

**getDescendantConcepts()** - Method in interface
javax.xml.registry.infomodel.ClassificationScheme
   Gets all descendant Concepts.

**getDescendantConcepts()** - Method in interface
javax.xml.registry.infomodel.Concept
   Gets all descendant Concepts.

**getDescendantOrganizations()** - Method in interface
javax.xml.registry.infomodel.Organization
   Gets all descendant Organizations.

**getDescription()** - Method in interface
javax.enterprise.deploy.spi.Target
   Retrieve other descriptive information about the target.
**getDescription()** - Method in class `javax.faces.model.SelectItem`
   Return a description of this item, for use in development tools.

**getDescription()** - Method in class `javax.mail.internet.MimeBodyPart`
   Returns the "Content-Description" header field of this body part.

**getDescription()** - Method in class `javax.mail.internet.MimeMessage`
   Returns the "Content-Description" header field of this Message.

**getDescription()** - Method in interface `javax.mail.Part`
   Return a description String for this part.

**getDescription()** - Method in interface
   `javax.management.j2ee.statistics.Statistic`
   A human-readable description of the Statistic.

**getDescription()** - Method in class
   `javax.servlet.jsp.tagext.TagAttributeInfo`
   Gets the description string of this tag attribute.

**getDescription()** - Method in interface
   `javax.xml.registry.infomodel.RegistryObject`
   Gets the textual description for this object.

**getDeserializer(Class, QName)** - Method in interface
   `javax.xml.rpc.encoding.TypeMapping`
   Gets the DeserializerFactory registered for the specified pair of Java type and XML data type.

**getDeserializerAs(String)** - Method in interface
   `javax.xml.rpc.encoding.DeserializerFactory`
   Returns a Deserializer for the specified XML processing mechanism type.

**getDestination()** - Method in interface `javax.jms.MessageProducer`
   Gets the destination associated with this MessageProducer.

**getDestination()** - Method in interface
   `javax.management.j2ee.statistics.JMSProducerStats`
   Returns a string that encapsulates the identity of the message destination.

**getDetail()** - Method in class `javax.faces.application.FacesMessage`
   Return the localized detail text.

**getDetail()** - Method in exception
   `javax.xml.rpc.soap.SOAPFaultException`
   Gets the detail element.

**getDetail()** - Method in interface `javax.xml.soap.SOAPFault`
   Returns the optional detail element for this SOAPFault object.

**getDetailEntries()** - Method in interface `javax.xml.soap.Detail`
Gets an Iterator over all of the DetailEntry in this Detail object.

**getDir()** - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlForm
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlInputSecret
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlMessage
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlMessages
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlOutputFormat
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlOutputText
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the dir property.

**getDir()** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
   Return the value of the dir property.
getDir() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
   Return the value of the dir property.
getDir() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
   Return the value of the dir property.
getDir() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
   Return the value of the dir property.
getDir() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
   Return the value of the dir property.
getDir() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
   Return the value of the dir property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
   Return the value of the disabledClass property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
   Return the value of the disabledClass property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
   Return the value of the disabledClass property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
   Return the value of the disabledClass property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
   Return the value of the disabledClass property.
getDisabledClass() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
   Return the value of the disabledClass property.
getDisableMessageID() - Method in interface
javax.jms.MessageProducer
   Gets an indication of whether message IDs are disabled.
getDisableMessageTimestamp() - Method in interface
javax.jms.MessageProducer
  Gets an indication of whether message timestamps are disabled.

getDisconnectedDeploymentManager(String) - Method in class
javax.enterprise.deploy.shared.factories.DeploymentFactoryManager
  Return a disconnected DeploymentManager instance.

getDisconnectedDeploymentManager(String) - Method in interface
javax.enterprise.deploy.spi.factories.DeploymentFactory
  Return a disconnected DeploymentManager instance.

getDisplayName() - Method in interface
javax.enterprise.deploy.spi.factories.DeploymentFactory
  Provide a string with the name of this vendor's DeploymentManager.

getDisplayName() - Method in class javax.servlet.jsp.tagext.TagInfo
  Get the displayName.

getAddression() - Method in class javax.mail.internet.ContentDisposition
  Return the disposition value.

getAddression() - Method in class javax.mail.internet.MimeBodyPart
  Returns the value of the "Content-Disposition" header field.

getAddression() - Method in class javax.mail.internet.MimeMessage
  Returns the value of the "Content-Disposition" header field.

getAddression() - Method in interface javax.mail.Part
  Return the disposition of this part.

getAddression() - Method in class
javax.faces.webapp.UIComponentClassicTagBase
  Return the flag value that should be returned from the doAfterBody() method when it is called.

getAddression() - Method in interface
javax.xml.stream.events.DTD
  Returns the entire Document Type Declaration as a string, including the internal DTD subset.

getAddression() - Method in class
javax.faces.webapp.UIComponentClassicTagBase
  Return the flag value that should be returned from the doEnd() method when it is called.

getAddression() - Method in class
javax.servlet.http.Cookie
  Returns the domain name set for this cookie.

getAddression() - Method in class
javax.faces.webapp.UIComponentClassicTagBase
  Return the flag value that should be returned from the doStart() method when it is called.
**getDouble(String)** - Method in interface *javax.jms.MapMessage*

Returns the `double` value with the specified name.

**getDoubleProperty(String)** - Method in interface *javax.jms.Message*

Returns the value of the `double` property with the specified name.

**getDTDType()** - Method in interface *javax.xml.stream.events.Attribute*

Gets the type of this attribute, default is the String "CDATA"

**getDurableSubscriptionCount()** - Method in interface *javax.management.j2ee.statistics.JMSSessionStats*

Number of durable subscriptions.

**getEISProductName()** - Method in interface *javax.resource.cci.ConnectionMetaData*

Returns product name of the underlying EIS instance connected through the Connection that produced this metadata.

**getEISProductName()** - Method in interface *javax.resource.spi.ManagedConnectionMetaData*

Returns Product name of the underlying EIS instance connected through the ManagedConnection.

**getEISProductVersion()** - Method in interface *javax.resource.cci.ConnectionMetaData*

Returns product version of the underlying EIS instance.

**getEISProductVersion()** - Method in interface *javax.resource.spi.ManagedConnectionMetaData*

Returns product version of the underlying EIS instance connected through the ManagedConnection.

**getEJBHome()** - Method in interface *javax.ejb.EJBContext*

Obtain the enterprise bean's remote home interface.

**getEJBHome()** - Method in interface *javax.ejb.EJBMetaData*

Obtain the remote home interface of the enterprise Bean.

**getEJBHome()** - Method in interface *javax.ejb.EJBObject*

Obtain the enterprise Bean's remote home interface.

**getEJBHome()** - Method in interface *javax.ejb.HomeHandle*

Obtain the home object represented by this handle.

**getEJBLocalHome()** - Method in interface *javax.ejb.EJBContext*

Obtain the enterprise bean's local home interface.

**getEJBLocalHome()** - Method in interface *javax.ejb.EJBLocalObject*

Obtain the enterprise Bean's local home interface.

**getEJBLocalObject()** - Method in interface *javax.ejb.EntityContext*

Obtain a reference to the EJB local object that is currently associated with the instance.
**getEJBLocalObject()** - Method in interface javax.ejb.SessionContext
  Obtain a reference to the EJB local object that is associated with the instance.

**getEJBMetaData()** - Method in interface javax.ejb.EJBHome
  Obtain the EJBMetaData interface for the enterprise Bean.

**getEJBObject()** - Method in interface javax.ejb.EntityContext
  Obtain a reference to the EJB object that is currently associated with the instance.

**getEJBObject()** - Method in interface javax.ejb.Handle
  Obtain the EJB object reference represented by this handle.

**getEJBObject()** - Method in interface javax.ejb.SessionContext
  Obtain a reference to the EJB object that is currently associated with the instance.

**getELContext()** - Method in class javax.el.ELContextEvent
  Returns the ELContext that was created.

**getELContext()** - Method in class javax.faces.context.FacesContext
  Return the ELContext instance for this FacesContext instance.

**getELContext()** - Method in class javax.faces.webapp.UIComponentELTag
  Return the ELContext for the FacesContext for this request.

**getELContext()** - Method in class javax.faces.webapp.UIComponentTagBase
  Return the ELContext for the FacesContext for this request.

**getELContext()** - Method in class javax.servlet.jsp.JspContext
  Returns the ELContext associated with this JspContext.

**getELContextListeners()** - Method in class javax.faces.application.Application
  If no calls have been made to
    Application.addELContextListener(javax.el.ELContextListener),
  this method must return an empty array.

**getElement(ResultT)** - Method in interface javax.xml.bind.annotation.DomHandler
  Once the portion is sent to the Result.

**getElement(DOMResult)** - Method in class javax.xml.bind.annotation.W3CDomHandler

**getElementName(Object)** - Method in class javax.xml.bind.JAXBIntrospector
  Get xml element qname for jaxbElement.
**getElementName()** - Method in interface javax.xml.soap.SOAPElement

Returns the name of this SOAPElement object.

**getElementQName()** - Method in interface javax.xml.soap.SOAPElement

Returns the qname of this SOAPElement object.

**getElementText()** - Method in class javax.xml.stream.util.EventReaderDelegate

**getElementText()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**getElementText()** - Method in interface javax.xml.stream.XMLEventReader

Reads the content of a text-only element.

**getElementText()** - Method in interface javax.xml.stream.XMLStreamReader

Reads the content of a text-only element, an exception is thrown if this is not a text-only element.

**getELResolver()** - Method in class javax.el.ELContext

Retrieves the ELResolver associated with this context.

**getELResolver()** - Method in class javax.faces.application.Application

Return the singleton ELResolver instance to be used for all EL resolution.

**getEmailAddresses()** - Method in interface javax.xml.registry.infomodel.User

Gets the email addresses for this User.

**getEnabledClass()** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox

Return the value of the enabledClass property.

**getEnabledClass()** - Method in class javax.faces.component.html.HtmlSelectManyListbox

Return the value of the enabledClass property.

**getEnabledClass()** - Method in class javax.faces.component.html.HtmlSelectManyMenu

Return the value of the enabledClass property.

**getEnabledClass()** - Method in class javax.faces.component.html.HtmlSelectOneListbox

Return the value of the enabledClass property.
javax.faces.component.html.HtmlSelectOneMenu

Return the value of the enabledClass property.

**getEnabledClass()** - Method in class

javax.faces.component.html.HtmlSelectOneRadio

Return the value of the enabledClass property.

**getEnclosingWriter()** - Method in class

javax.servlet.jsp.tagext.BodyContent

Get the enclosing JspWriter.

**getEncoding()** - Method in class java.mail.internet.MimeBodyPart

Returns the content transfer encoding from the "Content-Transfer-Encoding" header field.

**getEncoding()** - Method in class java.mail.internet.MimeMessage

Returns the content transfer encoding from the "Content-Transfer-Encoding" header field.

**getEncoding()** - Method in interface java.mail.internet.MimePart

Get the transfer encoding of this part.

**getEncoding(DataSource)** - Static method in class

java.mail.internet.MimeUtility

Get the content-transfer-encoding that should be applied to the input stream of this datasource, to make it mailsafe.

**getEncoding(DataHandler)** - Static method in class

java.mail.internet.MimeUtility

Same as getEncoding(DataSource) except that instead of reading the data from an InputStream it uses the writeTo method to examine the data.

**getEncoding()** - Method in class

java.mail.internet.PreencodedMimeBodyPart

Returns the content transfer encoding specified when this object was created.

**getEncoding()** - Method in class

java.xml.bind.helpers.AbstractMarshallerImpl

Convenience method for getting the current output encoding.

**getEncoding()** - Method in class

java.xml.stream.util.StreamReaderDelegate

Return input encoding if known or null if unknown.

**getEncodingStyle()** - Method in interface java.xml.soap.SOAPElement
Returns the encoding style for this SOAPElement object.

**getEnctype()** - Method in class `javax.faces.component.html.HtmlForm`
Returns the value of the enctype property.

**getEngineInfo()** - Method in class `javax.servlet.jsp.JspFactory`
called to get implementation-specific information on the current JSP engine.

**getEntities()** - Method in interface `javax.xml.stream.events/DTD`
Return a List containing the general entities, both external and internal, declared in the DTD.

**getEntity()** - Method in exception `javax.persistence.OptimisticLockException`
Returns the entity that caused this exception.

**getEntry(String)** - Method in interface `javax.enterprise.deploy.model.DeployableObject`
Returns the InputStream for the given entry name The file name must be relative to the root of the module.

**getEnumeration()** - Method in interface `javax.jms.QueueBrowser`
Gets an enumeration for browsing the current queue messages in the order they would be received.

**getEnumValueTable()** - Method in class `javax.enterprise.deploy.shared.ActionType`
Returns the enumeration value table for class ActionType

**getEnumValueTable()** - Method in class `javax.enterprise.deploy.shared.CommandType`
Returns the enumeration value table for class CommandType

**getEnumValueTable()** - Method in class `javax.enterprise.deploy.shared.DConfigBeanVersionType`
Returns the enumeration value table for class DConfigBeanVersionType

**getEnumValueTable()** - Method in class `javax.enterprise.deploy.shared.ModuleType`
Returns the enumeration value table for class ModuleType

**getEnumValueTable()** - Method in class `javax.enterprise.deploy.shared.StateType`
Returns the enumeration value table for class StateType

**getEnvelope()** - Method in class `javax.xml.soap.SOAPPart`
Gets the SOAPEnvelope object associated with this SOAPPart object.

**getEnvironment()** - Method in interface `javax.ejb.EJBContext`
Deprecated. Use the JNDI naming context java:comp/env to access
enterprise bean's environment.

**getErrorClass()** - Method in class
javax.faces.component.html.HtmlMessage

Return the value of the errorClass property.

**getErrorClass()** - Method in class
javax.faces.component.html.HtmlMessages

Return the value of the errorClass property.

**getErrorCode()** - Method in exception javax.jms.JMSException

Gets the vendor-specific error code.

**getErrorCode()** - Method in exception
javax.resource.ResourceException

Get the error code.

**getErrorCode()** - Method in exception javax.xml.bind.JAXBException

Get the vendor specific error code

**getErrorCode()** - Method in exception
javax.xml.bind.TypeConstraintException

Get the vendor specific error code

**getErrorData()** - Method in class javax.servlet.jsp.PageContext

Provides convenient access to error information.

**getErrorObjectKey()** - Method in exception
javax.xml.registry.RegistryException

Gets the Key to the first object that encountered an error in the registry.

**getErrorCode()** - Method in class
javax.faces.component.html.HtmlMessage

Return the value of the errorStyle property.

**getErrorCode()** - Method in class
javax.faces.component.html.HtmlMessages

Return the value of the errorStyle property.

**getEventAllocator()** - Method in class
javax.xml.stream.XMLInputFactory

Gets the allocator used by streams created with this factory

**getEventHandler()** - Method in class javax.xml.bind.Binder

Return the current event handler or the default event handler if one hasn't been set.

**getEventHandler()** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**getEventHandler()** - Method in class
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**
Return the current event handler or the default event handler if one hasn't been set.

**getEventHandler()** - Method in interface javax.xml.bind.**Marshaller**
Return the current event handler or the default event handler if one hasn't been set.

**getEventHandler()** - Method in interface javax.xml.bind.**Unmarshaller**
Return the current event handler or the default event handler if one hasn't been set.

**getEventHandler()** - Method in interface javax.xml.bind.**Validator**
**Deprecated. since JAXB2.0**

**getEvents()** - Method in class javax.xml.bind.util.**ValidationEventCollector**
Return an array of ValidationEvent objects containing a copy of each of the collected errors and warnings.

**getEventType()** - Method in interface javax.xml.registry.infomodel.**AuditableEvent**
Gets the type of this event.

**getEventType()** - Method in interface javax.xml.stream.events.**XMLEvent**
Returns an integer code for this event.

**getEventType()** - Method in class javax.xml.stream.util.**StreamReaderDelegate**
Returns an integer code that indicates the type of the event the cursor is pointing to.

**getException()** - Method in class javax.resource.spi.**ConnectionEvent**
Get the exception.

**getException()** - Method in class javax.resource.spi.work.**WorkEvent**
Return the WorkException.

**getException()** - Method in class javax.servlet.jsp.**PageContext**
The current value of the exception object (an Exception).

**getException()** - Method in error javax.xml.stream.**FactoryConfigurationError**
Return the nested exception (if any)

**getExceptionListener()** - Method in interface javax.jms.**Connection**
Gets the ExceptionListener object for this connection.

**getExceptions()** - Method in interface javax.xml.registry.**BulkResponse**
Get the Collection of RegistryException instances in case of partial commit.

**getExecutor()** - Method in class javax.xml.ws.**Endpoint**

Returns the executor for this **Endpoint** instance.

**getExecutor()** - Method in class javax.xml.ws.**Service**

Returns the executor for this **Service** instance.

**getExecutor()** - Method in class javax.xml.ws.spi.**ServiceDelegate**

Returns the executor for this **Service** instance.

**getExpectedType()** - Method in class javax.el.**ValueExpression**

Returns the type of the expression will be coerced to after evaluation.

**getExpectedTypeName()** - Method in class javax.servlet.jsp.tagext.**TagAttributeInfo**

Returns the name of the expected type (as a String) of this deferred value attribute.

**getExpiration()** - Method in interface javax.xml.registry.infomodel.**RegistryEntry**

Gets expirationDate attribute of the RegistryEntry within the Registry.

**getExpiredMessageCount()** - Method in interface javax.management.j2ee.statistics.**JMSEndpointStats**

Number of messages that expired before delivery.

**getExpiredMessageCount()** - Method in interface javax.management.j2ee.statistics.**JMSSessionStats**

Number of expired messages.

**getExpressionEvaluator()** - Method in class javax.servlet.jsp.**JspContext**

**Deprecated.** As of JSP 2.1, replaced by

*JspApplicationContext.getExpressionFactory()*

**getExpressionFactory()** - Method in class javax.faces.application.**Application**

Returns the **ExpressionFactory** instance for this application.

**getExpressionFactory()** - Method in interface javax.servlet.jsp.**JspApplicationContext**

Returns a factory used to create **ValueExpression**s and **MethodExpression**s so that EL expressions can be parsed and evaluated.

**getExpressionString()** - Method in class javax.el.**Expression**

Returns the original String used to create this **Expression**, unmodified.
**getExpressionString()** - Method in class **javax.faces.el.MethodBinding**

Deprecated. Return the (possibly **null**) expression String, with leading and trailing delimiters, from which this **MethodBinding** was built.

**getExpressionString()** - Method in class **javax.faces.el.ValueBinding**

Deprecated. Return the (possibly **null**) expression String, including the delimiters, from which this **ValueBinding** was built.

**getExtension()** - Method in interface **javax.xml.registry.infomodel.TelephoneNumber**

Gets the internal extension.

**getExternalContext()** - Method in class **javax.faces.context.FacesContext**

Return the **ExternalContext** instance for this **FacesContext** instance.

**getExternalIdentifiers()** - Method in interface **javax.xml.registry.infomodel.RegistryObject**

Returns the ExternalIdentifiers associated with this object that are external identifiers for this object.

**getExternalLinks()** - Method in interface **javax.xml.registry.infomodel.RegistryObject**

Returns the ExternalLinks associated with this object.

**getExternalURI()** - Method in interface **javax.xml.registry.infomodel.ExternalLink**

Gets URI to the an external resource.

**getFacesContext()** - Method in class **javax.faces.component.UIComponent**

Convenience method to return the **FacesContext** instance for the current request.

**getFacesContext()** - Method in class **javax.faces.component.UIComponentBase**

**getFacesContext(Object, Object, Object, Lifecycle)** - Method in class **javax.faces.context.FacesContextFactory**

Create (if needed) and return a **FacesContext** instance that is initialized for the processing of the specified request and response objects, utilizing the specified **Lifecycle** instance, for this web application.

**getFacesContext()** - Method in class **javax.faces.event.PhaseEvent**

Return the **FacesContext** for the request being processed.

**getFacesContext()** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**

**getFacesContext()** - Method in class
javax.faces.webapp.**UIComponentTagBase**

Return the **FacesContext** instance for the current request.

**getFacesJspId()** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**

If this method has been called before on this tag's useful lifetime (before **UIComponentClassicTagBase.release()** was called), return the previously returned value.

**getFacesListeners(Class)** - Method in class
javax.faces.component.**UIComponent**

Return an array of registered **FacesListener**s that are instances of the specified class.

**getFacesListeners(Class)** - Method in class
javax.faces.component.**UIComponentBase**

**getFacesMessage()** - Method in exception
javax.faces.convert.**ConverterException**

Returns the FacesMessage associated with this exception; this will only be available if the converter that threw this exception instance placed it there.

**getFacesMessage()** - Method in exception
javax.faces.validator.**ValidatorException**

Returns the FacesMessage associated with the exception.

**getFacet(String)** - Method in class
javax.faces.component.**UIComponent**

Convenience method to return the named facet, if it exists, or **null** otherwise.

**getFacet(String)** - Method in class
javax.faces.component.**UIComponentBase**

**getFacetCount()** - Method in class
javax.faces.component.**UIComponent**

Return the number of facet **UIComponent**s that are associated with this **UIComponent**.

**getFacetCount()** - Method in class
javax.faces.component.**UIComponentBase**

**getFacetName()** - Method in class
javax.faces.webapp.**UIComponentClassicTagBase**
Return the facet name that we should be stored under, if any; otherwise, return null (indicating that we will be a child component).

**getFacets()** - Method in class `javax.faces.component.UIComponent`  
Return a mutable `Map` representing the facet `UIComponent`s associated with this `UIComponent`, keyed by facet name (which must be a String).

**getFacets()** - Method in class `javax.faces.component.UIComponentBase`

**getFacetsAndChildren()** - Method in class `javax.faces.component.UIComponent`  
Return a mutable `Map` representing the facet `UIComponent`s associated with this `UIComponent`, keyed by facet name (which must be a String).

**getFacetsAndChildren()** - Method in class `javax.faces.component.UIComponentBase`

**getFactory(String)** - Static method in class `javax.faces.FactoryFinder`  
Create (if necessary) and return a per-web-application instance of the appropriate implementation class for the specified JavaServer Faces factory class, based on the discovery algorithm described in the class description.

**getFamily()** - Method in class `javax.faces.component.UIColumn`

**getFamily()** - Method in class `javax.faces.component.UIColumn`

**getFamily()** - Method in class `javax.faces.component.UIInput`

**getFamily()** - Method in class `javax.faces.component.UIMessage`

**getFamily()** - Method in class `javax.faces.component.UIMessages`

**getFamily()** - Method in class `javax.faces.component.UIData`

**getFamily()** - Method in class `javax.faces.component.UIForm`

**getFamily()** - Method in class `javax.faces.component.UIInput`

**getFamily()** - Method in class `javax.faces.component.UIInput`

**getFamily()** - Method in class `javax.faces.component.UIMessage`

**getFamily()** - Method in class `javax.faces.component.UIMessages`

**getFamily()** - Method in class
javax.faces.component.UiNamingContainer

getFamily() - Method in class javax.faces.component.UIOutput

getFamily() - Method in class javax.faces.component.UIPanel

getFamily() - Method in class javax.faces.component.UIParameter

getFamily() - Method in class javax.faces.component.UISelectBoolean

getFamily() - Method in class javax.faces.component.UISelectItem

getFamily() - Method in class javax.faces.component.UISelectItems

getFamily() - Method in class javax.faces.component.UISelectMany

getFamily() - Method in class javax.faces.component.UISelectOne

getFamily() - Method in class javax.faces.component.UIViewRoot

getFatalClass() - Method in class
javax.faces.component.html.HtmlMessage
    Return the value of the fatalClass property.

getFatalClass() - Method in class
javax.faces.component.html.HtmlMessages
    Return the value of the fatalClass property.

getFatalStyle() - Method in class
javax.faces.component.html.HtmlMessage
    Return the value of the fatalStyle property.

getFatalStyle() - Method in class
javax.faces.component.html.HtmlMessages
    Return the value of the fatalStyle property.

gFault() - Method in interface javax.xml.soap.SOAPBody
    Returns the SOAPFault object in this SOAPBody object.

gFault() - Method in exception javax.xml.ws.soap.SOAPFaultException
    Gets the embedded SOAPFault instance.

gFaultActor() - Method in exception
javax.xml.rpc.soap.SOAPFaultException
    Gets the faultactor element.
**getFaultActor()** - Method in interface `javax.xml.soap.SOAPFault`
  Gets the fault actor for this `SOAPFault` object.

**getFaultCode()** - Method in exception
`javax.xml.rpc.soap.SOAPFaultException`
  Gets the fault code element.

**getFaultCode()** - Method in interface `javax.xml.soap.SOAPFault`
  Gets the fault code for this `SOAPFault` object.

**getFaultCodeAsName()** - Method in interface
`javax.xml.soap.SOAPFault`
  Gets the mandatory SOAP 1.1 fault code for this `SOAPFault` object as a SAAJ `Name` object.

**getFaultCodeAsQName()** - Method in interface
`javax.xml.soap.SOAPFault`
  Gets the fault code for this `SOAPFault` object as a `QName` object.

**getFaultNode()** - Method in interface
`javax.xml.soap.SOAPFault`
  Returns the optional Node element value for this `SOAPFault` object.

**getFaultReasonLocales()** - Method in interface
`javax.xml.soap.SOAPFault`
  Returns an `Iterator` over a distinct sequence of `Locales` for which there are associated `Reason` `Text` items.

**getFaultReasonText(Locale)** - Method in interface
`javax.xml.soap.SOAPFault`
  Returns the `Reason` `Text` associated with the given `Locale`.

**getFaultReasonTexts()** - Method in interface `javax.xml.soap.SOAPFault`
  Returns an `Iterator` over a sequence of `String` objects containing all of the `Reason` `Text` items for this `SOAPFault`.

**getFaultRole()** - Method in interface `javax.xml.soap.SOAPFault`
  Returns the optional `Role` element value for this `SOAPFault` object.

**getFaultString()** - Method in exception
`javax.xml.rpc.soap.SOAPFaultException`
  Gets the `faultstring` element.

**getFaultString()** - Method in interface `javax.xml.soap.SOAPFault`
  Gets the fault string for this `SOAPFault` object.

**getFaultStringLocale()** - Method in interface `javax.xml.soap.SOAPFault`
  Gets the locale of the fault string for this `SOAPFault` object.

**getFaultSubcodes()** - Method in interface `javax.xml.soap.SOAPFault`
  Gets the Subcodes for this `SOAPFault` as an iterator over `QNames`.

**getFeatureDescriptors(ELContext, Object)** - Method in class
`javax.el.ArrayELResolver`
Always returns `null`, since there is no reason to iterate through set of all integers.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.BeanELResolver`
If the base object is not `null`, returns an `Iterator` containing the set of JavaBeans properties available on the given object.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.CompositeELResolver`
Returns information about the set of variables or properties that can be resolved for the given base object.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.ELResolver`
Returns information about the set of variables or properties that can be resolved for the given base object.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.ListELResolver`
Always returns `null`, since there is no reason to iterate through set of all integers.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.MapELResolver`
If the base object is a map, returns an `Iterator` containing the set of keys available in the `Map`.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.el.ResourceBundleELResolver`
If the base object is a `ResourceBundle`, returns an `Iterator` containing the set of keys available in the `ResourceBundle`.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.servlet.jsp.el.ImplicitObjectELResolver`
If the base object is `null`, and the property matches the name of a JSP implicit object, returns an `Iterator` containing `FeatureDescriptor` objects with information about each JSP implicit object resolved by this resolver.

`getFeatureDescriptors(ELContext, Object)` - Method in class `javax.servlet.jsp.el.ScopedAttributeELResolver`
If the base object is `null`, returns an `Iterator` containing `FeatureDescriptor` objects with information about each scoped attribute resolved by this resolver.

`getFile()` - Method in class `javax.activation.FileDataSource`
Return the File object that corresponds to this FileDataSource.
**getFilename()** - Method in interface java.xs.engage.deployment.model.DDBeanRoot
Returns the filename relative to the root of the module of the XML instance document this DDBeanRoot represents.

**getFileName()** - Method in class javax.mail.internet.MimeBodyPart
Get the filename associated with this body part.

**getFileName()** - Method in class javax.mail.internet.MimeMessage
Get the filename associated with this Message.

**getFileName()** - Method in interface javax.mail.Part
Get the filename associated with this part, if possible.

**getFilterName()** - Method in interface javax.servlet.FilterConfig
Returns the filter-name of this filter as defined in the deployment descriptor.

**getFirst()** - Method in class javax.faces.component.UIData
Return the zero-relative row number of the first row to be displayed.

**getFirstName()** - Method in interface javax.xml.registry.infomodel.PersonName
Gets the first name for this Person.

**getFloat(String)** - Method in interface javax.jms.MapMessage
Returns the float value with the specified name.

**getFloatProperty(String)** - Method in interface javax.jms.Message
Returns the value of the float property with the specified name.

**getFlushMode()** - Method in interface javax.persistence.EntityManager
Get the flush mode that applies to all objects contained in the persistence context.

**getFolder()** - Method in class javax.mail.event.FolderEvent
Return the affected folder.

**getFolder(String)** - Method in class javax.mail.Folder
Return the Folder object corresponding to the given name.

**getFolder()** - Method in exception javax.mail.FolderClosedException
Returns the dead Folder object
getFolder() - Method in exception javax.mail.FolderNotFoundException
Returns the offending Folder object.

getFolder() - Method in class javax.mail.Message
Get the folder from which this message was obtained.

getFolder() - Method in exception javax.mail.ReadOnlyFolderException
Returns the dead Folder object.

getFolder(URLName) - Method in class javax.mail.Session
Get a closed Folder object for the given URLName.

getFolder(String) - Method in class javax.mail.Store
Return the Folder object corresponding to the given name.

getFolder(URLName) - Method in class javax.mail.Store
Return a closed Folder object, corresponding to the given URLName.

getFooter() - Method in class javax.faces.component.UIColumn
Return the footer facet of the column (if any).

getFooter() - Method in class javax.faces.component.UIData
Return the footer facet of this component (if any).

getFooterClass() - Method in class javax.faces.component.html.UIColumn
Return the value of the footerClass property.

getFooterClass() - Method in class javax.faces.component.html.HtmlColumn
Return the value of the footerClass property.

getFooterClass() - Method in class javax.faces.component.html.HtmlDataTable
Return the value of the footerClass property.

getFooterClass() - Method in class javax.faces.component.html.HtmlPanelGrid
Return the value of the footerClass property.

getFor() - Method in class javax.faces.component.html.HtmlOutputLabel
Return the value of the for property.

getFor() - Method in class javax.faces.component.UIMessage
Return the client identifier of the component for which this component represents associated message(s) (if any).

getFormatId() - Method in interface javax.transaction.xa.Xid
Obtain the format identifier part of the XID.

getFrame() - Method in class javax.faces.component.html.HtmlDataTable
Return the value of the frame property.

getFrame() - Method in class javax.faces.component.html.HtmlPanelGrid
Return the value of the frame property.

getFreePoolSize() - Method in interface javax.management.j2ee.statistics.JCACollectionPoolStats
The number of free connections in the pool

**getFreePoolSize()** - Method in interface
javax.management.j2ee.statistics.[JDBCConnectionPoolStats](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)

Number of free connections in the pool.

**getFrom()** - Method in class javax.mail.internet.[MimeMessage](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Returns the value of the RFC 822 "From" header fields.

**getFrom()** - Method in class javax.mail.[Message](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Returns the "From" attribute.

**getFullName()** - Method in class javax.mail.[Folder](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Returns the full name of this Folder.

**getFullName()** - Method in interface
javax.xml.registry.infomodel.[PersonName](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Gets the fully formatted name for this person.

**getFunction(String)** - Method in class
javax.servlet.jsp.tagext.[TagLibraryInfo](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Get the FunctionInfo for a given function name, looking through all the functions in this tag library.

**getFunctionClass()** - Method in class
javax.servlet.jsp.tagext.[FunctionInfo](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
The class of the function.

**getFunctionMapper()** - Method in class javax.el.[ELContext](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Retrieves the FunctionMapper associated with this ELContext.

**getFunctions()** - Method in class javax.servlet.jsp.tagext.[TagLibraryInfo](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
An array describing the functions that are defined in this tag library.

**getFunctionSignature()** - Method in class
javax.servlet.jsp.tagext.[FunctionInfo](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
The signature of the function.

**getGlobalTransactionId()** - Method in interface javax.transaction.xa.[Xid](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Obtain the global transaction identifier part of XID as an array of bytes.

**getGroup(boolean)** - Method in class javax.mail.internet.[InternetAddress](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Return the members of a group address.

**getHandle()** - Method in interface javax.ejb.[EJBObject](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Obtain a handle for the EJB object.

**getHandle()** - Method in interface javax.ejb.[Timer](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Get a serializable handle to the timer.

**getHandlerChain(QName)** - Method in interface
javax.xml.rpc.handler.[HandlerRegistry](https://java.specs.java.net/jmmt-api/jmmt-api-1.0.1-SNAPSHOT/docs/javadoc/)
Gets the handler chain for the specified service endpoint.
**getHandlerChain()** - Method in interface javax.xml.ws.Binding
Gets a copy of the handler chain for a protocol binding instance.

**getHandlerChain(PortInfo)** - Method in interface javax.xml.ws.handler.HandlerResolver
Gets the handler chain for the specified port.

**getHandlerClass()** - Method in class javax.xml.rpc.handler.HandlerInfo
Gets the Handler class.

**getHandlerConfig()** - Method in class javax.xml.rpc.handler.HandlerInfo
Gets the Handler configuration.

**getHandlerKeys()** - Static method in class javax.security.jacc.PolicyContext
This method may be used to obtain the keys that identify the container specific context handlers registered by the container.

**getHandlerRegistry()** - Method in interface javax.xml.rpc.Service
Returns the configured HandlerRegistry instance for this Service instance.

**getHandlerResolver()** - Method in class javax.xml.ws.Service
Returns the configured handler resolver.

**getHandlerResolver()** - Method in class javax.xml.ws.spi.ServiceDelegate
Returns the configured handler resolver.

**getHeader()** - Method in class javax.faces.component.UIColumn
Return the header facet of the column (if any).

**getHeader()** - Method in class javax.faces.component.UIData
Return the header facet of this component (if any).

**getHeader(String)** - Method in class javax.mail.internet.InternetHeaders
Get all the headers for this header_name, returned as a single String, with headers separated by the delimiter.

**getHeader(String)** - Method in class javax.mail.internet.MimeBodyPart
Get all the headers for this header_name.

**getHeader(String, String)** - Method in class javax.mail.internet.MimeBodyPart
Get all the headers for this header_name, returned as a single String, with headers separated by the delimiter.

**getHeader(String)** - Method in class javax.mail.internet.MimeMessage
Get all the headers for this header_name.
getHeader(String, String) - Method in class javax.mail.internet.MimeMessage
    Get all the headers for this header name, returned as a single String, with headers separated by the delimiter.
getHeader(String, String) - Method in interface javax.mail.internet.MimePart
    Get the values of all header fields available for this header, returned as a single String, with the values separated by the delimiter.
getHeader(String) - Method in interface javax.mail.Part
    Get all the headers for this header name.
getHeader(String) - Method in interface javax.mail.internet.MimePart
    Get the values of all header fields available for this header, returned as a single String, with the values separated by the delimiter.
getHeader(String) - Method in class javax.servlet.http.HttpServletRequest
    Returns the value of the specified request header as a String.
getHeader(String) - Method in class javax.servlet.http.HttpServletRequestWrapper
    The default behavior of this method is to return getHeader(String name) on the wrapped request object.
getHeader(String) - Method in class javax.xml.soap.MimeHeaders
    Returns all of the values for the specified header as an array of String objects.
getHeader() - Method in interface javax.xml.soap.SOAPEnvelope
    Returns the SOAPHeader object for this SOAPEnvelope object.
getHeaderClass() - Method in class javax.faces.component.html.HtmlColumn
    Return the value of the headerClass property.
getHeaderClass() - Method in class javax.faces.component.html.HtmlDataTable
    Return the value of the headerClass property.
getHeaderClass() - Method in class javax.faces.component.html.HtmlPanelGrid
    Return the value of the headerClass property.
getHeaderName() - Method in class javax.mail.search.HeaderTerm
    Return the name of the header to compare with.
getHeaderNames() - Method in class javax.mail.FetchProfile
    Get the names of the header-fields set in this profile.
    Returns an enumeration of all the header names this request contains.
**getHeaderNames()** - Method in class
javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return getHeaderNames() on the wrapped request object.

**getHeaders(String)** - Method in interface
javax.servlet.http.HttpServletRequest
Returns all the values of the specified request header as an Enumeration of String objects.

**getHeaders(String)** - Method in class
javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return getHeaders(String name) on the wrapped request object.

**getHeaders()** - Method in class
javax.xml.rpc.handler.GenericHandler
Gets the header blocks processed by this Handler instance.

**getHeaders()** - Method in interface
javax.xml.rpc.handler.Handler
Gets the header blocks that can be processed by this Handler instance.

**getHeaders()** - Method in class
javax.xml.rpc.handler.HandlerInfo
Gets the header blocks processed by this Handler.

**getHeaders()** - Method in interface
javax.xml.ws.handler.soap.SOAPHandler
Gets the header blocks that can be processed by this Handler instance.

**getHeaders(QName, JAXBContext, boolean)** - Method in interface
javax.xml.ws.handler.soap.SOAPMessageContext
Gets headers that have a particular qualified name from the message in the message context.

**getHeapSize()** - Method in interface
javax.management.j2ee.statistics.JVMStats
Size of the JVM's heap.

**getHeight()** - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the height property.

**getHighWaterMark()** - Method in interface
javax.management.j2ee.statistics.RangeStatistic
The highest value this attribute has held since the beginning of the measurement.

**getHomeHandle()** - Method in interface javax.ejb.EJBHome
Obtain a handle for the remote home object.
**getHomeInterfaceClass()** - Method in interface `javax.ejb.EJBMetaData`
  Obtain the Class object for the enterprise Bean's remote home interface.

**getHost()** - Method in class `javax.mail.internet.NewsAddress`
  Get the host.

**getHost()** - Method in class `javax.mail.URLName`
  Returns the host of this URLName.

**getHreflang()** - Method in class `javax.faces.component.html.HtmlCommandLink`
  Return the value of the `hreflang` property.

**getHreflang()** - Method in class `javax.faces.component.html.HtmlOutputLink`
  Return the value of the `hreflang` property.

**getHttpSession()** - Method in interface `javax.xml.rpc.server.ServletEndpointContext`
  The `getHttpSession` method returns the current HTTP session (as a `javax.servlet.http.HttpSession`).

**getHumanPresentableName()** - Method in class `javax.activation.ActivationDataFlavor`
  Return the Human Presentable name.

**getId()** - Method in interface `javax.enterprise.deploy.model.DDBean`
  Returns a tool-specific reference for attribute ID on an element in the deployment descriptor.

**getId()** - Method in class `javax.faces.component.UICOMPONENT UICOMPONENT`
  Return the component identifier of this `UICOMPONENT`.

**getId()** - Method in class `javax.faces.component.UICOMPONENTBase`

**getId()** - Method in class `javax.faces.webapp.UICOMPONENT ClassicTagBase`
  Return the `id` value assigned by the page author.

**getId()** - Method in class `javax.resource.spi.ConnectionEvent`
  Get the type of event

**getId()** - Method in interface `javax.servlet.http.HttpServletRequest`
  Returns a string containing the unique identifier assigned to this session.

**getId()** - Method in class `javax.servlet.jsp.tagext.TagData`
  The value of the tag's `id` attribute.

**getId()** - Method in class `javax.servlet.jsp.tagext.TagSupport`
  The value of the `id` attribute of this tag; or null.
**getld()** - Method in class javax.servlet.jsp.tagext.**ValidationMessage**
Get the jsp:id.

**getld()** - Method in interface javax.xml.registry.infomodel.**Key**
Returns the unique Id of this key.

**getIdAttribute(TagAttributeInfo[])** - Static method in class javax.servlet.jsp.tagext.**TagAttributeInfo**
Convenience static method that goes through an array of TagAttributeInfo objects and looks for "id".

**getIdentificationScheme()** - Method in interface javax.xml.registry.infomodel.**ExternalIdentifier**
Gets the ClassificationScheme that is used as the identification scheme for identifying this object.

**getld()** - Method in interface javax.servlet.http.**HttpSessionContext**
**Deprecated. As of Java Servlet API 2.1 with no replacement. This method must return an empty Enumeration and will be removed in a future version of this API.**

**getIgnoreCase()** - Method in class javax.mail.search.**StringTerm**
Return true if we should ignore case when matching.

**getImage()** - Method in class javax.faces.component.html.**HtmlCommandButton**
Return the value of the image property.

**getImplementor()** - Method in class javax.xml.ws.**Endpoint**
Returns the implementation object for this endpoint.

**getIndexOfNextChildTag()** - Method in class javax.faces.webapp.**UIComponentClassicTagBase**
Return the index of the next child to be added as a child of this tag.

**getInfo()** - Method in interface javax.ejb.**Timer**
Get the information associated with the timer at the time of creation.

**getInfo()** - Method in class javax.faces.component.html.**HtmlMessage**
Return the value of the infoClass property.

**getInfo()** - Method in class javax.faces.component.html.**HtmlMessages**
Return the value of the infoClass property.

**getInfoString()** - Method in class javax.servlet.jsp.tagext.**TagInfo**
The information string for the tag.
**getInfoString()** - Method in class `javax.servlet.jsp.tagext.TagLibraryInfo`
Information (documentation) for this TLD.

**getInfoStyle()** - Method in class `javax.faces.component.html.HtmlMessage`
Return the value of the `infoStyle` property.

**getInfoStyle()** - Method in class `javax.faces.component.html.HtmlMessages`
Return the value of the `infoStyle` property.

**getInitParameter(String)** - Method in class `javax.faces.context.ExternalContext`
Return the value of the specified application initialization parameter (if any).

**getInitParameter(String)** - Method in interface `javax.servlet.FilterConfig`
Returns a `String` containing the value of the named initialization parameter, or `null` if the parameter does not exist.

**getInitParameter(String)** - Method in class `javax.servlet.GenericServlet`
Returns a `String` containing the value of the named initialization parameter, or `null` if the parameter does not exist.

**getInitParameter(String)** - Method in interface `javax.servlet.ServletConfig`
Returns a `String` containing the value of the named initialization parameter, or `null` if the parameter does not exist.

**getInitParameter(String)** - Method in interface `javax.servlet.ServletContext`
Returns a `String` containing the value of the named context-wide initialization parameter, or `null` if the parameter does not exist.

**getInitParameterMap()** - Method in class `javax.faces.context.ExternalContext`
Return an immutable `Map` whose keys are the set of application initialization parameter names configured for this application, and whose values are the corresponding parameter values.

**getInitParameterNames()** - Method in interface `javax.servlet.FilterConfig`
Returns the names of the filter's initialization parameters as an `Enumeration` of `String` objects, or an empty `Enumeration` if the filter has no initialization parameters.

**getInitParameterNames()** - Method in class `javax.servlet.GenericServlet`
Returns the names of the servlet's initialization parameters as an `Enumeration` of `String` objects, or an empty `Enumeration` if the servlet has no initialization parameters.
**getInitParameterNames()** - Method in interface `javax.servlet.ServletConfig`
Returns the names of the servlet's initialization parameters as an `Enumeration` of `String` objects, or an empty `Enumeration` if the servlet has no initialization parameters.

**getInitParameterNames()** - Method in interface `javax.servlet.ServletContext`
Returns the names of the context's initialization parameters as an `Enumeration` of `String` objects, or an empty `Enumeration` if the context has no initialization parameters.

**getInitParameters()** - Method in class `javax.servlet.jsp.tagext.TagLibraryValidator`
Get the init parameters data as an immutable Map.

**getInputStream()** - Method in class `javax.activation.DataHandler`
Get the InputStream for this object.

**getInputStream()** - Method in interface `javax.activation.DataSource`
This method returns an `InputStream` representing the data and throws the appropriate exception if it can not do so.

**getInputStream()** - Method in class `javax.activation.FileDataSource`
This method will return an `InputStream` representing the the data and will throw an `IOException` if it can not do so.

**getInputStream()** - Method in class `javax.activation.URLDataSource`
The `getInputStream` method from the URL.

**getInputStream()** - Method in class `javax.mail.internet.MimeBodyPart`
Return a decoded input stream for this body part's "content".

**getInputStream()** - Method in class `javax.mail.internet.MimeMessage`
Return a decoded input stream for this Message's "content".

**getInputStream()** - Method in class `javax.mail.internet.MimePartDataSource`
Returns an input stream from this MimePart.

**getInputStream()** - Method in interface `javax.mail.Part`
Return an input stream for this part's "content".

**getInputStream()** - Method in class `javax.mail.util.ByteArrayDataSource`
Return an `InputStream` for the data.

**getInputStream()** - Method in class `javax.servlet.jsp.tagext.PageData`
Returns an input stream on the XML view of a JSP page.

**getInputStream()** - Method in interface `javax.servlet.ServletRequest`
Retrieves the body of the request as binary data using a `ServletInputStream`. 
**getInputStream()** - Method in class `javax.servlet.ServletRequestWrapper`

The default behavior of this method is to return `getInputStream()` on the wrapped request object.

**getInstance()** - Static method in class `javax.enterprise.deploy.shared.factories.DeploymentFactoryManager`

Retrieve the Singleton `DeploymentFactoryManager` `getInstance(Properties, Authenticator)`.

**getInstance(Properties, Authenticator)** - Static method in class `javax.mail.Session`

Get a new `Session` object.

**getInstance(Properties)** - Static method in class `javax.mail.Session`

Get a new `Session` object.

**getInt(String)** - Method in interface `javax.jms.MapMessage`

Returns the int value with the specified name.

**getIntHeader(String)** - Method in interface `javax.servlet.http.HttpServletRequest`

Returns the value of the specified request header as an int.

**getIntHeader(String)** - Method in class `javax.servlet.http.HttpServletRequestWrapper`

The default behavior of this method is to return `getIntHeader(String name)` on the wrapped request object.

**getIntProperty(String)** - Method in interface `javax.jms.Message`

Returns the value of the int property with the specified name.

**getInvalidAddresses()** - Method in class `javax.mail.event.TransportEvent`

Return the addresses to which this message could not be sent.

**getInvalidAddresses()** - Method in exception `javax.mail.SendFailedException`

Return the addresses to which this message could not be sent.

**getInvalidConnections(Set)** - Method in interface `javax.resource.spi.ValidatingManagedConnectionFactory`

This method returns a set of invalid `ManagedConnection` objects chosen from a specified set of `ManagedConnection` objects.

**getInvalidPropertyDescriptor()** - Method in exception `javax.resource.spi.InvalidPropertyException`
Get the list of invalid properties.

**getInvokedBusinessInterface()** - Method in interface javax.ejb.SessionContext
- Obtain the business interface through which the current business method invocation was made.

**getItemDescription()** - Method in class javax.faces.component.UISelectItem
- Return the description for this selection item.

**getItemLabel()** - Method in class javax.faces.component.UISelectItem
- Return the localized label for this selection item.

**getItems()** - Method in class javax.mail.FetchProfile
- Get the items set in this profile.

**getItemValue()** - Method in class javax.faces.component.UISelectItem
- Return the server value for this selection item.

**getJarFileUrls()** - Method in interface javax.persistence.spi.PersistenceUnitInfo
- Returns a list of URLs for the jar files or exploded jar file directories that the persistence provider must examine for managed classes of the persistence unit.

**getJavaEncoding(String)** - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl
- Gets the corresponding Java encoding name from an IANA name.

**getJAXBNode(XmlNode)** - Method in class javax.xml.bind.Binder
- Gets the JAXB object associated with the given XML element.

**getJdbcDataSource()** - Method in interface javax.management.j2ee.statistics.JDBCConnectionStats
- Identifies the JDBC driver for the corresponding JDBCConnection.

**getJMSCorrelationID()** - Method in interface javax.jms.Message
- Gets the correlation ID for the message.

**getJMSCorrelationIDAsBytes()** - Method in interface javax.jms.Message
- Gets the correlation ID as an array of bytes for the message.

**getJMSDeliveryMode()** - Method in interface javax.jms.Message
- Gets the DeliveryMode value specified for this message.

**getJMSDestination()** - Method in interface javax.jms.Message
- Gets the Destination object for this message.

**getJMSExpiration()** - Method in interface javax.jms.Message
- Gets the message's expiration value.

**getJMSMajorVersion()** - Method in interface
javax.jms.ConnectionMetaData
  Gets the JMS major version number.
**getJMSMessageID()** - Method in interface javax.jms.Message
  Gets the message ID.
**getJMSMinorVersion()** - Method in interface javax.jms.ConnectionMetaData
  Gets the JMS minor version number.
**getJMSPriority()** - Method in interface javax.jms.Message
  Gets the message priority level.
**getJMSProviderName()** - Method in interface javax.jms.ConnectionMetaData
  Gets the JMS provider name.
**getJMSRedelivered()** - Method in interface javax.jms.Message
  Gets an indication of whether this message is being redelivered.
**getJMSReplyTo()** - Method in interface javax.jms.Message
  Gets the Destination object to which a reply to this message should be sent.
**getJMSTimestamp()** - Method in interface javax.jms.Message
  Gets the message timestamp.
**getJMSType()** - Method in interface javax.jms.Message
  Gets the message type identifier supplied by the client when the message was sent.
**getJMSVersion()** - Method in interface javax.jms.ConnectionMetaData
  Gets the JMS API version.
**getJMSXPropertyNames()** - Method in interface javax.jms.ConnectionMetaData
  Gets an enumeration of the JMSX property names.
**getJspApplicationContext(ServletContext)** - Method in class javax.servlet.jsp.JspFactory
  Obtains the JspApplicationContext instance associated with the web application for the given ServletContext.
**getJspBody()** - Method in class javax.servlet.jsp.tagext.SimpleTagSupport
  Returns the body passed in by the container via setJspBody.
**getJspContext()** - Method in class javax.servlet.jsp.tagext.JspFragment
  Returns the JspContext that is bound to this JspFragment.
**getJspContext()** - Method in class javax.servlet.jsp.tagext.SimpleTagSupport
  Returns the page context passed in by the container via
setJspContext.

**getJspId()** - Method in class

`javax.faces.webapp.UIComponentClassicTagBase`

**getJtaDataSource()** - Method in interface

`javax.persistence.spi.PersistenceUnitInfo`

Returns the JTA-enabled data source to be used by the persistence provider.

**getJtaDataSource()** - Method in interface

`javax.xml.registry.infomodel.RegistryObject`

Gets the key representing the universally unique ID (UUID) for this object.

**getKey()** - Method in interface

`javax.security.jacc.PolicyContextHandler`

This public method returns the keys identifying the context objects supported by the handler.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlCommandButton`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlInputSecret`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlInputText`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlInputTextarea`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlSelectBooleanCheckbox`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlSelectManyCheckbox`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlSelectManyListbox`

Return the value of the label property.

**getLabel()** - Method in class

`javax.faces.component.html.HtmlSelectManyMenu`

Return the value of the label property.

**getLabel()** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
  Return the value of the label property.
  **getLabel()** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
  Return the value of the label property.
  **getLabel()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
  Return the value of the label property.
  **getLabel()** - Method in class
javax.faces.model.SelectItem
  Return the label of this item, to be rendered visibly for the user.
  **getLang()** - Method in class
javax.faces.component.html.HtmlCommandButton
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlCommandLink
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlDataTable
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlForm
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlGraphicImage
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlInputSecret
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlInputText
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlInputTextarea
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlMessage
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlMessages
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlOutputFormat
  Return the value of the lang property.
  **getLang()** - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlOutputLink`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlOutputText`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlPanelGrid`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
Return the value of the `lang` property.

**getLang()** - Method in class `javax.faces.component.html.HtmlSelectOneRadio`
Return the value of the `lang` property.

**getLargeIcon()** - Method in class `javax.servlet.jsp.tagext.TagInfo`
Get the path to the large icon.

Returns the last time the client sent a request associated with this session, as the number of milliseconds since midnight January 1, 1970 GMT, and marked by the time the container received the request.

**getLastModified(HttpServletRequest)** - Method in class `javax.servlet.http.HttpServlet`
Returns the time the `HttpServletRequest` object was last modified, in
milliseconds since midnight January 1, 1970 GMT.

**getLastName()** - Method in interface
javax.xml.registry.infomodel.PersonName
  Gets the last name (surname) for this Person.

**getLastSampleTime()** - Method in interface
javax.management.j2ee.statistics.Statistic
  The time of the last measurement represented as a long, whose value is the number of milliseconds since January 1, 1970, 00:00:00.

**getLayout()** - Method in class
javax.faces.component.html.HtmlMessages
  Return the value of the layout property.

**getLayout()** - Method in class
javax.faces.component.html.HtmlPanelGroup
  Return the value of the layout property.

**getLayout()** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
  Return the value of the layout property.

**getLayout()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
  Return the value of the layout property.

**getLifecycle(String)** - Method in class
javax.faces.lifecycle.LifecycleFactory
  Create (if needed) and return a Lifecycle instance for the specified lifecycle identifier.

**getLifecycleIds()** - Method in class
javax.faces.lifecycle.LifecycleFactory
  Return an Iterator over the set of lifecycle identifiers supported by this factory.

**getLifeCycleManager()** - Method in interface
javax.xml.registry.infomodel.RegistryObject
  Returns the LifeCycleManager that created this object.

**getLineCount()** - Method in class
javax.mail.internet.MimeBodyPart
  Return the number of lines for the content of this Part.

**getLineCount()** - Method in class
javax.mail.internet.MimeMessage
  Return the number of lines for the content of this message.

**getLineCount()** - Method in interface
javax.mail.Part
  Return the number of lines in the content of this part.

**getLineNumber()** - Method in class
javax.xml.bind.helpers.ValidationEventLocatorImpl
**getLineNumber()** - Method in interface javax.xml.bind.**ValidationEventLocator**
   Return the line number if available

**getLineNumber()** - Method in interface javax.xml.stream.**Location**
   Return the line number where the current event ends, returns -1 if none is available.

**getLinkedCause()** - Method in exception javax.xml.rpc.**JAXRPCException**
   Gets the Linked cause

**getLinkedCause()** - Method in exception javax.xml.rpc.**ServiceException**
   Gets the Linked cause

**getLinkedException()** - Method in exception javax.jms.**JMSException**
   Gets the exception linked to this one.

**getLinkedException()** - Method in exception javax.resource.**ResourceException**
   Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to `getCause` and `initCause` methods of the java.lang.Throwable class..

**getLinkedException()** - Method in exception javax.xml.bind.**JAXBException**

**getLinkedException()** - Method in interface javax.xml.bind.**ValidationEvent**
   Retrieve the linked exception for this warning/error.

**getLinkedObjects()** - Method in interface javax.xml.registry.infomodel.**ExternalLink**
   Gets the collection of RegistryObjects that are annotated by this ExternalLink.

**getLinkedWarning()** - Method in exception javax.resource.cci.**ResourceWarning**
   Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any,
that caused it to get thrown. Refer to `getCause` and `initCause` methods of the `java.lang.Throwable` class.

**getListener()** - Method in class
javax.xml.bind.helpers.Abs
t getListener() - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

getListener() - Method in class
getListener() - Method in interface javax.xml.bind.Marshaller
Return `Marshaler.Listener` registered with this `Marshaler`.
getListener() - Method in interface javax.xml.bind.Unmarshaler
Return `Unmarshaler.Listener` registered with this `Unmarshaler`.

**getListenerRegistry()** - Method in interface
javax.management.j2ee.Management
Returns the listener registry implementation for this MEJB.

**getLocalAddr()** - Method in interface javax.servlet.ServletRequest
Returns the Internet Protocol (IP) address of the interface on which the request was received.

**getLocalAddr()** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return `getLocalAddr()` on the wrapped request object.

**getLocalAddress(Session)** - Static method in class
javax.mail.internet.InternetAddress
Return an `InternetAddress` object representing the current user.

**getLocale()** - Method in class javax.el.ELContext
Get the `Locale` stored by a previous invocation to `ELContext.setLocale(java.util.Locale)`.

**getLocale()** - Method in class javax.faces.component.UIViewRoot
Return the `Locale` to be used in localizing the response being created for this view.

**getLocale()** - Method in class javax.faces.convert.DateTimeConverter
Return the `Locale` to be used when parsing or formatting dates and times.

**getLocale()** - Method in class javax.faces.convert.NumberConverter
Return the `Locale` to be used when parsing numbers.

**getLocale()** - Method in interface javax.servlet.ServletRequest
Returns the preferred `Locale` that the client will accept content in, based on the Accept-Language header.

**getLocale()** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return `getLocale()` on the wrapped request object.

**getLocale()** - Method in interface `javax.servlet.ServletResponse`  
Returns the locale specified for this response using the `ServletResponse.setLocale(java.util.Locale)` method.

**getLocale()** - Method in class `javax.servlet.ServletResponseWrapper`  
The default behavior of this method is to return `getLocale()` on the wrapped response object.

**getLocale()** - Method in interface `javax.xml.registry.infomodelLocalizedString`  
Get the Locale for this object.

**getLocales()** - Method in interface `javax.servlet.ServletRequest`  
Returns an `Enumeration` of `Locale` objects indicating, in decreasing order starting with the preferred locale, the locales that are acceptable to the client based on the Accept-Language header.

**getLocales()** - Method in class `javax.servlet.ServletRequestWrapper`  
The default behavior of this method is to return `getLocales()` on the wrapped request object.

**getLocaleString(Locale, String)** - Method in interface `javax.xml.registry.infomodelInternationalString`  
Gets the `LocalizedString` for the specified Locale and charsetName.

**getLocalizedStrings()** - Method in interface `javax.xml.registry.infomodelInternationalString`  
Returns the `LocalizedStrings` associated with this object.

**getLocalName()** - Method in interface `javax.servlet.ServletRequest`  
Returns the host name of the Internet Protocol (IP) interface on which the request was received.

**getLocalName()** - Method in class `javax.servlet.ServletRequestWrapper`  
The default behavior of this method is to return `getLocalName()` on the wrapped request object.

**getLocalName()** - Method in interface `javax.xml.soap.Name`  
Gets the local name part of the XML name that this `Name` object represents.

**getLocalName()** - Method in class  
`javax.xml.stream.util.StreamReaderDelegate`  
_Returns the (local) name of the current event.
**getLocalPort()** - Method in interface javax.servlet.ServletRequest
Returns the Internet Protocol (IP) port number of the interface on which the request was received.

**getLocalPort()** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return getLocalPort() on the wrapped request object.

**getLocalTransaction()** - Method in interface javax.resource.cci.Connection
Returns a LocalTransaction instance that enables a component to demarcate resource manager local transactions on the Connection.

**getLocalTransaction()** - Method in interface javax.resource.spi.ManagedConnection
Returns a javax.resource.spi.LocalTransaction instance.

**getLocalValue()** - Method in class javax.faces.component.UIOutput
**getLocalValue()** - Method in interface javax.faces.component.ValueHolder
Return the local value of this UIComponent (if any), without evaluating any associated ValueExpression.

**getLocation()** - Method in interface javax.xml.stream.events.XMLEvent
Return the location of this event.

**getLocation()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**getLocation()** - Method in exception javax.xml.stream.XMLStreamException
Gets the location of the exception

**getLocation()** - Method in interface javax.xml.stream.XMLStreamReader
Return the current location of the processor.

**getLocator()** - Method in class javax.xml.bind.helpers.ValidationEventImpl
Retrieve the locator for this warning/error.

**getLogWriter()** - Method in interface javax.resource.spi.ManagedConnection
Gets the log writer for this ManagedConnection instance.

**getLogWriter()** - Method in interface javax.resource.spi.ManagedConnectionFactory
Get the log writer for this ManagedConnectionFactory instance.

**getLog(String)** - Method in interface javax.jms.MapMessage
- Returns the `long` value with the specified name.

**getLongdesc()** - Method in class javax.faces.component.html.HtmlGraphicImage
- Return the value of the `longdesc` property.

**getLongProperty(String)** - Method in interface javax.jms.Message
- Returns the value of the `long` property with the specified name.

**getLowerBound()** - Method in interface javax.management.j2ee.statistics.BoundaryStatistic
- The lower limit of the value of this attribute. The upper limit of the value of this attribute.

**getLowWaterMark()** - Method in interface javax.management.j2ee.statistics.RangeStatistic
- The lowest value this attribute has held since the beginning of the measurement.

**getMajorVersion()** - Method in interface javax.servlet.ServletContext
- Returns the major version of the Java Servlet API that this servlet container supports.

**getMajorVersion()** - Method in interface javax.xml.registry.infomodel.Versionable
- Gets the major revision number for this version of the Versionable object.

**getManagedClassNames()** - Method in interface javax.persistence.spi.PersistenceUnitInfo
- Returns the list of the names of the classes that the persistence provider must add it to its set of managed classes.

**getManagedConnectionFactory()** - Method in interface javax.management.j2ee.statistics.JCAConnectionStats
- Returns the associated JCAManagedConnectionFactory

**getManagedConnectionFactory()** - Method in class javax.resource.spi.security.PasswordCredential
- Gets the target ManagedConnectionFactory for which the user name and password has been set by the application server.

**getMapNames()** - Method in interface javax.jms.MapMessage
- Returns an Enumeration of all the names in the MapMessage object.

**getMappingFileNames()** - Method in interface javax.persistence.spi.PersistenceUnitInfo
Returns the list of mapping file names that the persistence provider
must load to determine the mappings for the entity classes.

**getMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.InternetHeaders
Return all matching header lines as an Enumeration of Strings.

**getMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.MimeBodyPart
Get matching header lines as an Enumeration of Strings.

**getMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.MimeMessage
Get matching header lines as an Enumeration of Strings.

**getMatchingHeaderLines(String[])** - Method in interface
javax.mail.internet.MimePart
Get matching header lines as an Enumeration of Strings.

**getMatchingHeaders(String[])** - Method in class
javax.mail.internet.InternetHeaders
Return all matching Header objects.

**getMatchingHeaders(String[])** - Method in class
javax.mail.internet.MimeBodyPart
Return matching headers from this Message as an Enumeration of
Header objects.

**getMatchingHeaders(String[])** - Method in class
javax.mail.internet.MimeMessage
Return matching headers from this Message as an Enumeration of
Header objects.

**getMatchingHeaders(String[])** - Method in interface
javax.mail.Part
Return matching headers from this part as an Enumeration of
Header objects.

**getMatchingHeaders(String[])** - Method in class
javax.xml.soap.MimeHeaders
Returns all the MimeHeader objects whose name matches a name in
the given array of names.

**getMatchingMimeHeaders(String[])** - Method in class
javax.xml.soap.AttachmentPart
Retrieves all MimeHeader objects that match a name in the given
array.

**getMatchingMimeHeaders(String[])** - Method in class
javax.xml.soap.SOAPPart
Retrieves all MimeHeader objects that match a name in the given
array.

**getMaxAge()** - Method in class `javax.servlet.http.Cookie`  
Retrieves the maximum age of the cookie, specified in seconds. By default, -1 indicating the cookie will persist until browser shutdown.

**getMaxConnections()** - Method in interface `javax.resource.spi.ManagedConnectionMetaData`  
Returns the maximum limit on number of active concurrent connections that an EIS instance can support across client processes.

**getMaxFractionDigits()** - Method in class `javax.faces.convert.NumberConverter`  
Returns the maximum number of digits `getAsString()` should render in the fraction portion of the result.

**getMaximum()** - Method in class `javax.faces.validator.DoubleRangeValidator`  
Returns the maximum value to be enforced by this `Validator` or `Double.MAX_VALUE` if it has not been set.

**getMaximum()** - Method in class `javax.faces.validator.LengthValidator`  
Returns the maximum length to be enforced by this `Validator`, or 0 if the maximum has not been set.

**getMaximum()** - Method in class `javax.faces.validator.LongRangeValidator`  
Returns the maximum value to be enforced by this `Validator`.

**getMaximumSeverity()** - Method in class `javax.faces.context.FacesContext`  
Returns the maximum severity level recorded on any `FacesMessage` that has been queued, whether or not they are associated with any specific `UIComponent`.

**getMaxInactiveInterval()** - Method in interface `javax.servlet.http.HttpSession`  
Returns the maximum time interval, in seconds, that the servlet container will keep this session open between client accesses.

**getMaxIntegerDigits()** - Method in class `javax.faces.convert.NumberConverter`  
Returns the maximum number of digits `getAsString()` should render in the integer portion of the result.

**getMaxLength()** - Method in class `javax.faces.component.html.HtmlInputSecret`  
Returns the value of the `maxLength` property.

**getMaxLength()** - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the maxLength property.

getMaxTime() - Method in interface
javax.management.j2ee.statistics.TimeStatistic
The maximum amount of time taken to complete one invocation of this operation since the beginning of this measurement.

getMBeanCount() - Method in interface
javax.management.j2ee.Management
Returns the number of managed objects registered in the MEJB.

getMBeanInfo(ObjectName) - Method in interface
javax.management.j2ee.Management
This method discovers the attributes and operations that a managed object exposes for management.

getMechanismType() - Method in interface
javax.xml.rpc.encoding.Deserializer
Gets the type of the XML processing mechanism and representation used by this Deserializer.

getMechanismType() - Method in interface
javax.xml.rpc.encoding.Serializer
Gets the type of the XML processing mechanism and representation used by this Serializer.

getMechType() - Method in interface
javax.resource.spi.security.GenericCredential
Deprecated. Returns the mechanism type for the GenericCredential instance.

getMessage() - Method in exception javax.ejb.EJBException
Returns the detail message, including the message from the nested exception if there is one.

getMessage() - Method in interface
javax.enterprise.deploy.spi.status.DeploymentStatus
Retrieve any additional information about the status of this event.

getMessage() - Method in exception
javax.faces.application.ViewExpiredException
Return the message for this exception prepended with the view identifier if the view identifier is not null, otherwise, return the message.

getMessage() - Method in class
javax.mail.event.MessageChangedEvent
Return the changed Message.
**getMessage()** - Method in class *javax.mail.event.StoreEvent*
Get the message from the Store.

**getMessage()** - Method in class *javax.mail.event.TransportEvent*
Get the Message object associated with this Transport Event.

**getMessage(int)** - Method in class *javax.mail.Folder*
Get the Message object corresponding to the given message number.

**getMessage()** - Method in class *javax.mail.MessageContext*
Return the Message that contains the content.

**getMessage()** - Method in exception *javax.resource.ResourceException*
Returns a detailed message string describing this exception.

**getMessage()** - Method in class *javax.servlet.jsp.tagext.ValidationMessage*
Get the localized validation message.

**getMessage()** - Method in class *javax.xml.bind.helpers.ValidationEventImpl*

**getMessage()** - Method in interface *javax.xml.bind.ValidationEvent*
Retrieve the text message for this warning/error.

**getMessage()** - Method in exception *javax.xml.registry.JAXRException*
Returns the detail message for this JAXRException object.

**getMessage()** - Method in interface *javax.xml.rpc.handler.soap.SOAPMessageContext*
Gets the SOAPMessage from this message context.

**getMessage()** - Method in interface *javax.xml.ws.handler.soap.SOAPMessageContext*
Gets the SOAPMessage from this message context.

**getMessage()** - Method in interface *javax.xml.ws.handler.soap.SOAPMessageContext*
Gets the SOAPMessage from this message context.

**getLocalizedMessage()** - Method in class *javax.faces.application.Application*
Return the fully qualified class name of the ResourceBundle to be used for JavaServer Faces messages for this application.
**getMessageByUID(long)** - Method in interface `javax.mail.UIDFolder`
Get the Message corresponding to the given UID.

**getMessageChangeType()** - Method in class `javax.mail.event.MessageChangedEvent`
Return the type of this event.

**getMessageContext()** - Method in interface `javax.ejb.SessionContext`
Obtain a reference to the JAX-RPC MessageContext.

**getMessageContext()** - Method in class `javax.mail.internet.MimePartDataSource`
Return the MessageContext for the current part.

**getMessageContext()** - Method in interface `javax.mail.MessageAware`
Return the message context.

**getMessageContext()** - Method in interface `javax.xml.rpc.server.ServletEndpointContext`
The method `getMessageContext` returns the MessageContext targeted for this endpoint instance.

Returns the MessageContext for the request being served at the time this method is called.

**getMessageCount()** - Method in class `javax.mail.Folder`
Get total number of messages in this Folder.

**getMessageCount()** - Method in interface `javax.management.j2ee.statistics.JMSEndpointStats`
Number of messages sent or received.

**getMessageCount()** - Method in interface `javax.management.j2ee.statistics.JMSSessionStats`
Number of messages exchanged.

**getMessageCount()** - Method in interface `javax.management.j2ee.statistics.MessageDrivenBeanStats`
Number of messages received.

**getMessageFactory()** - Method in interface `javax.xml.ws.soap.SOAPBinding`
Gets the SAAJ MessageFactory instance used by this SOAP binding.

**getMessageID()** - Method in class `javax.mail.internet.MimeMessage`
Returns the value of the "Message-ID" header field.

**getMessageListener()** - Method in interface `javax.jms.MessageConsumer`
Gets the message consumer's MessageListener.
**getMessagListener()** - Method in interface `javax.jms.Session`  
Returns the session's distinguished message listener (optional).

**getMessagNumber()** - Method in class `javax.mail.Message`  
Get the Message number for this Message.

**getMessagss()** - Method in class `javax.faces.context.FacesContext`  
Return an Iterator over the FacesMessage that have been queued, whether or not they are associated with any specific client identifier.

**getMessagss(String)** - Method in class `javax.faces.context.FacesContext`  
Return an Iterator over the FacesMessage that have been queued that are associated with the specified client identifier (if clientId is not null), or over the FacesMessage that have been queued that are not associated with any specific client identifier (if clientId is null).

**getMessagss()** - Method in class `javax.mail.event.MessageCountEvent`  
Return the array of messages added or removed.

**getMessagss(int, int)** - Method in class `javax.mail.Folder`  
Get the Message objects for message numbers ranging from start through end, both start and end inclusive.

**getMessagss(int[])** - Method in class `javax.mail.Folder`  
Get the Message objects for message numbers specified in the array.

**getMessagss()** - Method in class `javax.mail.Folder`  
Get all Message objects from this Folder.

**getMessagssByUID(long, long)** - Method in interface `javax.mail.UIDFolder`  
Get the Messages specified by the given range.

**getMessagssByUID(long[])** - Method in interface `javax.mail.UIDFolder`  
Get the Messages specified by the given array of UIDs.

**getMessageSelector()** - Method in interface `javax.jms.MessageConsumer`  
Gets this message consumer's message selector expression.

**getMessageSelector()** - Method in interface `javax.jms.QueueBrowser`  
Gets this queue browser's message selector expression.

**getMessageType()** - Method in class `javax.mail.event.StoreEvent`  
Return the type of this event.

**getMessageWaitTime()** - Method in interface `javax.management.j2ee.statistics.JMSEndpointStats`  
Time spent by a message before being delivered.
javax.management.j2ee.statistics.JMSSessionStats
Time spent by a message before being delivered.

**getMetaData()** - Method in interface javax.jms.Connection
Gets the metadata for this connection.

**getMetaData()** - Method in interface javax.resource.cci.Connection
Gets the information on the underlying EIS instance represented through an active connection.

**getMetaData()** - Method in interface javax.resource.cci.ConnectionFactory
Gets metadata for the Resource Adapter.

**getMetaData()** - Method in interface javax.resource.spi.ManagedConnection
Gets the metadata information for this connection's underlying EIS resource manager instance.

**getMetadata()** - Method in class javax.xml.ws.Endpoint
Returns a list of metadata documents for the service.

**getMethod()** - Method in interface javax.interceptor.InvocationContext
Returns the method of the bean class for which the interceptor was invoked.

Returns the name of the HTTP method with which this request was made, for example, GET, POST, or PUT.

**getMethod()** - Method in class javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return getMethod() on the wrapped request object.

**getMethodInfo(ELContext)** - Method in class javax.el.MethodExpression
Evaluates the expression relative to the provided context, and returns information about the actual referenced method.

**getMethodReadyCount()** - Method in interface javax.management.j2ee.statistics.SessionBeanStats
Number of beans in the method-ready state.

**getMethodSignature()** - Method in class javax.servlet.jsp.tagext.TagAttributeInfo
Returns the expected method signature of this deferred method attribute.

**getMiddleName()** - Method in interface javax.xml.registry.infomodel.PersonName
Gets the middle name for this Person.

**getMimeHeader(String)** - Method in class \(\text{javax.xml.soap.AttachmentPart}\)

- Gets all the values of the header identified by the given String.

**getMimeHeader(String)** - Method in class \(\text{javax.xml.soap.SOAPPart}\)

- Gets all the values of the MimeHeader object in this SOAPPart object that is identified by the given String.

**getMimeHeaders()** - Method in class \(\text{javax.xml.soap.SOAPMessage}\)

- Returns all the transport-specific MIME headers for this SOAPMessage object in a transport-independent fashion.

**getMimeType()** - Method in class \(\text{javax.activation.ActivationDataFlavor}\)

- Return the MIME type for this DataFlavor.

**getMimeType(String)** - Method in interface \(\text{javax.servlet.ServletContext}\)

- Returns the MIME type of the specified file, or \text{null} if the MIME type is not known.

**getMimeType()** - Method in interface \(\text{javax.xml.registry.infomodel.ExtrinsicObject}\)

- Gets the mime type associated with this object.

**getMimeType()** - Method in class \(\text{javax.activation.MailcapCommandMap}\)

- Get all the MIME types known to this command map.

**getMinFractionDigits()** - Method in class \(\text{javax.faces.convert.NumberConverter}\)

- Return the minimum number of digits \text{getAsString()} should render in the fraction portion of the result.

**getMinimum()** - Method in class \(\text{javax.faces.validator.DoubleRangeValidator}\)

- Return the minimum value to be enforced by this \text{validator}, or \text{Double.MIN_VALUE} if it has not been set.

**getMinimum()** - Method in class \(\text{javax.faces.validator.LengthValidator}\)

- Return the minimum length to be enforced by this \text{validator}, or \text{0} if the minimum has not been set.

**getMinimum()** - Method in class \(\text{javax.faces.validator.LongRangeValidator}\)

- Return the minimum value to be enforced by this \text{validator}.

**getMinIntegerDigits()** - Method in class \(\text{javax.faces.convert.NumberConverter}\)
Return the minimum number of digits `getAsString()` should render in the integer portion of the result.

`getMinorVersion()` - Method in interface `javax.servlet.ServletContext`

Returns the minor version of the Servlet API that this servlet container supports.

`getMinorVersion()` - Method in interface `javax.xml.registry.infomodel.Versionable`

 Gets the minor revision number for this version of the Versionable object.

`getMinTime()` - Method in interface `javax.management.j2ee.statistics.TimeStatistic`

 The minimum amount of time taken to complete one invocation of this operation since the beginning of this measurement.

`getMode()` - Method in class `javax.mail.Folder`

 Return the open mode of this folder.

`getModuleDTDVersion()` - Method in interface `javax.enterprise.deploy.model.DDBeanRoot`

  *Deprecated.* As of version 1.1 replaced by
`DDBeanRoot.getDDBeanRootVersion()`

`getModuleDTDVersion()` - Method in interface `javax.enterprise.deploy.model.DeployableObject`

  *Deprecated.* As of version 1.1 replaced by
`DDBeanRoot.getDDBeanRootVersion()`

`getModuleExtension()` - Method in class `javax.enterprise.deploy.shared.ModuleType`

 Return the file extension string for this enumeration.

`getModuleID()` - Method in interface `javax.enterprise.deploy.spi.TargetModuleID`

 Retrieve the id assigned to represent the deployed module.

`getModuleType(int)` - Static method in class `javax.enterprise.deploy.shared.ModuleType`

 Return an object of the specified value.

`getModuleUris(ModuleType)` - Method in interface `javax.enterprise.deploy.model.J2eeApplicationObject`

 Return the list of URIs of the designated module type.

`getModuleUris()` - Method in interface `javax.enterprise.deploy.model.J2eeApplicationObject`

 Return the list of URIs for all modules in the application.

`getMustUnderstand()` - Method in interface
javax.xml.soap.SOAPHeaderElement
Returns the boolean value of the mustUnderstand attribute for this SOAPHeaderElement.

**getName()** - Method in class javax.activation.DataHandler
Return the name of the data object.

**getName()** - Method in interface javax.activation.DataSource
Return the *name* of this object where the name of the object is dependant on the nature of the underlying objects.

**getName()** - Method in class javax.activation.FileDataSource
Return the *name* of this object.

**getName()** - Method in class javax.activation.URLDataSource
Calls the getFile method on the URL used to instantiate the object.

**getName()** - Method in class javax.el.MethodInfo
Returns the name of the method

**getName()** - Method in interface javax.enterprise.deploy.spi.Target
Retrieve the name of the target server.

**getName()** - Method in class javax.faces.component.UIParameter
Return the optional parameter name for this parameter.

**getName()** - Method in class javax.faces.webapp.FacetTag
Return the name to be assigned to this facet.

**getName()** - Method in class javax.mail.Folder
Returns the name of this Folder.

**getName()** - Method in class javax.mail.Header
Returns the name of this header.

**getName()** - Method in class javax.mail.internet.MimePartDataSource
DataSource method to return a name.

**getName()** - Method in class javax.mail.util.ByteArrayDataSource
Get the name of the data.

**getName()** - Method in interface javax.management.j2ee.statistics.Statistic
The name of this Statistic.

**getName()** - Method in interface javax.resource.spi.security.GenericCredential
**Deprecated.** Returns the name of the resource principal associated with a GenericCredential instance.

**getName()** - Method in class javax.servlet.http.Cookie
Returns the name of the cookie.

**getName()** - Method in class javax.servlet.http.HttpSessionBindingEvent
Returns the name with which the attribute is bound to or unbound
from the session.

**getName()** - Method in class `javax.servlet.jsp.tagext.FunctionInfo`
The name of the function.

**getName()** - Method in class `javax.servlet.jsp.tagext.TagAttributeInfo`
The name of this attribute.

**getName()** - Method in class `javax.servlet.jsp.tagext.TagFileInfo`
The unique action name of this tag.

**getName()** - Method in class `javax.servlet.ServletContextAttributeEvent`
Return the name of the attribute that changed on the ServletContext.

**getName()** - Method in class `javax.servlet.ServletRequestAttributeEvent`
Return the name of the attribute that changed on the ServletRequest.

**getName()** - Method in class `javax.xml.bind.JAXBElement`
Returns the xml element tag name.

**getName()** - Method in interface
`javax.xml.registry.infomodel.RegistryObject`
Gets the user-friendly name of this object.

**getName()** - Method in interface `javax.xml.registry.infomodel.Slot`
Gets the name for this Slot.

**getName()** - Method in class `javax.xml.soap.MimeHeader`
Returns the name of this MimeHeader object.

**getName()** - Method in interface `javax.xml.stream.events.Attribute`
Returns the QName for this attribute.

**getName()** - Method in interface `javax.xml.stream.events.EndElement`
Get the name of this event.

**getName()** - Method in interface `javax.xml.stream.events.EntityDeclaration`
The entity's name.

**getName()** - Method in interface `javax.xml.stream.events.EntityReference`
The name of the entity.

**getName()** - Method in interface `javax.xml.stream.events.NotationTokenDeclaration`
The notation name.

**getName()** - Method in interface `javax.xml.stream.eventsStartElement`
Get the name of this event.

**getName()** - Method in class `javax.xml.stream.util.StreamReaderDelegate`
**getName()** - Method in interface javax.xml.stream.XMLStreamReader
   Returns a QName for the current START_ELEMENT or END_ELEMENT event

**getNamedDispatcher(String)** - Method in interface javax.servlet.ServletContext
   Returns a RequestDispatcher object that acts as a wrapper for the named servlet.

**getNameFromAttribute()** - Method in class javax.servlet.jsp.tagext.TagVariableInfo
   The body of the <name-from-attribute> element.

**getNameGiven()** - Method in class javax.servlet.jsp.tagext.TagVariableInfo
   The body of the <name-given> element.

**getNames()** - Method in class javax.activation.MimeTypeParameterList
   Retrieve an enumeration of all the names in this list.

**getNames()** - Method in class javax.mail.internet.ParameterList
   Return an enumeration of the names of all parameters in this list.

**getNamespaceContext()** - Method in interface javax.xml.stream.eventsStartElement
   Gets a read-only namespace context.

**getNamespaceContext()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**getNamespaceContext()** - Method in interface javax.xml.stream.XMLEventWriter
   Returns the current namespace context.

**getNamespaceContext()** - Method in interface javax.xml.stream.XMLStreamReader
   Returns a read only namespace context for the current position.

**getNamespaceContext()** - Method in interface javax.xml.stream.XMLStreamWriter
   Returns the current namespace context.

**getNamespaceCount()** - Method in class javax.xml.stream.util.StreamReaderDelegate
   Returns the count of namespaces declared on this START_ELEMENT or END_ELEMENT, this method is only valid on
getNamespacePrefix(int) - Method in class
javax.xml.stream.util.StreamReaderDelegate

getNamespacePrefix(int) - Method in interface
javax.xml.stream.XMLStreamReader

  Returns the prefix for the namespace declared at the index.

getNamespacePrefixes() - Method in interface
javax.xml.soap.SOAPElement

  Returns an Iterator over the namespace prefix Strings declared by
  this element.

getNamespaces() - Method in interface
javax.xml.stream.events.EndElement

  Returns an Iterator of namespaces that have gone out of scope.

getNamespaces() - Method in interface
javax.xml.stream.events.StartElement

  Returns an Iterator of namespaces declared on this element.

getNamespaceURI(String) - Method in interface
javax.xml.soap.SOAPElement

  Returns the URI of the namespace that has the given prefix.

getNamespaceURI() - Method in interface
javax.xml.stream.events.Namespace

  Gets the uri bound to the prefix of this namespace

getNamespaceURI(String) - Method in interface
javax.xml.stream.events.StartElement

  Gets the value that the prefix is bound to in the context of this
  element.

getNamespaceURI(String) - Method in class
javax.xml.stream.util.StreamReaderDelegate

getNamespaceURI(int) - Method in class
javax.xml.stream.util.StreamReaderDelegate

getNamespaceURI() - Method in class
javax.xml.stream.util.StreamReaderDelegate

getNamespaceURI(String) - Method in interface
javax.xml.stream.XMLStreamReader

  Return the uri for the given prefix.
**getNamespaceURI(int)** - Method in interface javax.xml.stream.XMLStreamReader
Returns the uri for the namespace declared at the index.

**getNamespaceURI()** - Method in interface javax.xml.stream.XMLStreamReader
If the current event is a START_ELEMENT or END_ELEMENT this method returns the URI of the prefix or the default namespace.

**getNativeCommands(String)** - Method in class javax.activation.MailcapCommandMap
Get the native commands for the given MIME type.

**getNavigationHandler()** - Method in class javax.faces.application.Application
Return the NavigationHandler instance that will be passed the outcome returned by any invoked application action for this web application.

**getNestedException()** - Method in exception javax.xml.stream.XMLStreamException
Gets the nested exception.

**getNewFolder()** - Method in class javax.mail.event.FolderEvent
If this event indicates that a folder is renamed, (i.e, the event type is RENAMED), then this method returns the Folder object representing the new name.

**getNewMessageCount()** - Method in class javax.mail.Folder
Get the number of new messages in this Folder.

**getNewsgroup()** - Method in class javax.mail.internet.NewsAddress
Get the newsgroup.

**getNewTempClassLoader()** - Method in interface javax.persistence.spi.PERSISTENCEUNITINFO
Return a new instance of a ClassLoader that the provider may use to temporarily load any classes, resources, or open URLs.

**getNewValue()** - Method in class javax.faces.event.ValueChangeEvent
Return the current local value of the source UIComponent.

**getNextException()** - Method in exception javax.mail.MessagingException
Get the next exception chained to this one.

**getNextTimeout()** - Method in interface javax.ejb.Timer
Get the point in time at which the next timer expiration is scheduled to occur.

**getNode(Object)** - Method in class
By default, the getNode method is unsupported and throw a
UnsupportedOperationException.

**getNode()** - Method in class
javax.xml.bind.helpers.**ValidationEventLocatorImpl**

**getNode(Object)** - Method in interface javax.xml.bind.**Marshaller**
Get a DOM tree view of the content tree(Optional).

**getNode()** - Method in interface javax.xml.bind.**ValidationEventLocator**
Return a reference to the DOM Node if available

**getNoLocal()** - Method in interface javax.jms.**TopicSubscriber**
Gets the NoLocal attribute for this subscriber.

**getNonJtaDataSource()** - Method in interface
javax.persistence.spi.**PersistenceUnitInfo**
Returns the non-JTA-enabled data source to be used by the persistence provider for accessing data outside a JTA transaction.

**getNonMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.**InternetHeaders**
Return all non-matching header lines

**getNonMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.**MimeBodyPart**
Get non-matching header lines as an Enumeration of Strings.

**getNonMatchingHeaderLines(String[])** - Method in class
javax.mail.internet.**MimeMessage**
Get non-matching header lines as an Enumeration of Strings.

**getNonMatchingHeaderLines(String[])** - Method in interface
javax.mail.internet.**MimePart**
Get non-matching header lines as an Enumeration of Strings.

**getNonMatchingHeaders(String[])** - Method in class
javax.mail.internet.**InternetHeaders**
Return all non-matching Header objects.

**getNonMatchingHeaders(String[])** - Method in class
javax.mail.internet.**MimeBodyPart**
Return non-matching headers from this Message as an Enumeration of Header objects.

**getNonMatchingHeaders(String[])** - Method in class
javax.mail.internet.**MimeMessage**
Return non-matching headers from this Message as an Enumeration of Header objects.
**getNonMatchingHeaders(String[])**  - Method in interface javax.mail.Part
- Return non-matching headers from this envelope as an Enumeration of Header objects.

**getNonMatchingHeaders(String[])**  - Method in class javax.xml.soap.MimeHeaders
- Returns all of the MimeHeaders objects whose name does not match a name in the given array of names.

**getNonMatchingMimeHeaders(String[])**  - Method in class javax.xml.soap.AttachmentPart
- Retrieves all MimeHeader objects whose name does not match a name in the given array.

**getNonMatchingMimeHeaders(String[])**  - Method in class javax.xml.soap.SOAPPart
- Retrieves all MimeHeader objects whose name does not match a name in the given array.

**getNonRunningModules(ModuleType, Target[])**  - Method in interface javax.enterprise.deploy.spi.DeploymentManager
- Retrieve the list of J2EE application modules distributed to the identified targets and that are currently not running on the associated server or servers.

**getNoNSSchemaLocation()**  - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl
- Convenience method for getting the current noNamespaceSchemaLocation.

**getNotationName()**  - Method in interface javax.xml.stream.events.EntityDeclaration
- The name of the associated notation.

**getNotations()**  - Method in interface javax.xml.stream.events/DTD
- Return a List containing the notations declared in the DTD.

**getNumber()**  - Method in class javax.mail.search.IntegerComparisonTerm
- Return the number to compare with.

**getNumber()**  - Method in interface javax.xml.registry.infomodel.TelphoneNumber
- Gets the telephone number suffix, not including the country or area code.

**getObject(String)**  - Method in interface javax.jms.MapMessage
- Returns the value of the object with the specified name.

**getObject()**  - Method in interface javax.jms.ObjectMessage
Gets the serializable object containing this message's data.

**getObject()** - Method in class
javax.xml.bind.helpers.**ValidationEventLocatorImpl**

**getObject()** - Method in interface javax.xml.bind.**ValidationEventLocator**
Return a reference to the object in the Java content tree if available

**getableObjectProperty(String)** - Method in interface javax.jms.**Message**
Returns the value of the Java object property with the specified name.

**getObjectType()** - Method in interface
javax.xml.registry.infomodel.**RegistryObject**
Gets the object type that best describes the RegistryObject.

**getOffset()** - Method in class
javax.enterprise.deploy.shared.**ActionType**
Returns the lowest integer value used by this enumeration value's enumeration class.

**getOffset()** - Method in class
javax.enterprise.deploy.shared.**CommandType**
Returns the lowest integer value used by this enumeration value's enumeration class.

**getOffset()** - Method in class
javax.enterprise.deploy.shared.**DConfigBeanVersionType**
Returns the lowest integer value used by this enumeration value's enumeration class.

**getOffset()** - Method in class
javax.enterprise.deploy.shared.**ModuleType**
Returns the lowest integer value used by this enumeration value's enumeration class.

**getOffset()** - Method in class
javax.enterprise.deploy.shared.**StateType**
Returns the lowest integer value used by this enumeration value's enumeration class.

**getOffset()** - Method in class
javax.xml.bind.helpers.**ValidationEventLocatorImpl**

**getOffset()** - Method in interface javax.xml.bind.**ValidationEventLocator**
Return the byte offset if available

**getOldValue()** - Method in class javax.faces.event.**ValueChangeEvent**
Return the previous local value of the source **UIComponent**.

**getOnblur()** - Method in class
javax.faces.component.html.**HtmlCommandButton**
Return the value of the onblur property.
**getOnblur** - Method in class javax.faces.component.html.HtmlCommandLink
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlInputSecret
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlInputText
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlInputTextarea
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlOutputLabel
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlOutputLink
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onblur property.

**getOnblur** - Method in class javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onblur property.

**getOnchange** - Method in class
getOnchange() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlInputSecret
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onchange property.

getOnchange() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onchange property.

getOnclick() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the onclick property.

getOnclick() - Method in class javax.faces.component.html.HtmlForm
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlInputSecret
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onclick property.

getOnclick() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onclick property.

**getOnclick()** - Method in class
javax.faces.component.html.HttpSelectOneMenu
   Return the value of the onclick property.

**getOnclick()** - Method in class
javax.faces.component.html.HttpSelectOneRadio
   Return the value of the onclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HttpCommandButton
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HttpCommandLink
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlDataTable
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlForm
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlGraphicImage
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlInputSecret
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlInputText
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlInputTextarea
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlOutputLabel
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlOutputLink
   Return the value of the ondblclick property.

**getOndblclick()** - Method in class
javax.faces.component.html.HtmlPanelGrid
   Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectManyListbox
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectManyMenu
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectOneListbox
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectOneMenu
    Return the value of the ondblclick property.
getOndblclick() - Method in class javax.faces.component.html.HtmlSelectOneRadio
    Return the value of the ondblclick property.
getOnfocus() - Method in class javax.faces.component.html.HtmCommandButton
    Return the value of the onfocus property.
getOnfocus() - Method in class javax.faces.component.html.HtmCommandLink
    Return the value of the onfocus property.
getOnfocus() - Method in class javax.faces.component.html.HtmInputSecret
    Return the value of the onfocus property.
getOnfocus() - Method in class javax.faces.component.html.HtmInputText
    Return the value of the onfocus property.
getOnfocus() - Method in class javax.faces.component.html.HtmInputTextarea
    Return the value of the onfocus property.
getOnfocus() - Method in class javax.faces.component.html.HtmOutputLabel
    Return the value of the onfocus property.
**getOnfocus()** - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onfocus property.

**getOnfocus()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onfocus property.

**getOnkeydown()** - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onkeydown property.

**getOnkeydown()** - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the onkeydown property.

**getOnkeydown()** - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the onkeydown property.

**getOnkeydown()** - Method in class
javax.faces.component.html.HtmlForm
Return the value of the onkeydown property.

**getOnkeydown()** - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the onkeydown property.
`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlInputSecret`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlInputText`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlInputTextarea`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlOutputLabel`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlOutputLink`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlPanelGrid`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
Return the value of the `onkeydown` property.

`getOnkeydown()` - Method in class `javax.faces.component.html.HtmlSelectOneRadio`
Return the value of the `onkeydown` property.
**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmCommandButton
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmCommandLink
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmDataTable
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmForm
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmGraphicImage
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmInputSecret
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmInputText
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmInputTextarea
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmOutputLabel
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmOutputLink
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmPanelGrid
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmSelectBooleanCheckbox
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class
javafx.faces.component.html.HtmSelectManyCheckbox
Return the value of the onkeypress property.
**getOnkeypress()** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
Return the value of the onkeypress property.

**getOnkeypress()** - Method in class `javax.faces.component.html.HtmlSelectOneRadio`
Return the value of the onkeypress property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlCommandButton`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlCommandLink`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlDataTable`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlForm`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlGraphicImage`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlInputSecret`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlInputText`
Return the value of the onkeyup property.

**getOnkeyup()** - Method in class `javax.faces.component.html.HtmlInputTextarea`
Return the value of the onkeyup property.
javax.faces.component.html.HtmlOutputLabel
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onkeyup property.

getOnkeyup() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onkeyup property.

getOnmousedown() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onmousedown property.

getOnmousedown() - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the onmousedown property.

getOnmousedown() - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the onmousedown property.

getOnmousedown() - Method in class
javax.faces.component.html.HtmlForm

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlGraphicImage

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlInputSecret

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlInputText

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlInputTextarea

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlOutputLabel

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlOutputLink

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlPanelGrid

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlSelectBooleanCheckbox

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlSelectManyCheckbox

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlSelectManyListbox

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlSelectManyMenu

Return the value of the onmousedown property.

getOnmousedown() - Method in class

javax.faces.component.html.HtmlSelectOneListbox

Return the value of the onmousedown property.

getOnmousedown() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onmousedown property.
**getOnmousedown()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onmousedown property.
**getOnmousedown()** - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlForm
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlInputSecret
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the onmousemove property.
**getOnmousemove()** - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the onmousemove property.
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onmousemove property.

getOnmousemove() - Method in class

javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onmousemove property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlCommandButton
Return the value of the onmouseout property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlCommandLink
Return the value of the onmouseout property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlDataTable
Return the value of the onmouseout property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlForm
Return the value of the onmouseout property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlGraphicImage
Return the value of the onmouseout property.

getOnmouseout() - Method in class

javax.faces.component.html.HtmlInputSecret
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onmouseout property.

getOnmouseout() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onmouseout property.

getOnmouseover() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onmouseover property.

getOnmouseover() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onmouseover property.
javax.faces.component.html.HtmlCommandLink
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlDataTable
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlForm
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlGraphicImage
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlInputSecret
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlInputText
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlInputTextarea
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlOutputLabel
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlOutputLink
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlPanelGrid
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
  Return the value of the onmouseover property.
getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
  Return the value of the onmouseover property.
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the onmouseover property.

getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the onmouseover property.

getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Return the value of the onmouseover property.

getOnmouseover() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Return the value of the onmouseover property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlCommandButton
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlCommandLink
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlForm
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlInputSecret
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlInputText
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlOutputLabel
javax.faces.component.html.HtmlOutputLink
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlPanelGrid
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
   Return the value of the onmouseup property.

getOnmouseup() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
   Return the value of the onmouseup property.

getOnreset() - Method in class javax.faces.component.html.HtmlForm
   Return the value of the onreset property.

getOnselect() - Method in class
javax.faces.component.html.HtmlCommandButton
   Return the value of the onselect property.

getOnselect() - Method in class
javax.faces.component.html.HtmlInputSecret
   Return the value of the onselect property.

getOnselect() - Method in class
javax.faces.component.html.HtmlInputText
   Return the value of the onselect property.

getOnselect() - Method in class
javax.faces.component.html.HtmlInputTextarea
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`  
Return the value of the `onselect` property.

`getOnselect()` - Method in class `javax.faces.component.html.HtmlSelectOneRadio`  
Return the value of the `onselect` property.

`getOnsubmit()` - Method in class `javax.faces.component.html.HtmlForm`  
Return the value of the `onsubmit` property.

`getOperationName()` - Method in interface `javax.xml.rpc.Call`  
Gets the name of the operation to be invoked using this `Call` instance.

`getOrdinal()` - Method in class `javax.faces.application.FacesMessage.Severity`  
Return the ordinal value of this `FacesMessage.Severity` instance.

`getOrdinal()` - Method in class `javax.faces.event.PhaseId`  
Return the ordinal value of this `PhaseId` instance.

`getOrganization()` - Method in interface `javax.xml.registry.infomodel.User`  
Gets the Organization that this User is affiliated with.

`getOrigin()` - Method in interface `javax.management.j2ee.statistics.JMSSConsumerStats`  
Returns a string that encapsulates the identity of a message origin.

`getOut()` - Method in class `javax.servlet.jsp.JspContext`
The current value of the `out` object (a `JspWriter`).

**`getOutputParams()`** - Method in interface `javax.xml.rpc.Call`
Returns a Map of `{name, value}` for the output parameters of the last invoked operation.

**`getOutputStream()`** - Method in class `javax.activation.DataHandler`
Get an OutputStream for this DataHandler to allow overwriting the underlying data.

**`getOutputStream()`** - Method in interface `javax.activation.DataSource`
This method returns an `OutputStream` where the data can be written and throws the appropriate exception if it cannot do so.

**`getOutputStream()`** - Method in class `javax.activation.FileDataSources`
This method will return an `OutputStream` representing the data and will throw an IOException if it cannot do so.

**`getOutputStream()`** - Method in class `javax.activation.URLDataSource`
The `getOutputStream` method from the URL.

**`getOutputStream()`** - Method in class `javax.mail.internet.MimePartDataSource`
DataSource method to return an output stream.

**`getOutputStream()`** - Method in class `javax.mail.util.ByteArrayDataSource`
Return an `OutputStream` for the data.

**`getOutputValues()`** - Method in interface `javax.xml.rpc.Call`
Returns a List values for the output parameters of the last invoked operation.

**`getPage()`** - Method in class `javax.servlet.jsp.PageContext`
The current value of the page object (In a Servlet environment, this is an instance of `javax.servlet.Servlet`).

**`getPageContext(Servlet, ServletRequest, ServletResponse, String, boolean, int, boolean)`** - Method in class `javax.servlet.jsp.JspFactory`
Obtains an instance of an implementation dependent `javax.servlet.jsp.PageContext` abstract class for the calling Servlet and currently pending request and response.
**getParameter(String)** - Method in class `javax.activation.MimeType`  
Retrieve the value associated with the given name, or null if there is no current association.

**getParameter(String)** - Method in class `javax.mail.internet.ContentDisposition`  
Return the specified parameter value.

**getParameter(String)** - Method in class `javax.mail.internet.ContentType`  
Return the specified parameter value.

**getParameter(String)** - Method in interface `javax.servletServletRequest`  
Returns the value of a request parameter as a String, or null if the parameter does not exist.

**getParameter(String)** - Method in class `javax.servlet.ServletRequestWrapper`  
The default behavior of this method is to return `getParameter(String name)` on the wrapped request object.

**getParameterList()** - Method in class `javax.mail.internet.ContentDisposition`  
Return a ParameterList object that holds all the available parameters.

**getParameterList()** - Method in class `javax.mail.internet.ContentType`  
Return a ParameterList object that holds all the available parameters.

**getParameterMap()** - Method in interface `javax.servletServletRequest`  
Returns a java.util.Map of the parameters of this request.

**getParameterMap()** - Method in class `javax.servletServletRequestWrapper`  
The default behavior of this method is to return `getParameterMap()` on the wrapped request object.

**getParameterNames()** - Method in interface `javax.servletServletRequest`  
Returns an Enumeration of String objects containing the names of the parameters contained in this request.

**getParameterNames()** - Method in class `javax.servletServletRequestWrapper`  
The default behavior of this method is to return `getParameterNames()` on the wrapped request object.

**getParameters()** - Method in class `javax.activation.MimeType`  
Retrieve this object's parameter list.

**getParameters()** - Method in interface
javax.interceptor.**InvocationContext**
   Returns the parameters that will be used to invoke the business method.

**getParameterTypeByName(String)** - Method in interface javax.xml.rpc.**Call**
   Gets the XML type of a parameter by name

**getParameterValues(String)** - Method in interface javax.servlet.**ServletRequest**
   Returns an array of String objects containing all of the values the
given request parameter has, or null if the parameter does not exist.

**getParameterValues(String)** - Method in class javax.servlet.**ServletRequestWrapper**
   The default behavior of this method is to return
   getParameterValues(String name) on the wrapped request object.

**getParamTypes()** - Method in class javax.el.**MethodInfo**
   Returns the parameter types of the method

**getParent()** - Method in class javax.faces.component.**UIComponent**
   Return the parent **UIComponent** of this UIComponent, if any.

**getParent()** - Method in class javax.faces.component.**UIComponentBase**

**getParent()** - Method in class javax.faces.webapp.**UIComponentClassicTagBase**
   Return the Tag that is the parent of this instance.

**getParent()** - Method in class javax.mail.**BodyPart**
   Return the containing Multipart object, or null if not known.

**getParent()** - Method in class javax.mail.**Folder**
   Returns the parent folder of this folder.

**getParent()** - Method in class javax.mail.**Multipart**
   Return the Part that contains this Multipart object, or null if not known.

**getParent()** - Method in interface javax.servlet.jsp.tagext.**SimpleTag**
   Returns the parent of this tag, for collaboration purposes.

**getParent()** - Method in class javax.servlet.jsp.tagext.**SimpleTagSupport**
   Returns the parent of this tag, for collaboration purposes.

**getParent()** - Method in interface javax.servlet.jsp.tagext.**Tag**
   Get the parent (closest enclosing tag handler) for this tag handler.

**getParent()** - Method in class javax.servlet.jsp.tagext.**TagAdapter**
   Returns the parent of this tag, which is always
   getAdaptee().getParent().
**getParent()** - Method in class `javax.servlet.jsp.tagext.TagSupport`
The Tag instance most closely enclosing this tag instance.

**getParent()** - Method in interface `javax.xml.registry.infomodel.Concat` or `javax.xml.registry.infomodel.ClassificationScheme`
Gets the parent Concept or ClassificationScheme for this object.

**getParent()** - Method in class `javax.xml.stream.util.EventReaderDelegate`
Get the parent of this instance.

**getParent()** - Method in class `javax.xml.stream.util.StreamReaderDelegate`
Get the parent of this instance.

**getParentConcept()** - Method in interface `javax.xml.registry.infomodel.Concat` or `javax.xml.registry.infomodel.ClassificationScheme`
Gets the parent Concept or null if parent is a ClassificationScheme.

**getParentElement()** - Method in interface `javax.xml.soap.Node`
Returns the parent element of this `Node` object.

**getParentOrganization()** - Method in interface `javax.xml.registry.infomodel.Organization`
Gets the parent (container) organization.

**getParentTargetModuleID()** - Method in interface `javax.enterprise.deploy.spi.TargetModuleID`
Retrieve the identifier of the parent object of this deployed module.

**getParentUIComponentClassicTagBase(PageContext)** - Static method in class `javax.faces.webapp.UIComponentClassicTagBase`
Locate and return the nearest enclosing `UIComponentClassicTagBase` if any; otherwise, return `null`.

**getParentUIComponentTag(PageContext)** - Static method in class `javax.faces.webapp.UIComponentTag`
Deprecated. Locate and return the nearest enclosing `UIComponentTag` if any; otherwise, return `null`.

**getPart()** - Method in class `javax.mail.MessageContext`
Return the Part that contains the content.

**getPassiveCount()** - Method in interface `javax.management.j2ee.statistics.StatefulSessionBeanStats`
Number of beans that are in the passive state.

**getPassword()** - Method in class `javax.mail.PasswordAuthentication`
Returns the password of this `PasswordAuthentication`.

**getPassword()** - Method in class `javax.mail.URLName`
Returns the password of this `URLName`.

**getPassword()** - Method in class `java.util.DriverManager`
javax.resource.spi.security.PasswordCredential
   Returns the user password.
**getPasswordAuthentication()** - Method in class javax.mail.Authenticator
   Called when password authentication is needed.
**getPasswordAuthentication(URLName)** - Method in class javax.mail.Session
   Return any saved PasswordAuthentication for this (store or transport) URLName.
**getPath()** - Method in class javax.servlet.http.Cookie
   Returns the path on the server to which the browser returns this cookie.
**getPath()** - Method in class javax.servlet.jsp.tagext.TagFileInfo
   Where to find the .tag file implementing this action.
**getPath()** - Method in interface javax.xml.registry.infomodel.Concept
   Gets the canonical path representation for this Concept.
   Returns any extra path information associated with the URL the client sent when it made this request.
**getPathInfo()** - Method in class javax.servlet.http.HttpServletRequestWrapper
   The default behavior of this method is to return getPathInfo() on the wrapped request object.
**getPathTranslated()** - Method in interface javax.servlet.http.HttpServletRequest
   Returns any extra path information after the servlet name but before the query string, and translates it to a real path.
**getPathTranslated()** - Method in class javax.servlet.http.HttpServletRequestWrapper
   The default behavior of this method is to return getPathTranslated() on the wrapped request object.
**getPattern()** - Method in class javax.faces.convert.DateTimeConverter
   Return the format pattern to be used when formatting and parsing dates and times.
**getPattern()** - Method in class javax.faces.convert.NumberConverter
   Return the format pattern to be used when formatting and parsing numbers.
**getPattern()** - Method in class javax.mail.search.StringTerm
   Return the string to match with.
**getPayload()** - Method in interface `javax.xml.ws.LogicalMessage`

Gets the message payload as an XML source, may be called multiple times on the same `LogicalMessage` instance, always returns a new Source that may be used to retrieve the entire message payload.

**getPayload(JAXBContext)** - Method in interface `javax.xml.ws.LogicalMessage`

 Gets the message payload as a JAXB object.

**getPendingMessageCount()** - Method in interface `javax.management.j2ee.statistics.JMSEndpointStats`

Number of pending messages.

**getPendingMessageCount()** - Method in interface `javax.management.j2ee.statistics.JMSSessionStats`

Number of pending messages.

**getPermanentFlags()** - Method in class `javax.mail.Folder`

Get the permanent flags supported by this Folder.

**getPersistenceProviderClassName()** - Method in interface `javax.persistence.spi.PersistenceUnitInfo`

Returns the fully qualified name of the persistence provider implementation class.

**getPersistenceUnitName()** - Method in interface `javax.persistence.spi.PersistenceUnitInfo`

Returns the name of the persistence unit.

**getPersistenceUnitRootUrl()** - Method in interface `javax.persistence.spi.PersistenceUnitInfo`

Returns the URL for the jar file or directory that is the root of the persistence unit.

**getPersonal()** - Method in class `javax.mail.internet.InternetAddress`

Get the personal name.

**getPersonalNamespaces()** - Method in class `javax.mail.Store`

Return a set of folders representing the *personal* namespaces for the current user.

**getPersonName()** - Method in interface `javax.xml.registry.infomodel.User`

Returns the name of this User.

**getPhaseId()** - Method in class `javax.faces.event.FacesEvent`

Return the identifier of the request processing phase during which this event should be delivered.

**getPhaseId()** - Method in class `javax.faces.event.PhaseEvent`
Return the PhaseId representing the current request processing lifecycle phase.

**getPhaseId()** - Method in interface javax.faces.event.PhaseListener

Return the identifier of the request processing phase during which this listener is interested in processing PhaseEvent events.

**getPhaseListeners()** - Method in class javax.faces.lifecycle.Lifecycle

Return the set of registered PhaseListener for this Lifecycle instance.

**getPIData()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**getPIData()** - Method in interface javax.xml.stream.XMLStreamReader

Get the data section of a processing instruction

**getPITarget()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**getPITarget()** - Method in interface javax.xml.stream.XMLStreamReader

Get the target of a processing instruction

**getPolicyConfiguration(String, boolean)** - Method in class javax.security.jacc.PolicyConfigurationFactory

This method is used to obtain an instance of the provider specific class that implements the PolicyConfiguration interface that corresponds to the identified policy context within the provider.

**getPolicyConfigurationFactory()** - Static method in class javax.security.jacc.PolicyConfigurationFactory

This static method uses a system property to find and instantiate (via a public constructor) a provider specific factory implementation class.

**getPooledCount()** - Method in interface javax.management.j2ee.statistics.EntityBeanStats

Number of beans in the pooled state.

**getPoolSize()** - Method in interface javax.management.j2ee.statistics.JCACredentialPoolStats

The size of the connection pool

**getPoolSize()** - Method in interface javax.management.j2ee.statistics.JDBCConnectionPoolStats

Size of the connection pool.

**getPort()** - Method in class javax.mail.URLName

Returns the port number of this URLName.

**getPort(QName, Class)** - Method in interface javax.xml.rpc.Service
The `getPort` method returns either an instance of a generated stub implementation class or a dynamic proxy.

**getPort(Class) - Method in interface javax.xml.rpc.Service**
- The `getPort` method returns either an instance of a generated stub implementation class or a dynamic proxy.

**getPort(QName, Class<T>) - Method in class javax.xml.ws.Service**
- The `getPort` method returns a stub.

**getPort(Class<T>) - Method in class javax.xml.ws.Service**
- The `getPort` method returns a stub.

**getPort(QName, Class<T>) - Method in class javax.xml.ws.spi.ServiceDelegate**
- The `getPort` method returns a stub.

**getPort(Class<T>) - Method in class javax.xml.ws.spi.ServiceDelegate**
- The `getPort` method returns a stub.

**getPortName() - Method in interface javax.xml.ws.handler.PortInfo**
- Gets the qualified name of the WSDL port being accessed.

**getPorts() - Method in interface javax.xml.rpc.Service**
- Returns an Iterator for the list of QNameS of service endpoints grouped by this service.

**getPorts() - Method in class javax.xml.ws.Service**
- Returns an Iterator for the list of QNameS of service endpoints grouped by this service.

**getPorts() - Method in class javax.xml.ws.spi.ServiceDelegate**
- Returns an Iterator for the list of QNameS of service endpoints grouped by this service.

**getPortTypeName() - Method in interface javax.xml.rpc.Call**
- Gets the qualified name of the port type.

**getPos() - Method in exception javax.mail.internet.AddressException**
- Get the position with the reference string where the error was detected (-1 if not relevant).

**getPosition() - Method in interface javax.mail.internet.SharedInputStream**
- Return the current position in the InputStream, as an offset from the beginning of the InputStream.

**getPosition() - Method in class javax.mail.util.SharedByteArrayInputStream**
- Return the current position in the InputStream, as an offset from the beginning of the InputStream.

**getPosition() - Method in class javax.mail.util.SharedFileInputStream**
Return the current position in the InputStream, as an offset from the beginning of the InputStream.

**getPostalAddress()** - Method in interface
javax.xml.registry.infomodel.**Organization**
Gets the Address for this Organization.

**getPostalAddresses()** - Method in interface
javax.xml.registry.infomodel.**User**
Gets the postal address for this User.

**getPostalCode()** - Method in interface
javax.xml.registry.infomodel.**PostalAddress**
Returns the postal or zip code.

**getPostalScheme()** - Method in interface
javax.xml.registry.infomodel.**PostalAddress**
Returns a user-defined postal scheme for codifying the attributes of PostalAddress.

**getPreamble()** - Method in class javax.mail.internet.**MimeMultipart**
Get the preamble text, if any, that appears before the first body part of this multipart.

**getPreferredCommands(String)** - Method in class
javax.activation.**CommandMap**
Get the preferred command list from a MIME Type.

**getPreferredCommands(String, DataSource)** - Method in class
javax.activation.**CommandMap**
Get the preferred command list from a MIME Type.

**getPreferredCommands()** - Method in class
javax.activation.**DataHandler**
Return the preferred commands for this type of data.

**getPreferredCommands(String)** - Method in class
javax.activation.**MailcapCommandMap**
Get the preferred command list for a MIME Type.

**getPrefix()** - Method in interface javax.xml.soap.**Name**
Returns the prefix that was specified when this Name object was initialized.

**getPrefix()** - Method in interface javax.xml.stream.events.**Namespace**
Gets the prefix, returns "" if this is a default namespace declaration.

**getPrefix()** - Method in class
javax.xml.stream.util.**StreamReaderDelegate**

**getPrefix(String)** - Method in interface
javax.xml.stream.**XMLEventWriter**

Gets the prefix the uri is bound to

**getPrefix()** - Method in interface javax.xml.stream.**XMLStreamReader**

Returns the prefix of the current event or null if the event does not have a prefix

**getPrefix(String)** - Method in interface javax.xml.stream.**XMLStreamWriter**

Gets the prefix the uri is bound to

**getPrefixString()** - Method in class javax.servlet.jsp.tagext.**TagLibraryInfo**

The prefix assigned to this taglib from the taglib directive

**getPreviousOut()** - Method in class javax.faces.webapp.**UIComponentClassicTagBase**

Get the JspWriter from our BodyContent.

**getPreviousOut()** - Method in class javax.servlet.jsp.tagext.**BodyTagSupport**

Get surrounding out JspWriter.

**getPrimaryContact()** - Method in interface javax.xml.registry.infomodel.**Organization**

Gets the primary Contact for this Organization.

**getPrimaryKey()** - Method in interface javax.ejb.**EJBLocalObject**

Obtain the primary key of the EJB local object.

**getPrimaryKey()** - Method in interface javax.ejb.**EJBObject**

Obtain the primary key of the EJB object.

**getPrimaryKey()** - Method in interface javax.ejb.**EntityContext**

Obtain the primary key of the EJB object that is currently associated with this instance.

**getPrimaryKeyClass()** - Method in interface javax.ejb.**EJBMetaData**

Obtain the Class object for the enterprise Bean's primary key class.

**getPrimaryType()** - Method in class javax.activation.**MimeType**

Retrieve the primary type of this object.

**getPrimaryType()** - Method in class javax.mail.internet.**ContentType**

Return the primary type.

**getPriority()** - Method in interface javax.jms.**MessageProducer**

Gets the producer's default priority.

**getProcessedDTD()** - Method in interface javax.xml.stream.events.**DTD**

Returns an implementation defined representation of the DTD.

**getProducers()** - Method in interface javax.management.j2ee.statistics.**JMSSessionStats**
Returns an array of JMSProducerStats that provide statistics about the message producers associated with the referencing JMS session statistics.

**getProductVersion()** - Method in interface
javax.enterprise.deploy.spi.factories.**DeploymentFactory**
Provide a string identifying version of this vendor's DeploymentManager.

**getProperties()** - Method in class javax.mail.**Session**
Returns the Properties object associated with this Session

**getProperties()** - Method in interface
javax.persistence.spi.**PersistenceUnitInfo**
Returns properties object.

**getProperties()** - Method in class javax.xml.registry.**ConnectionFactory**
Gets the Properties used during createConnection and createFederatedConnection calls.

**getProperties()** - Method in class javax.xml.ws.**Endpoint**
Returns the property bag for this Endpoint instance.

**getProperty(String)** - Method in class javax.mail.**Session**
Returns the value of the specified property.

**getProperty(String)** - Method in class javax.xml.bind.**Binder**
Get the particular property in the underlying implementation of Binder.

**getProperty(String)** - Method in class
javax.xml.bind.helpers.**AbstractMarshallerImpl**
Default implementation of the getProperty method handles the four defined properties in Marshaller.

**getProperty(String)** - Method in class
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**
Default implementation of the getProperty method always throws PropertyException since there are no required properties.

**getProperty(String)** - Method in interface javax.xml.bind.**Marshaller**
Get the particular property in the underlying implementation of Marshaller.

**getProperty(String)** - Method in interface javax.xml.bind.**Unmarshaller**
Get the particular property in the underlying implementation of Unmarshaller.

**getProperty(String)** - Method in interface javax.xml.bind.**Validator**
*Deprecated. since JAXB2.0*

**getProperty(String)** - Method in interface javax.xml.rpc.**Call**
Gets the value of a named property.

**`getProperty(String)` - Method in interface javax.xml.rpc.handler.MessageContext**
- Gets the value of a specific property from the MessageContext

**`getProperty(String)` - Method in class javax.xml.soap.SOAPMessage**
- Retrieves value of the specified property.

**`getProperty(String)` - Method in class javax.xml.stream.util.EventReaderDelegate**

**`getProperty(String)` - Method in class javax.xml.stream.util.StreamReaderDelegate**

**`getProperty(String)` - Method in interface javax.xml.stream.XMLEventReader**
- Get the value of a feature/property from the underlying implementation

**`getProperty(String)` - Method in class javax.xml.stream.XMLInputFactory**
- Get the value of a feature/property from the underlying implementation

**`getProperty(String)` - Method in class javax.xml.stream.XMLOutputFactory**
- Get a feature/property on the underlying implementation

**`getProperty(String)` - Method in interface javax.xml.stream.XMLStreamReader**
- Get the value of a feature/property from the underlying implementation

**`getProperty(String)` - Method in interface javax.xml.stream.XMLStreamWriter**
- Get the value of a feature/property from the underlying implementation

**`getPropertyNames()` - Method in interface javax.jms.Message**
- Returns an Enumeration of all the property names.

**`getPropertyNames()` - Method in interface javax.xml.rpc.Call**
- Gets the names of configurable properties supported by this Call object.

**`getPropertyNames()` - Method in interface javax.xml.rpc.handler.MessageContext**
- Returns an Iterator view of the names of the properties in this
**MessageContext**

**getPropertyResolver()** - Method in class javax.faces.application.Application

*Deprecated.* This has been replaced by Application.getELResolver().

**getPropertyType()** - Method in class javax.el.BeanELResolver.BeanProperty

**getProtocol()** - Method in class javax.mail.Provider

Returns the protocol supported by this Provider.

**getProtocol()** - Method in class javax.mail.URLName

Returns the protocol of this URLName.

**getProtocol()** - Method in interface javax.mail.URLName

Returns the name and version of the protocol the request uses in the form protocol/majorVersion.minorVersion, for example, HTTP/1.1.

**getProtocol()** - Method in class javax.servlet.HttpServletRequest

The default behavior of this method is to return getProtocol() on the wrapped request object.

**getProvider(String)** - Method in class javax.mail.Session

Returns the default Provider for the protocol specified.

**getProviderMajorVersion()** - Method in interface javax.jms.ConnectionMetaData

Gets the JMS provider major version number.

**getProviderMinorVersion()** - Method in interface javax.jms.ConnectionMetaData

Gets the JMS provider minor version number.

**getProviders()** - Method in class javax.mail.Session

This method returns an array of all the implementations installed via the javamail.

**getProviderVersion()** - Method in interface javax.jms.ConnectionMetaData

Gets the JMS provider version.

**getProvidingOrganization()** - Method in interface javax.xml.registry.infomodel.Service

Gets the Organization that provides this service.

**getPublicId()** - Method in interface javax.xml.stream.events.EntityDeclaration

The entity’s public identifier, or null if none was given.

**getPublicId()** - Method in interface
javax.xml.stream.events.\texttt{NotationDeclaration}

The notation's public identifier, or null if none was given.

\texttt{getPublicId()} - Method in interface javax.xml.stream.\texttt{Location}

Returns the public ID of the XML

\texttt{getQualifiedName()} - Method in interface javax.xml.soap.\texttt{Name}

Gets the namespace-qualified name of the XML name that this \texttt{Name}
object represents.

\texttt{getQueryString()} - Method in interface
javax.servlet.http.\texttt{HttpServletRequest}

Returns the query string that is contained in the request URL after
the path.

\texttt{getQueryString()} - Method in class
javax.servlet.http.\texttt{HttpServletRequestWrapper}

The default behavior of this method is to return \texttt{getQueryString()} on
the wrapped request object.

\texttt{getQueue()} - Method in interface javax.jms.\texttt{QueueBrowser}

Gets the queue associated with this queue browser.

\texttt{getQueue()} - Method in interface javax.jms.\texttt{QueueReceiver}

Gets the queue associated with this queue receiver.

\texttt{getQueue()} - Method in interface javax.jms.\texttt{QueueSender}

Gets the queue associated with this \texttt{QueueSender}.

\texttt{getQueueName()} - Method in interface javax.jms.\texttt{Queue}

Gets the name of this queue.

\texttt{getQueueSession()} - Method in interface javax.jms.\texttt{XAQueueSession}

Gets the queue session associated with this \texttt{XAQueueSession}.

\texttt{getQuota(String)} - Method in interface javax.mail.\texttt{QuotaAwareStore}

Get the quotas for the named quota root.

\texttt{getRawContent()} - Method in class javax.xml.soap.\texttt{AttachmentPart}

Gets the content of this \texttt{AttachmentPart} object as an InputStream as
if a call had been made to \texttt{getContent} and no \texttt{DataContentHandler}
had been registered for the \texttt{content-type} of this \texttt{AttachmentPart}.

\texttt{getRawContentBytes()} - Method in class
javax.xml.soap.\texttt{AttachmentPart}

Gets the content of this \texttt{AttachmentPart} object as a byte[] array as if
a call had been made to \texttt{getContent} and no \texttt{DataContentHandler}
had been registered for the \texttt{content-type} of this \texttt{AttachmentPart}.

\texttt{getRawInputStream()} - Method in class
javax.mail.internet.\texttt{MimeBodyPart}

Return an InputStream to the raw data with any Content-Transfer-
getRawInputStream() - Method in class javax.mail.internet.MimeMessage
Return an InputStream to the raw data with any Content-Transfer-Encoding intact.

getReader() - Method in class javax.servlet.jsp.tagext.BodyContent
Return the value of this BodyContent as a Reader.

getReader() - Method in interface javax.servlet.ServletRequest
Retrieves the body of the request as character data using a BufferedReader.

getReader() - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return getReader() on the wrapped request object.

getReadMethod() - Method in class javax.el.BeanELResolver.BeanProperty

getReadyCount() - Method in interface javax.management.j2ee.statistics.EntityBeanStats
Number of beans in the ready state.

getRealPath(String) - Method in interface javax.servlet.ServletContext
Returns a String containing the real path for a given virtual path.

getRealPath(String) - Method in interface javax.servlet.ServletRequest
 Deprecated. As of Version 2.1 of the Java Servlet API, use ServletContext.getRealPath(java.lang.String) instead.

getRealPath(String) - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return getRealPath(String path) on the wrapped request object.

getReceivedDate() - Method in class javax.mail.internet.MimeMessage
Returns the Date on this message was received.

getReceivedDate() - Method in class javax.mail.Message
Get the date this message was received.

getRecipients(Message.RecipientType) - Method in class javax.mail.internet.MimeMessage
Returns the recipients specified by the type.

getRecipients(Message.RecipientType) - Method in class javax.mail.Message
Get all the recipient addresses of the given type.

getRecipientType() - Method in class
javax.mail.search.**RecipientStringTerm**
   Return the type of recipient to match with.

**getRecipientType()** - Method in class javax.mail.search.**RecipientTerm**
   Return the type of recipient to match with.

**getRecordFactory()** - Method in interface javax.resource.cci.**ConnectionFactory**
   Gets a RecordFactory instance.

**getRecordName()** - Method in interface javax.resource.cci.**Record**
   Gets the name of the Record.

**getRecordShortDescription()** - Method in interface javax.resource.cci.**Record**
   Gets a short description string for the Record.

**getRef()** - Method in exception javax.mail.internet.**AddressException**
   Get the string that was being parsed when the error was detected (null if not relevant).

**getRef()** - Method in class javax.mail.**URLName**
   Returns the reference of this URLName.

**getReference(Class<T>, Object)** - Method in interface javax.persistence.**EntityManager**
   Get an instance, whose state may be lazily fetched.

**getRegisteredEncodingStyleURIs()** - Method in interface javax.xml.rpc.encoding.**TypeMappingRegistry**
   Returns a list of registered encodingStyle URIs in this TypeMappingRegistry instance.

**getRegistryObject()** - Method in interface javax.xml.registry.infomodel.**AuditableEvent**
   Gets the RegistryObject associated with this AuditableEvent.

**getRegistryObject()** - Method in interface javax.xml.registry.infomodel.**ExternalIdentifier**
   Gets the parent RegistryObject for this ExternalIdentifier.

**getRegistryObject(String, String)** - Method in interface javax.xml.registry.**QueryManager**
   Gets the RegistryObject specified by the Id and type of object.

**getRegistryObject(String)** - Method in interface javax.xml.registry.**QueryManager**
   Gets the RegistryObject specified by the Id.

**getRegistryObjects()** - Method in interface javax.xml.registry.infomodel.**RegistryPackage**
   Gets the collection of member RegistryObjects of this
RegistryPackage.

**getRegistryObjects(Collection)** - Method in interface javax.xml.registry.**QueryManager**
- Gets the specified RegistryObjects.

**getRegistryObjects(Collection, String)** - Method in interface javax.xml.registry.**QueryManager**
- Gets the specified RegistryObjects.

**getRegistryObjects()** - Method in interface javax.xml.registry.**QueryManager**
- Gets the RegistryObjects owned by the caller.

**getRegistryObjects(String)** - Method in interface javax.xml.registry.**QueryManager**
- Gets the RegistryObjects owned by the caller, that are of the specified type.

**getRegistryPackages()** - Method in interface javax.xml.registry.infomodel.**RegistryObject**
- Returns the Package associated with this object.

**getRegistryService()** - Method in interface javax.xml.registry.**Connection**
- Gets the RegistryService interface associated with the Connection.

**getRegistryService()** - Method in interface javax.xml.registry.**LifeCycleManager**
- Returns the parent RegistryService that created this object.

**getRegistryService()** - Method in interface javax.xml.registry.**QueryManager**
- Returns the parent RegistryService that created this object.

**getRel()** - Method in class javax.faces.component.html.**HtmlCommandLink**
- Return the value of the rel property.

**getRel()** - Method in class javax.faces.component.html.**HtmlOutputLink**
- Return the value of the rel property.

**getRelay()** - Method in interface javax.xml.soap.**SOAPHeaderElement**
- Returns the boolean value of the relay attribute for this SOAPHeaderElement

**getReliableURN()** - Method in class javax.servlet.jsp.tagext.**TagLibraryInfo**
- The "reliable" URN indicated in the TLD (the uri element).

**getRemainder()** - Method in class javax.mail.internet.**HeaderTokenizer**
- Return the rest of the Header.

**getRemaining()** - Method in class javax.servlet.jsp.**JspWriter**
This method returns the number of unused bytes in the buffer.

**getRemoteAddr()** - Method in interface `javax.servlet.ServletRequest` Returns the Internet Protocol (IP) address of the client or last proxy that sent the request.

**getRemoteAddr()** - Method in class `javax.servlet.ServletRequestWrapper` The default behavior of this method is to return getRemoteAddr() on the wrapped request object.

**getRemoteHost()** - Method in interface `javax.servlet.ServletRequest` Returns the fully qualified name of the client or the last proxy that sent the request.

**getRemoteHost()** - Method in class `javax.servlet.ServletRequestWrapper` The default behavior of this method is to return getRemoteHost() on the wrapped request object.

**getRemoteInterfaceClass()** - Method in interface `javax.ejb.EJBMetaData` Obtain the Class object for the enterprise Bean's remote interface.

**getRemotePort()** - Method in interface `javax.servlet.ServletRequest` Returns the Internet Protocol (IP) source port of the client or last proxy that sent the request.

**getRemotePort()** - Method in class `javax.servlet.ServletRequestWrapper` The default behavior of this method is to return getRemotePort() on the wrapped request object.

**getRemoteUser()** - Method in class `javax.faces.context.ExternalContext` Return the login name of the user making the current request if any; otherwise, return `null`.

**getRemoteUser()** - Method in interface `javax.servlet.http.HttpServletRequest` Returns the login of the user making this request, if the user has been authenticated, or `null` if the user has not been authenticated.

**getRemoteUser()** - Method in class `javax.servlet.http.HttpServletRequestWrapper` The default behavior of this method is to return getRemoteUser() on the wrapped request object.

**getRemoveCount()** - Method in interface `javax.management.j2ee.statistics.EJBStats` Number of times remove was called.

**getRenderer(FacesContext)** - Method in class...
javax.faces.component.*.UIComponent
Convenience method to return the Renderer instance associated with this component, if any; otherwise, return null.

**getRenderer(FacesContext)** - Method in class
javax.faces.component.*.UIComponentBase

**getRenderer(String, String)** - Method in class
javax.faces.render.*.RenderKit
Return the Renderer instance most recently registered for the specified component family and rendererType, if any; otherwise, return null.

**getRendererType()** - Method in class
javax.faces.component.*.UIComponent
Return the Renderer type for this UIComponent (if any).

**getRendererType()** - Method in class
javax.faces.component.*.UIComponentBase

**getRendererType()** - Method in class
javax.faces.webapp.*.UIComponentTagBase
Return the rendererType property that selects the Renderer to be used for encoding this component, or null to ask the component to render itself directly.

**getRenderKit()** - Method in class
javax.faces.context.*.FacesContext
Return the RenderKit instance for the render kit identifier specified on our UIViewRoot, if there is one.

**getRenderKit(FacesContext, String)** - Method in class
javax.faces.render.*.RenderKitFactory
Return a RenderKit instance for the specified render kit identifier, possibly customized based on dynamic characteristics of the specified FacesContext, if non-null.

**getRenderKitId()** - Method in class
javax.faces.component.*.UIViewRoot
Return the render kit identifier of the RenderKit associated with this view.

**getRenderKitIds()** - Method in class
javax.faces.render.*.RenderKitFactory
Return an Iterator over the set of render kit identifiers registered with this factory.

**getRenderResponse()** - Method in class
javax.faces.context.*.FacesContext
Return true if the `renderResponse()` method has been called for the current request.

**getRendersChildren()** - Method in class `javax.faces.component.UIComponent`

Return a flag indicating whether this component is responsible for rendering its child components.

**getRendersChildren()** - Method in class `javax.faces.component.UIComponentBase`

**getRendersChildren()** - Method in class `javax.faces.render.Renderer`

Return a flag indicating whether this `Renderer` is responsible for rendering the children the component it is asked to render.

**getReplacementText()** - Method in interface `javax.xml.stream.events.EntityDeclaration`

The replacement text of the entity.

**getReplyTo()** - Method in class `javax.mail.internet.MimeMessage`

Return the value of the RFC 822 "Reply-To" header field.

**getReplyTo()** - Method in class `javax.mail.Message`

Get the addresses to which replies should be directed.

**getRepositoryItem()** - Method in interface `javax.xml.registry.infomodel.ExtrinsicObject`

Gets the repository item for this object.

**getRepresentationClass()** - Method in class `javax.activation.ActivationDataFlavor`

Return the representation class.

**getRequest()** - Method in class `javax.faces.context.ExternalContext`

Return the environment-specific object instance for the current request.

**getRequest()** - Method in class `javax.servlet.jsp.PageContext`

The current value of the request object (a ServletRequest).

**getRequest()** - Method in class `javax.servlet.ServletRequestWrapper`

Return the wrapped request object.

**getRequestCharacterEncoding()** - Method in class `javax.faces.context.ExternalContext`

Return the character encoding currently being used to interpret this request.

**getRequestContentType()** - Method in class `javax.faces.context.ExternalContext`

Return the MIME Content-Type for this request.
**getRequestContext()** - Method in interface javax.xml.ws.BindingProvider
Get the context that is used to initialize the message context for request messages.

**getRequestContextPath()** - Method in class javax.faces.context.ExternalContext
Return the portion of the request URI that identifies the web application context for this request.

**getRequestCookieMap()** - Method in class javax.faces.context.ExternalContext
Return an immutable map whose keys are the set of cookie names included in the current request, and whose values (of type javax.servlet.http.Cookie) are the first (or only) cookie for each cookie name returned by the underlying request.

**getRequestDispatcher(String)** - Method in interface javax.servlet.ServletContext
Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.

**getRequestDispatcher(String)** - Method in interface javax.servletServletRequest
Returns a RequestDispatcher object that acts as a wrapper for the resource located at the given path.

**getRequestDispatcher(String)** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return getRequestDispatcher(String path) on the wrapped request object.

**getRequestedSessionId()** - Method in interface javax.servlet.HttpServletRequest
Returns the session ID specified by the client.

**getRequestedSessionId()** - Method in class javax.servlet.HttpServletRequestWrapper
The default behavior of this method is to return getRequestedSessionId() on the wrapped request object.

**getRequestHeaderMap()** - Method in class javax.faces.context.ExternalContext
Return an immutable map whose keys are the set of request header names included in the current request, and whose values (of type String) are the first (or only) value for each header name returned by the underlying request.

**getRequestHeaderValueMap()** - Method in class
javax.faces.context.ExternalContext
Return an immutable Map whose keys are the set of request header names included in the current request, and whose values (of type String[]) are all of the value for each header name returned by the underlying request.

getRequestId() - Method in exception javax.xml.registry.JAXRException
getRequestId() - Method in interface javax.xml.registry.JAXRResponse
Returns the unique id for the request that generated this response.

getRequestingPort() - Method in class javax.mail.Authenticator

getRequestingPrompt() - Method in class javax.mail.Authenticator

getRequestingProtocol() - Method in class javax.mail.Authenticator
Give the protocol that’s requesting the connection.

getRequestingSite() - Method in class javax.mail.Authenticator

getRequestLocale() - Method in class javax.faces.context.ExternalContext
Return the preferred Locale in which the client will accept content.

getRequestLocales() - Method in class javax.faces.context.ExternalContext
Return an Iterator over the preferred Locales specified in the request, in decreasing order of preference.

getRequestMap() - Method in class javax.faces.context.ExternalContext
Return a mutable Map representing the request scope attributes for the current application.

getRequestParameterMap() - Method in class javax.faces.context.ExternalContext
Return an immutable Map whose keys are the set of request parameters names included in the current request, and whose values (of type String) are the first (or only) value for each parameter name returned by the underlying request.

getRequestParameterNames() - Method in class javax.faces.context.ExternalContext
Return an Iterator over the names of all request parameters included in the current request.

getRequestParameterValuesMap() - Method in class javax.faces.context.ExternalContext
Return an immutable map whose keys are the set of request parameters names included in the current request, and whose values (of type String[]) are all of the values for each parameter name returned by the underlying request.

getRequestParamInfo() - Method in class javax.faces.context.ExternalContext
Return the extra path information (if any) included in the request URI; otherwise, return null.

getRequestParamServletPath() - Method in class javax.faces.context.ExternalContext
Return the servlet path information (if any) included in the request URI; otherwise, return null.

getRequestParamURI() - Method in class javax.servlet.http.HttpServletRequest
Returns the part of this request's URL from the protocol name up to the query string in the first line of the HTTP request.

getRequestParamURI() - Method in class javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return getRequestParamURI() on the wrapped request object.

getRequestParamURI() - Method in class javax.servlet.jsp>ErrorData
Returns the request URI.

Reconstructs the URL the client used to make the request.

getRequestParamURL() - Method in class javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return getRequestParamURL() on the wrapped request object.

getRequestParamURL(HttpServletRequest) - Static method in class javax.servlet.http.HttpUtil
Deprecated. Reconstructs the URL the client used to make the request, using information in the HttpServletRequest object.

getRequiredMessage() - Method in class javax.faces.component.UIInput
If there has been a call to UIInput.setRequiredMessage(java.lang.String) on this instance, return the message.

getRequiredVersion() - Method in class
javax.servlet.jsp.tagext.\TagLibraryInfo

A string describing the required version of the JSP container.

**getResource(String)** - Method in class
javax.faces.context.\ExternalContext

Return a URL for the application resource mapped to the specified path, if it exists; otherwise, return null.

**getResource(String)** - Method in interface javax.servlet.\ServletContext

Returns a URL to the resource that is mapped to a specified path.

**getResource(Object)** - Method in interface
javax.transaction.\TransactionSynchronizationRegistry

Get an object from the Map of resources being managed for the transaction bound to the current thread at the time this method is called.

**getResourceAdapter()** - Method in interface
javax.resource.spl.\ResourceAdapterAssociation

Get the associated ResourceAdapter object.

**getResourceAsStream(String)** - Method in class
javax.faces.context.\ExternalContext

Return an InputStream for an application resource mapped to the specified path, if it exists; otherwise, return null.

**getResourceAsStream(String)** - Method in interface
javax.servlet.\ServletContext

Returns a directory-like listing of all the paths to resources within the web application whose longest sub-path matches the supplied path argument.

**getResourceURL(FacesContext, String)** - Method in class
javax.faces.application.\ViewHandler
Return a URL suitable for rendering (after optional encoding perfomed by the encodeResourceURL() method of ExternalContext) that selects the specified web application resource.

**getResourceURL(FacesContext, String)** - Method in class javax.faces.application.ViewHandlerWrapper
The default behavior of this method is to call ViewHandler.getResourceURL(javax.faces.context.FacesContext, String) on the wrapped ViewHandler object.

**getResponse()** - Method in class javax.faces.context.ExternalContext
Return the environment-specific object instance for the current response.

**getResponse()** - Method in class javax.servlet.jsp.PageContext
The current value of the response object (a ServletResponse).

**getResponse()** - Method in class javax.servlet.ServletResponseWrapper
Return the wrapped ServletResponse object.

**getResponseCharacterEncoding()** - Method in class javax.faces.context.ExternalContext
Returns the name of the character encoding (MIME charset) used for the body sent in this response.

**getResponseComplete()** - Method in class javax.faces.context.FacesContext
Return true if the responseComplete() method has been called for the current request.

**getResponseContentType()** - Method in class javax.faces.context.ExternalContext
Return the MIME Content-Type for this response.

**getResponseContext()** - Method in interface javax.xml.ws.BindingProvider
Get the context that resulted from processing a response message.

**getResponseStateManager()** - Method in class javax.faces.render.RenderKit
Return an instance of ResponseStateManager to handle rendering technology specific state management decisions.

**getResponseStream()** - Method in class javax.faces.context.FacesContext
Return the ResponseStream to which components should direct their binary output.

**getResponseWriter()** - Method in class javax.faces.context.FacesContext
Return the `ResponseWriter` to which components should direct their character-based output.

`getResult()` - Method in interface `javax.xml.bind.UnmarshallerHandler`
Obtains the unmarshalled result.

`getResult()` - Method in class `javax.xml.bind.util.JAXBResult`
Gets the unmarshalled object created by the transformation.

`getResult()` - Method in class `javax.xml.soap.SAAJResult`

`getResultList()` - Method in interface `javax.persistence.Query`
Execute a SELECT query and return the query results as a List.

`getResultSetInfo()` - Method in interface `javax.resource.cci.Connection`
Gets the information on the ResultSet functionality supported by a connected EIS instance.

`getResultTargetModuleIDs()` - Method in interface `javax.enterprise.deploy.spi.status.ProgressObject`
Retrieve the list of TargetModuleIDs successfully processed or created by the associated DeploymentManager operation.

`getReturnType()` - Method in class `javax.el.MethodInfo`
Returns the return type of the method

`getReturnType()` - Method in interface `javax.xml.rpc.Call`
Gets the return type for a specific operation

`getRev()` - Method in class `javax.faces.component.html.HtmlCommandLink`
Return the value of the `rev` property.

`getRev()` - Method in class `javax.faces.component.html.HtmlOutputLink`
Return the value of the `rev` property.

`getRole()` - Method in interface `javax.xml.soap.SOAPHeaderElement`
Returns the value of the `Role` attribute of this `SOAPHeaderElement`.

`getRoles()` - Method in interface `javax.xml.rpc.handler.HandlerChain`
Gets SOAP actor roles registered for this HandlerChain at this SOAP node.

`getRoles()` - Method in interface `javax.xml.ws.handler.soap.SOAPMessageContext`
Gets the SOAP actor roles associated with an execution of the HandlerChain and its contained Handler instances.

`getRoles()` - Method in interface `javax.xml.ws.handler.soap.SOAPMessageContext`
Gets the SOAP actor roles associated with an execution of the handler chain.
**getRoles()** - Method in interface `javax.xml.ws.soap.SOAPBinding`
  Gets the roles played by the SOAP binding instance.

**getRollbackOnly()** - Method in interface `javax.ejb.EJBContext`
  Test if the transaction has been marked for rollback only.

**getRollbackOnly()** - Method in interface `javax.persistence.EntityTransaction`
  Determine whether the current transaction has been marked for rollback.

**getRollbackOnly()** - Method in interface `javax.transaction.TransactionSynchronizationRegistry`
  Get the rollbackOnly status of the transaction bound to the current thread at the time this method is called.

**getRolledbackCount()** - Method in interface `javax.management.j2ee.statistics.JTASstats`
  Number of rolled-back transactions.

**getRoot()** - Method in interface `javax.enterprise.deploy.model.DDBean`
  Return the root element for this DDBean.

**getRootCause()** - Method in exception `javax.servlet.jsp.el.ELException`
  Deprecated. Returns the root cause.

**getRootCause()** - Method in exception `javax.servlet.jsp.JspException`
  Deprecated. As of JSP 2.1, replaced by `Throwable.getCause()`

**getRootCause()** - Method in exception `javax.servlet.ServletException`
  Returns the exception that caused this servlet exception.

**getRootOrganization()** - Method in interface `javax.xml.registry.infomodel.Organization`
  Gets the root organization.

**getRowClasses()** - Method in class `javax.faces.component.html.HtmlDataTable`
  Return the value of the rowClasses property.

**getRowClasses()** - Method in class `javax.faces.component.html.HtmlPanelGrid`
  Return the value of the rowClasses property.

**getRowCount()** - Method in class `javax.faces.component.UIData`
  Return the number of rows in the underlying data model.

**getRowCount()** - Method in class `javax.faces.model.ArrayDataModel`
  If there is wrappedData available, return the length of the array.

**getRowCount()** - Method in class `javax.faces.model.DataModel`
  Return the number of rows of data objects represented by this DataModel.
**getRowCount()** - Method in class `javax.faces.model.ListDataModel`

If there is wrappedData available, return the length of the list.

**getRowCount()** - Method in class `javax.faces.model.ResultSetDataModel`

If there is wrappedData available, return the length of the array returned by calling `getRows()` on the underlying Result.

**getRowCount()** - Method in class `javax.faces.model.ResultSetDataModel`

Return -1, since ResultSet does not provide a standard way to determine the number of available rows without scrolling through the entire ResultSet, and this can be very expensive if the number of rows is large.

**getRowCount()** - Method in class `javax.faces.model.ScalarDataModel`

If there is wrappedData available, return 1.

**getRowData()** - Method in class `javax.faces.component.UIData`

Return the data object representing the data for the currently selected row index, if any.

**getRowData()** - Method in class `javax.faces.model.ArrayDataModel`

If row data is available, return the array element at the index specified by rowIndex.

**getRowData()** - Method in class `javax.faces.model.DataModel`

Return an object representing the data for the currently selected row index.

**getRowData()** - Method in class `javax.faces.model.DataModelEvent`

Return the object representing the data for the specified row index, or null for no associated row data.

**getRowData()** - Method in class `javax.faces.model.ListDataModel`

If row data is available, return the array element at the index specified by rowIndex.

**getRowData()** - Method in class `javax.faces.model.ResultSetDataModel`

If row data is available, return the `SortedMap` array element at the index specified by rowIndex of the array returned by calling `getRows()` on the underlying Result.

**getRowData()** - Method in class `javax.faces.model.ResultSetDataModel`

If row data is available, return a `Map` representing the values of the columns for the row specified by rowIndex, keyed by the corresponding column names.

**getRowData()** - Method in class `javax.faces.model.ScalarDataModel`

If wrapped data is available, return the wrapped data instance.

**getRowIndex()** - Method in class `javax.faces.component.UIData`
Return the zero-relative index of the currently selected row.

**getRowIndex()** - Method in class `javax.faces.model.ArrayDataModel`

**getRowIndex()** - Method in class `javax.faces.model.DataModel`
Return the zero-relative index of the currently selected row.

**getRowIndex()** - Method in class `javax.faces.model.DataModelEvent`
Return the row index for this event, or -1 for no specific row.

**getRowIndex()** - Method in class `javax.faces.model.ListDataModel`

**getRowIndex()** - Method in class `javax.faces.model.ResultSetDataModel`

**getRowIndex()** - Method in class `javax.faces.model.ScalarDataModel`

**getRows()** - Method in class `javax.faces.component.html.HtmlInputTextarea`
Return the value of the `rows` property.

**getRows()** - Method in class `javax.faces.component.UIViewData`
Return the number of rows to be displayed, or zero for all remaining rows in the table.

**getRules()** - Method in class `javax.faces.component.html.HtmlDataTable`
Return the value of the `rules` property.

**getRules()** - Method in class `javax.faces.component.html.HtmlPanelGrid`
Return the value of the `rules` property.

**getRunningModules(ModuleType, Target[])** - Method in interface `javax.enterprise.deploy.spi.DeploymentManager`
Retrieve the list of J2EE application modules distributed to the identified targets and that are currently running on the associated server or servers.

**getSchema()** - Method in class `javax.xml.bind.Binder`
Gets the last `Schema` object (including null) set by the `Binder.setSchema(Schema)` method.

**getSchema()** - Method in class `javax.xml.bind.helpers.AbstractMarshallerImpl`

**getSchema()** - Method in class `javax.xml.bind.helpers.AbstractUnmarshallerImpl`
**getSchema()** - Method in interface javax.xml.bind.Marshaller
Get the JAXP 1.3 **Schema** object being used to perform marshal-time validation.

**getSchema()** - Method in interface javax.xml.bind.Unmarshaller
Get the JAXP 1.3 **Schema** object being used to perform unmarshal-time validation.

**getSchemaLocation()** - Method in class javax.xml.bind.helpers.AbstractUnmarshallerImpl
Convenience method for getting the current schemaLocation.

**getSchemaType()** - Method in interface javax.xml.stream.events.XMLEvent
This method is provided for implementations to provide optional type information about the associated event.

**getScheme()** - Method in interface javax.servlet.ServletRequest
Returns the name of the scheme used to make this request, for example, http, https, or ftp.

**getScheme()** - Method in class javax.servlet.ServletRequestWrapper
The default behavior of this method is to return getScheme() on the wrapped request object.

**getScope()** - Method in class javax.servlet.jsp.tagext.TagVariableInfo
The body of the <scope> element.

**getScope()** - Method in class javax.servlet.jsp.tagext.VariableInfo
Returns the lexical scope of the variable.

**getScope()** - Method in class javax.xml.bind.JAXBElement
Returns scope of xml element declaration.

**getScope(String)** - Method in interface javax.xml.ws.handler.MessageContext
Gets the scope of a property.

**getSecure()** - Method in class javax.servlet.http.Cookie
Returns true if the browser is sending cookies only over a secure protocol, or false if the browser can send cookies using any protocol.

**getSelectedValues()** - Method in class javax.faces.component.UISelectMany
Return the currently selected values, or null if there are no currently selected values.

**getSelectItems()** - Method in class javax.faces.model.SelectItemGroup
Return the set of subordinate SelectItems for this group.

**getSender()** - Method in class javax.mail.internet.MimeMessage
Returns the value of the RFC 822 "Sender" header field.

**getSentDate()** - Method in class javax.mail.internet.MimeMessage
Returns the value of the RFC 822 "Date" field.

**getSentDate()** - Method in class javax.mail.Message
Get the date this message was sent.

**getSentMailCount()** - Method in interface javax.management.j2ee.statistics.JavaMailStats
The number of mail messages sent.

**getSeparator()** - Method in class javax.mail.Folder
Return the delimiter character that separates this Folder's pathname from the names of immediate subfolders.

**getSerializer(Class, QName)** - Method in interface javax.xml.rpc.encoding.TypeMapping
Gets the SerializerFactory registered for the specified pair of Java type and XML data type.

**getSerializerAs(String)** - Method in interface javax.xml.rpc.encoding.SerializerFactory
Returns a Serializer for the specified XML processing mechanism type.

**getServerInfo()** - Method in interface javax.servlet.ServletContext
Returns the name and version of the servlet container on which the servlet is running.

**getServerName()** - Method in interface javax.servlet.ServletRequest
Returns the host name of the server to which the request was sent.

**getServerName()** - Method in class javax.servletServletRequestWrapper
The default behavior of this method is to return getServerName() on the wrapped request object.

**getServerPort()** - Method in interface javax.servlet.ServletRequest
Returns the port number to which the request was sent.

**getServerPort()** - Method in class javax.servletServletRequestWrapper
The default behavior of this method is to return getServerPort() on the wrapped request object.

**getServerSession()** - Method in interface javax.jms.ServerSessionPool
Return a server session from the pool.

**getServerSessionPool()** - Method in interface javax.jms.ConnectionConsumer
Gets the server session pool associated with this connection consumer.
**getService()** - Method in interface `javax.xml.registry.infomodel.ServiceBinding`  
Gets the parent service for which this is a binding.

**getServiceBinding()** - Method in interface `javax.xml.registry.infomodel.SpecificationLink`  
Gets the parent ServiceBinding for this SpecificationLink.

**getServiceBindings()** - Method in interface `javax.xml.registry.infomodel.Service`  
Gets all children ServiceBindings.

**getServiceName()** - Method in interface `javax.xml.rpc.Service`  
Gets the name of this service.

**getServiceName()** - Method in interface `javax.xml.ws.handler.PortInfo`  
Gets the qualified name of the WSDL service name containing the port being accessed.

**getServiceName()** - Method in class `javax.xml.ws.Service`  
Gets the name of this service.

**getServiceName()** - Method in class `javax.xml.ws.spi.ServiceDelegate`  
Gets the name of this service.

**getServices()** - Method in interface `javax.xml.registry.infomodel.Organization`  
Gets all children Services.

**getServiceTime()** - Method in interface `javax.management.j2ee.statistics.ServletStats`  
Execution times for the methods.

**getServlet(String)** - Method in interface `javax.servlet.ServletContext`  
*Deprecated.* As of Java Servlet API 2.1, with no direct replacement.

This method was originally defined to retrieve a servlet from a ServletContext. In this version, this method always returns `null` and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.

In lieu of this method, servlets can share information using the ServletContext class and can perform shared business logic by invoking methods on common non-servlet classes.

**getServlet()** - Method in exception `javax.servlet.UnavailableException`  
*Deprecated.* As of Java Servlet API 2.2, with no replacement.  
*Returns the servlet that is reporting its unavailability.*
**getServletConfig()** - Method in class `javax.faces.webapp.FacesServlet`
  Return the ServletConfig instance for this servlet.

**getServletConfig()** - Method in class `javax.servlet.GenericServlet`
  Returns this servlet's `ServletConfig` object.

**getServletConfig()** - Method in class `javax.servlet.jsp.PageContext`
  The ServletConfig instance.

**getServletConfig()** - Method in interface `javax.servlet.Servlet`
  Returns a `ServletConfig` object, which contains initialization and startup parameters for this servlet.

**getServletContext()** - Method in interface `javax.servlet.FilterConfig`
  Returns a reference to the `ServletContext` in which the caller is executing.

**getServletContext()** - Method in class `javax.servlet.GenericServlet`
  Returns a reference to the `ServletContext` in which this servlet is running.

  Returns the ServletContext to which this session belongs.

**getServletContext()** - Method in class `javax.servlet.jsp.PageContext`
  The ServletContext instance.

**getServletContext()** - Method in interface `javax.servlet.ServletConfig`
  Returns a reference to the `ServletContext` in which this Servlet is running.

**getServletContext()** - Method in class `javax.servlet.ServletContextEvent`
  Return the ServletContext that changed.

**getServletContext()** - Method in class `javax.servlet.ServletRequestEvent`
  Returns the ServletContext of this web application.

**getServletContext()** - Method in interface `javax.xml.rpc.server.ServletEndpointContext`
  The method `getServletContext` returns the ServletContext associated with the web application that contain this endpoint.

**getServletContextName()** - Method in interface `javax.servlet.ServletContext`
  Returns the name of this web application corresponding to this ServletContext as specified in the deployment descriptor for this web application by the display-name element.

**getServletInfo()** - Method in class `javax.faces.webapp.FacesServlet`
  Return information about this Servlet.

**getServletInfo()** - Method in class `javax.servlet.GenericServlet`
Returns information about the servlet, such as author, version, and copyright.

**getServletInfo()** - Method in interface javax.servlet.Servlet

Returns information about the servlet, such as author, version, and copyright.

**getServletName()** - Method in class javax.servlet.GenericServlet

Returns the name of this servlet instance.

**getServletName()** - Method in class javax.servlet.jsp.ErrorData

Returns the name of the servlet invoked.

**getServletName()** - Method in interface javax.servlet.ServletConfig

Returns the name of this servlet instance.

**getServletNames()** - Method in interface javax.servlet.ServletContext

Deprecated. As of Java Servlet API 2.1, with no replacement.

This method was originally defined to return an Enumeration of all the servlet names known to this context. In this version, this method always returns an empty Enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.


Returns the part of this request's URL that calls the servlet.

**getServletPath()** - Method in class javax.servlet.http.HttpServletRequestWrapper

The default behavior of this method is to return getServletPath() on the wrapped request object.

**getServletRequest()** - Method in class javax.servlet.ServletRequestEvent

Returns the ServletRequest that is changing.

**getServlets()** - Method in interface javax.servlet.ServletContext

Deprecated. As of Java Servlet API 2.0, with no replacement.

This method was originally defined to return an Enumeration of all the servlets known to this servlet context. In this version, this method always returns an empty enumeration and remains only to preserve binary compatibility. This method will be permanently removed in a future version of the Java Servlet API.
**getSession(boolean)** - Method in class `javax.faces.context.ExternalContext`
   If the `create` parameter is true, create (if necessary) and return a session instance associated with the current request.

**getSession()** - Method in interface `javax.jms.ServerSession`
   Return the ServerSession's Session.

**getSession()** - Method in interface `javax.jms.XASession`
   Gets the session associated with this XASession.

**getSession()** - Method in class `javax.mail.MessageContext`
   Return the Session we're operating in.

**getSession(boolean)** - Method in interface `javax.servlet.http.HttpServletRequest`
   Returns the current HttpSession associated with this request or, if there is no current session and `create` is true, returns a new session.

**getSession()** - Method in interface `javax.servlet.http.HttpServletRequest`
   Returns the current session associated with this request, or if the request does not have a session, creates one.

**getSession(boolean)** - Method in class `javax.servlet.http.HttpServletRequestWrapper`
   The default behavior of this method is to return `getSession(boolean create)` on the wrapped request object.

**getSession()** - Method in class `javax.servlet.http.HttpServletRequestWrapper`
   The default behavior of this method is to return `getSession()` on the wrapped request object.

**getSession()** - Method in class `javax.servlet.http.HttpServletResponseWrapper`
   Return the session that changed.

**getSession(String)** - Method in interface `javax.servlet.http.HttpServletResponseContext`
   Deprecated. As of Java Servlet API 2.1 with no replacement. This method must return null and will be removed in a future version of this API.

**getSession()** - Method in class `javax.servlet.http.HttpServletResponse`
   Return the session that changed.

**getSession()** - Method in class `javax.servlet.jsp.PageContext`
   The current value of the session object (an HttpSession).

**Deprecated.** As of Version 2.1, this method is deprecated and has no replacement. It will be removed in a future version of the Java Servlet API.

**getSessions()** - Method in interface javax.management.j2ee.statistics.JMSConnectionStats

Returns an array of JMSSessionStats that provide statistics about the sessions associated with the referencing JMSConnectionStats.

**getSeverity()** - Method in class javax.faces.application.FacesMessage

Return the severity level.

**getSeverity()** - Method in class javax.xml.bind.helpers.ValidationEventImpl

Retrieve the severity code for this warning/error.

**getShape()** - Method in class javax.faces.component.html.HtmlCommandLink

Return the value of the shape property.

**getShape()** - Method in class javax.faces.component.html.HtmlOutputLink

Return the value of the shape property.

**getSharedNamespaces()** - Method in class javax.mail.Store

Return a set of folders representing the shared namespaces.

**getShort(String)** - Method in interface javax.jms.MapMessage

Returns the short value with the specified name.

**getShortName()** - Method in class javax.servlet.jsp.tagext.TagLibraryInfo

The preferred short name (prefix) as indicated in the TLD.

**getShortProperty(String)** - Method in interface javax.jms.Message

Returns the value of the short property with the specified name.

**getSingleResult()** - Method in interface javax.persistence.Query

Execute a SELECT query that returns a single result.

**getSize()** - Method in class javax.faces.component.html.HtmlInputSecret

Return the value of the size property.

**getSize()** - Method in class javax.faces.component.html.HtmlInputText

Return the value of the size property.

**getSize()** - Method in class javax.faces.component.html.HtmlSelectManyListbox
Return the value of the size property.

**getSize()** - Method in class javax.faces.component.html.HtmlSelectOneListbox

Return the value of the size property.

**getSize()** - Method in class javax.mail.internet.MimeBodyPart

Return the size of the content of this body part in bytes.

**getSize()** - Method in class javax.mail.internet.MimeMessage

Return the size of the content of this message in bytes.

**getSize()** - Method in interface javax.mail.Part

Return the size of the content of this part in bytes.

**getSize()** - Method in class javax.xml.soap.AttachmentPart

Returns the number of bytes in this AttachmentPart object.

**getSlot(String)** - Method in interface javax.xml.registry.infomodel.ExtensibleObject

Gets the slot specified by slotName.

**getSlots()** - Method in interface javax.xml.registry.infomodel.ExtensibleObject

Returns the Slots associated with this object.

**getSlotType()** - Method in interface javax.xml.registry.infomodel.Slot

Gets the slotType for this Slot.

**getSmallIcon()** - Method in class javax.servlet.jsp.tagext.TagInfo

Get the path to the small icon.

**getSOAPBody()** - Method in class javax.xml.soap.SOAPMessage

Gets the SOAP Body contained in this SOAPMessage object.

**getSOAPFactory()** - Method in interface javax.xml.ws.soap.SOAPBinding

Gets the SAAJ SOAPFactory instance used by this SOAP binding.

**getSOAPHeader()** - Method in class javax.xml.soap.SOAPMessage

Gets the SOAP Header contained in this SOAPMessage object.

**getSOAPPart()** - Method in class javax.xml.soap.SOAPMessage

Gets the SOAP part of this SOAPMessage object.

**getSourceObject()** - Method in interface javax.xml.registry.infomodel.Association

Gets the Object that is the source of this Association.

**getSpecificationLinks()** - Method in interface javax.xml.registry.infomodel.ServiceBinding

Gets all children SpecificationLinks.

**getSpecificationObject()** - Method in interface javax.xml.registry.infomodelSpecificationLink
Gets the specification object for this object.

**getSpecificationVersion()** - Method in class
javax.servlet.jsp.JspEngineInfo
Return the version number of the JSP specification that is supported
by this JSP engine.

**getSpecVersion()** - Method in interface
javax.resource.cci.ResourceAdapterMetaData
Returns a string representation of the version of the connector
architecture specification that is supported by the resource adapter.

**getStability()** - Method in interface
javax.xml.registry.infomodel.RegistryEntry
Gets the stability indicator for the RegistryEntry within the Registry.

**getStartDuration()** - Method in class
javax.resource.spi.work.WorkEvent
Return the start interval duration.

**getStartTime()** - Method in interface
javax.management.j2ee.statistics.Statistic
The time of the first measurement represented as a long, whose
value is the number of milliseconds since January 1, 1970, 00:00:00.

**getState()** - Method in interface
javax.enterprise.deploy.spi.status.DeploymentStatus
Retrieve the StateType value.

**getState()** - Method in class
javax.faces.application.StateManager.SerializedView
Deprecated.

**getState(FacesContext, String)** - Method in class
javax.faces.render.ResponseStateManager
The implementation must inspect the current request and return an
Object representing the tree structure and component state passed
in to a previous invocation of
ResponseStateManager.writeState(javax.faces.context.FacesContext,java.lang.Object)

**getStateManager()** - Method in class
javax.faces.application.Application
Return the StateManager instance that will be utilized during the
Restore View and Render Response phases of the request
processing lifecycle.

**getStateOrProvince()** - Method in interface
javax.xml.registry.infomodel.PostalAddress
Returns the state or province.

**getStateType(int)** - Static method in class
javax.enterprise.deploy.shared.StateType
Return an object of the specified value.

**getStatistic(String)** - Method in interface javax.management.j2ee.statistics.Stats
Get a Statistic by name.

**getStatisticNames()** - Method in interface javax.management.j2ee.statistics.Stats
Returns an array of Strings which are the names of the attributes from the specific Stats submodel that this object supports.

**getStatistics()** - Method in interface javax.management.j2ee.statistics.Stats
Returns an array containing all of the Statistic objects supported by this Stats object.

**getStatus()** - Method in interface javax.transaction.Transaction
Obtain the status of the transaction associated with the target Transaction object.

**getStatus()** - Method in interface javax.transaction.TransactionManager
Obtain the status of the transaction associated with the current thread.

**getStatus()** - Method in interface javax.transaction.UserTransaction
Obtain the status of the transaction associated with the current thread.

**getStatus()** - Method in interface javax.xml.registry.infomodel.RegistryEntry
Gets the life cycle status of the RegistryEntry within the registry.

**getStatus()** - Method in exception javax.xml.registry.JAXRException
Returns the status for this response.

**getStatusCode()** - Method in class javax.servlet.jsp>ErrorData
Returns the status code of the error.

**getStatusCode()** - Method in exception javax.xml.ws.http.HTTPException
Gets the HTTP status code.

**getStore()** - Method in class javax.mail.Folder
Returns the Store that owns this Folder object.

**getStore()** - Method in class javax.mail.Session
Get a Store object that implements this user's desired Store protocol.

**getStore(String)** - Method in class javax.mail.Session
Get a Store object that implements the specified protocol.
**getStore(URLName)** - Method in class `javax.mail.Session`
Get a Store object for the given URLName.

**getStore(Provider)** - Method in class `javax.mail.Session`
Get an instance of the store specified by Provider.

**getStore()** - Method in exception `javax.mail.StoreClosedException`
Returns the dead Store object

**getStreet()** - Method in interface `javax.xml.registry.infomodel.PostalAddress`
Returns the street name.

**getStreetNumber()** - Method in interface `javax.xml.registry.infomodel.PostalAddress`
Returns the street number.

**getString(String)** - Method in interface `javax.jms.MapMessage`
Returns the String value with the specified name.

**getString()** - Method in class `javax.servlet.jsp.tagext.BodyContent`
Return the value of the BodyContent as a String.

**getStringProperty(String)** - Method in interface `javax.jms.Message`
Returns the value of the String property with the specified name.

**getStringTable()** - Method in class `javax.enterprise.deploy.shared.ActionType`
Returns the string table for class ActionType

**getStringTable()** - Method in class `javax.enterprise.deploy.shared.CommandType`
Returns the string table for class CommandType

**getStringTable()** - Method in class `javax.enterprise.deploy.shared.DConfigBeanVersionType`
Returns the string table for class DConfigBeanVersionType

**getStringTable()** - Method in class `javax.enterprise.deploy.shared.ModuleType`
Returns the string table for class ModuleType

**getStringTable()** - Method in class `javax.enterprise.deploy.shared.StateType`
Returns the string table for class StateType

**getStructure()** - Method in class `javax.faces.application.StateManager.SerializedView`
Deprecated.

**getStyle()** - Method in class `javax.faces.component.html.HtmlCommandButton`
Return the value of the style property.
**getStyle()** - Method in class
javax.faces.component.html.HtmlCommandLink
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlDataTable
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlForm
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlGraphicImage
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlInputSecret
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlInputText
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlInputTextarea
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlMessage
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlMessages
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlOutputFormat
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlOutputLabel
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlOutputLink
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlOutputText
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlPanelGrid
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlPanelGroup
    Return the value of the style property.
**getStyle()** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
    Return the value of the style property.
**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`

Return the value of the style property.

**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`

Return the value of the style property.

**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectManyMenu`

Return the value of the style property.

**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`

Return the value of the style property.

**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectOneMenu`

Return the value of the style property.

**getStyle()** - Method in class `javax.faces.component.html.HtmlSelectOneRadio`

Return the value of the style property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlCommandButton`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlCommandLink`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlDataTable`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlForm`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlGraphicImage`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlInputSecret`

Return the value of the styleClass property.

**getStyleClass()** - Method in class `javax.faces.component.html.HtmlInputText`

Return the value of the styleClass property.
javax.faces.component.html.HtmlInputTextarea
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlMessage
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlMessages
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlOutputFormat
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlOutputLabel
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlOutputLink
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlOutputText
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlPanelGroup
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Return the value of the styleClass property.

getStyleClass() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Return the value of the styleClass property.

**getStyleClass()** - Method in class

javax.faces.component.html.HtmlSelectOneMenu
Return the value of the styleClass property.

**getStyleClass()** - Method in class

javax.faces.component.html.HtmlSelectOneRadio
Return the value of the styleClass property.

**getSubject()** - Method in class javax.mail.internet.MimeMessage
Returns the value of the "Subject" header field.

**getSubject()** - Method in class javax.mail.Message
Get the subject of this message.

**getSubmittedValue()** - Method in interface
javax.faces.component.EditableValueHolder
Return the submittedValue value of this component.

**getSubmittedValue()** - Method in class javax.faces.component.UIInput
Return the submittedValue value of this UIInput component.

**getSubmittingOrganization()** - Method in interface
javax.xml.registry.infomodel.RegistryObject
Gets the Organization that submitted this RegistryObject.

**getSubType()** - Method in class javax.activation.MimeType
Retrieve the subtype of this object.

**getSubType()** - Method in class javax.mail.internet.ContentType
Return the subType.

**getSummary()** - Method in class javax.faces.application.FacesMessage
Return the localized summary text.

**getSummary()** - Method in class
javax.faces.component.html.HtmlDataTable
Return the value of the summary property.

**getSummary()** - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the summary property.

**getSupportedEncodings()** - Method in interface
javax.xml.rpc.encoding.TypeMapping
Returns the encodingStyle URIs (as String[]) supported by this TypeMapping instance.

**getSupportedLocales()** - Method in interface
javax.enterprise.deploy.spi.DeploymentManager
Returns an array of supported locales for this implementation.
**getSupportedLocales()** - Method in class javax.faces.application.Application
   Return an Iterator over the supported Locales for this application.

**getSupportedMechanismTypes()** - Method in interface javax.xml.rpc.encoding.DeserializerFactory
   Returns a list of all XML processing mechanism types supported by this DeserializerFactory.

**getSupportedMechanismTypes()** - Method in interface javax.xml.rpc.encoding.SerializerFactory
   Returns a list of all XML processing mechanism types supported by this SerializerFactory.

**getSystemFlags()** - Method in class javax.mail.Flags
   Return all the system flags in this Flags object.

**getSystemId()** - Method in interface javax.xml.stream.events.EntityDeclaration
   The entity’s system identifier.

**getSystemId()** - Method in interface javax.xml.stream.events.NotationDeclaration
   The notation's system identifier, or null if none was given.

**getSystemId()** - Method in interface javax.xml.stream.events.StartDocument
   Returns the system ID of the XML data

**getSystemId()** - Method in interface javax.xml.stream.Location
   Returns the system ID of the XML

**getTabindex()** - Method in class javax.faces.component.html.HtmlCommandButton
   Return the value of the tabindex property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlCommandLink
   Return the value of the tabindex property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlInputSecret
   Return the value of the tabindex property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlInputText
   Return the value of the tabindex property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlInputTextarea
   Return the value of the tabindex property.
**getTabindex()** - Method in class javax.faces.component.html.HtmlOutputLabel
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlOutputLink
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectManyListbox
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectManyMenu
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectOneListbox
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectOneMenu
- Return the value of the `tabindex` property.

**getTabindex()** - Method in class javax.faces.component.html.HtmlSelectOneRadio
- Return the value of the `tabindex` property.

**getTag(String)** - Method in class javax.servlet.jsp.tagext.TagLibraryInfo
- Get the TagInfo for a given tag name, looking through all the tags in this tag library.

**getTagName()** - Method in class javax.servlet.jsp.tagext.TagInfo
- Name of the class that provides the handler for this tag.

**getTagExtraInfo()** - Method in class javax.servlet.jsp.tagext.TagInfo
- The instance (if any) for extra tag information.

**getTagFile(String)** - Method in class javax.servlet.jsp.tagext.TagLibraryInfo
- Get the TagFileInfo for a given tag name, looking through all the tag files in this tag library.

**getTagFiles()** - Method in class javax.servlet.jsp.tagext.TagLibraryInfo
An array describing the tag files that are defined in this tag library.

`getTagInfo()` - Method in class `javax.servlet.jsp.tagext.TagExtraInfo`
Get the TagInfo for this class.

`getTagInfo()` - Method in class `javax.servlet.jsp.tagext.TagFileInfo`
Returns information about this tag, parsed from the directives in the tag file.

`getTagLibrary()` - Method in class `javax.servlet.jsp.tagext.TagInfo`
The instance of TagLibraryInfo we belong to.

`getTagLibraryInfos()` - Method in class `javax.servlet.jsp.tagext.TagLibraryInfo`
Returns an array of TagLibraryInfo objects representing the entire set of tag libraries (including this TagLibraryInfo) imported by taglib directives in the translation unit that references this TagLibraryInfo.

`getTagName()` - Method in class `javax.servlet.jsp.tagext.TagInfo`
The name of the Tag.

`getTags()` - Method in class `javax.servlet.jsp.tagext.TagLibraryInfo`
An array describing the tags that are defined in this tag library.

`getTagVariableInfos()` - Method in class `javax.servlet.jsp.tagext.TagInfo`
Get TagVariableInfo objects associated with this TagInfo.

`getTarget()` - Method in interface `javax.enterprise.deploy.spi.TargetModuleID`
Retrieve the name of the target server.

`getTarget()` - Method in class `javax.faces.component.html.HtmlCommandLink`
Return the value of the target property.

`getTarget()` - Method in class `javax.faces.component.html.HtmlForm`
Return the value of the target property.

`getTarget()` - Method in class `javax.faces.component.html.HtmlOutputLink`
Return the value of the target property.

`getTarget()` - Method in interface `javax.interceptor.InvocationContext`
Returns the target instance.

`getTargetBinding()` - Method in interface `javax.xml.registry.infomodel.ServiceBinding`
Gets the next ServiceBinding in case there is a redirection from one service provider to another service provider.
**getTargetEndpointAddress()** - Method in interface javax.xml.rpc.Call
- Gets the address of a target service endpoint.

**getTargetModuleID()** - Method in class javax.enterprise.deploy.spi.status.ProgressEvent
- Retrieve the TargetModuleID for this event

**getTargetObject()** - Method in interface javax.xml.registry.infomodel.Association
- Gets the Object that is the target of this Association.

**getTargets()** - Method in interface javax.enterprise.deploy.spi.DeploymentManager
- Retrieve the list of deployment targets supported by this DeploymentManager.

**getTelephoneNumbers(String)** - Method in interface javax.xml.registry.infomodel.Organization
- Gets the telephone numbers for this User that match the specified telephone number type.

**getTelephoneNumbers(String)** - Method in interface javax.xml.registry.infomodel.User
- Gets the telephone numbers for this User that match the specified telephone number type.

**getTerm()** - Method in class javax.mail.search.NotTerm
- Return the term to negate.

**getTerms()** - Method in class javax.mail.search.AndTerm
- Return the search terms.

**getTerms()** - Method in class javax.mail.search.OrTerm
- Return the search terms.

**getTestSet()** - Method in class javax.mail.search.FlagTerm
- Return true if testing whether the flags are set.

**getText()** - Method in interface javax.enterprise.deploy.model.DDBean
- Returns the XML text for by this bean.

**getText(String)** - Method in interface javax.enterprise.deploy.model.DDBean
- Return a list of text values for a given XPath in the deployment descriptor.

**getText(String)** - Method in interface javax.enterprise.deploy.model.DeployableObject
- Return the XML content associated with the XPath from a deployment descriptor.

**getText(ModuleType, String)** - Method in interface
javax.enterprise.deploy.model.\texttt{J2eeApplicationObject}

Return the text value from the XPath; search only the deployment descriptors of the specified type.

\texttt{getText()} - Method in interface \texttt{javax.jms.TextMessage}

Gets the string containing this message's data.

\texttt{getText()} - Method in interface \texttt{javax.xml.stream.events.Comment}

Return the string data of the comment, returns empty string if it does not exist

\texttt{getText()} - Method in class \texttt{javax.xml.stream.util.StreamReaderDelegate}

\texttt{getText()} - Method in interface \texttt{javax.xml.stream.XMLStreamReader}

Returns the current value of the parse event as a string, this returns the string value of a CHARACTERS event, returns the value of a COMMENT, the replacement value for an ENTITY\_REFERENCE, the string value of a CDATA section, the string value for a SPACE event, or the String value of the internal subset of the DTD.

\texttt{getTextCharacters(int, char[], int, int)} - Method in class \texttt{javax.xml.stream.util.StreamReaderDelegate}

\texttt{getTextCharacters()} - Method in class \texttt{javax.xml.stream.util.StreamReaderDelegate}

\texttt{getTextCharacters()} - Method in interface \texttt{javax.xml.stream.XMLStreamReader}

Returns an array which contains the characters from this event.

\texttt{getTextCharacters(int, char[], int, int)} - Method in interface \texttt{javax.xml.stream.XMLStreamReader}

Gets the the text associated with a CHARACTERS, SPACE or CDATA event.

\texttt{getTextLength()} - Method in class \texttt{javax.xml.stream.util.StreamReaderDelegate}

\texttt{getTextLength()} - Method in interface \texttt{javax.xml.stream.XMLStreamReader}

Returns the length of the sequence of characters for this Text event within the text character array.

\texttt{getTextStart()} - Method in class \texttt{javax.xml.stream.util.StreamReaderDelegate}
**getTextStart()** - Method in interface javax.xml.stream.XMLStreamReader
  Returns the offset into the text character array where the first character (of this text event) is stored.

**getThrowable()** - Method in class javax.servlet.jsp>ErrorData
  Returns the Throwable that caused the error.

**getTimer()** - Method in interface javax.ejb.TimerHandle
  Obtain a reference to the timer represented by this handle.

**getTimeRemaining()** - Method in interface javax.ejb.Timer
  Get the number of milliseconds that will elapse before the next scheduled timer expiration.

**getTimers()** - Method in interface javax.ejb.TimerService
  Get all the active timers associated with this bean.

**getTimerService()** - Method in interface javax.ejb.EJBContext
  Get access to the EJB Timer Service.

**getTimestamp()** - Method in interface javax.xml.registry.infomodel.AuditableEvent
  Gets the Timestamp for when this event occurred.

**getTimeStyle()** - Method in class javax.faces.convert.DateTimeConverter
  Return the style to be used to format or parse times.

**getTimeToLive()** - Method in interface javax.jms.MessageProducer
  Gets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.

**getTimeZone()** - Method in class javax.faces.convert.DateTimeConverter
  Return the TimeZone used to interpret a time value.

**getTitle()** - Method in class javax.faces.component.html.HtmlCommandButton
  Return the value of the title property.

**getTitle()** - Method in class javax.faces.component.html.HtmlCommandLink
  Return the value of the title property.

**getTitle()** - Method in class javax.faces.component.html.HtmlDataTable
  Return the value of the title property.

**getTitle()** - Method in class javax.faces.component.html.HtmlForm
  Return the value of the title property.

**getTitle()** - Method in class javax.faces.component.html.HtmlGraphicImage
  Return the value of the title property.

**getTitle()** - Method in class javax.faces.component.html.HtmlInputSecret
Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlInputText`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlInputTextarea`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlMessage`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlMessages`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlOutputFormat`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlOutputLabel`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlOutputLink`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlOutputText`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlPanelGrid`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
- Return the value of the title property.

**getTitle()** - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
- Return the value of the title property.
getTitle() - Method in class javax.faces.component.html.HtmlSelectOneRadio
  Return the value of the title property.

getTopic() - Method in interface javax.jms.TopicPublisher
  Gets the topic associated with this TopicPublisher.

getTopic() - Method in interface javax.jms.TopicSubscriber
  Gets the Topic associated with this subscriber.

getTopicName() - Method in interface javax.jms.Topic
  Gets the name of this topic.

getTopicSession() - Method in interface javax.jms.XATopicSession
  Gets the topic session associated with this XATopicSession.

g getTotalTime() - Method in interface javax.management.j2ee.statistics.TimeStatistic
  This is the sum total of time taken to complete every invocation of
  this operation since the beginning of this measurement.

getTransacted() - Method in interface javax.jms.Session
  Indicates whether the session is in transacted mode.

getTransacted() - Method in interface javax.jms.XASession
  Indicates whether the session is in transacted mode.

getTransaction() - Method in interface javax.persistence.EntityManager
  Returns the resource-level transaction object.

getTransaction() - Method in interface javax.transaction.TransactionManager
  Get the transaction object that represents the transaction context of
  the calling thread.

getTransactionKey() - Method in interface javax.transaction.TransactionSynchronizationRegistry
  Return an opaque object to represent the transaction bound to the
  current thread at the time this method is called.

gett ransactionStatus() - Method in interface javax.transaction.TransactionSynchronizationRegistry
  Return the status of the transaction bound to the current thread at
  the time this method is called.

getTransactionTimeout() - Method in class javax.resource.spi.work.ExecutionContext
  Get the transaction timeout value for a imported transaction.

getTransactionTimeout() - Method in interface javax.transaction.xa.XAResource
  Obtains the current transaction timeout value set for this
XAResource instance.

**getTransactionType()** - Method in interface javax.persistence.spi.PersistenceUnitInfo
- Returns the transaction type of the entity managers created by the EntityManagerFactory.

**getTransferData(DataFlavor, DataSource)** - Method in interface javax.activation.DataContentHandler
- Returns an object which represents the data to be transferred.

**getTransferData(DataFlavor)** - Method in class javax.activation.DataHandler
- Returns an object that represents the data to be transferred.

**getTransferDataFlavors()** - Method in interface javax.activation.DataContentHandler
- Returns an array of DataFlavor objects indicating the flavors the data can be provided in.

**getTransferDataFlavors()** - Method in class javax.activation.DataHandler
- Return the DataFlavors in which this data is available.

**getTransport()** - Method in class javax.mail.Session
- Get a Transport object that implements this user's desired Transport protocol.

**getTransport(String)** - Method in class javax.mail.Session
- Get a Transport object that implements the specified protocol.

**getTransport(URLName)** - Method in class javax.mail.Session
- Get a Transport object for the given URLName.

**getTransport(Provider)** - Method in class javax.mail.Session
- Get an instance of the transport specified in the Provider.

**getTransport(Address)** - Method in class javax.mail.Session
- Get a Transport object that can transport a Message to the specified address type.

**getTreeStructureToRestore(FacesContext, String)** - Method in class javax.faces.render.ResponseStateManager
- Deprecated. This method has been replaced by ResponseStateManager.getState(javax.faces.context.FacesContext, java.lang.String). The default implementation returns null.

**getTreeStructureToSave(FacesContext)** - Method in class javax.faces.application.StateManager
- Deprecated. the distinction between tree structure and component state is now an implementation detail. The default implementation
returns null.

**getTreeStructureToSave(FacesContext)** - Method in class javax.faces.application.StateManagerWrapper
The default behavior of this method is to call StateManager.getTreeStructureToSave(javax.faces.context.FacesContext) on the wrapped StateManager object.

**getType(ELContext, Object, Object)** - Method in class javax.el.ArrayELResolver
If the base object is an array, returns the most general acceptable type for a value in this array.

**getType(ELContext, Object, Object)** - Method in class javax.el.BeanELResolver
If the base object is not null, returns the most general acceptable type that can be set on this bean property.

**getType(ELContext, Object, Object)** - Method in class javax.el.CompositeELResolver
For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the CompositeELResolver.setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) method.

**getType(ELContext, Object, Object)** - Method in class javax.el.ELResolver
For a given base and property, attempts to identify the most general type that is acceptable for an object to be passed as the value parameter in a future call to the ELResolver.setValue(javax.el.ELContext, java.lang.Object, java.lang.Object, java.lang.Object) method.

**getType(ELContext, Object, Object)** - Method in class javax.el.MapELResolver
If the base object is a list, returns the most general acceptable type for a value in this list.

**getType(ELContext, Object, Object)** - Method in class javax.el.MapELResolver
If the base object is a map, returns the most general acceptable type for a value in this map.

**getType(ELContext, Object, Object)** - Method in class javax.el.ResourceBundleELResolver
If the base object is an instance of ResourceBundle, return null, since
the resolver is read only.

**`getType(ELContext)`** - Method in class `javax.el.ValueExpression`  
Evaluates the expression relative to the provided context, and returns the most general type that is acceptable for an object to be passed as the *value* parameter in a future call to the `ValueExpression.setValue(javax.el.ELContext, java.lang.Object)` method.

**`getType()`** - Method in interface  
`javax.enterprise.deploy.model.DDBeanRoot`  
Return the ModuleType of deployment descriptor.

**`getType()`** - Method in interface  
`javax.enterprise.deploy.model.DeployableObject`  
Return the ModuleType of deployment descriptor (i.e., EAR, JAR, WAR, RAR) this deployable object represents.

**`getType()`** - Method in class  
`javax.faces.component.html.HtmlCommandButton`  
Return the value of the *type* property.

**`getType()`** - Method in class  
`javax.faces.component.html.HtmlCommandLink`  
Return the value of the *type* property.

**`getType()`** - Method in class  
`javax.faces.component.html.HtmlOutputLink`  
Return the type of value to be formatted or parsed.

**`getType()`** - Method in class  
`javax.faces.convert.DateTimeConverter`  
Return the type to be used when formatting and parsing numbers.

**`getType(FacesContext)`** - Method in class `javax.faces.el.MethodBinding`  
Deprecated. Return the Java class representing the return type from the method identified by this method binding expression.

**`getType(Object, Object)`** - Method in class  
`javax.faces.el.PropertyResolver`  
Deprecated. Return the `java.lang.Class` representing the type of the specified property.

**`getType(Object, int)`** - Method in class `javax.faces.el.PropertyResolver`  
Deprecated. Return the `java.lang.Class` representing the type of the specified index.

**`getType(FacesContext)`** - Method in class `javax.faces.el.ValueBinding`  
Deprecated. Return the type of the property represented by this
ValueBinding, relative to the specified FacesContext.

**getType()** - Method in class javax.mail.Address
Return a type string that identifies this address type.

**getType()** - Method in class javax.mail.event.ConnectionEvent
Return the type of this event.

**getType()** - Method in class javax.mail.event.FolderEvent
Return the type of this event.

**getType()** - Method in class javax.mail.event.MessageCountEvent
Return the type of this event.

**getType()** - Method in class javax.mail.event.TransportEvent
Return the type of this event.

**getType()** - Method in class javax.mail.Folder
Returns the type of this Folder, that is, whether this folder can hold messages or subfolders or both.

**getType()** - Method in class javax.mail.internet.HeaderTokenizer.Token
Return the type of the token.

**getType()** - Method in class javax.mail.internet.InternetAddress
Return the type of this address.

**getType()** - Method in class javax.mail.internet.NewsAddress
Return the type of this address.

**getType()** - Method in class javax.mail.Provider
Returns the type of this Provider

**getType()** - Method in class javax.mail.resource.spi.work.WorkEvent
Return the type of this event.

**getType(ELContext, Object, Object)** - Method in class javax.servlet.jsp.el.ImplicitObjectELResolver
If the base object is null, and the property matches the name of a JSP implicit object, returns null to indicate that no types are ever accepted to setValue().

**getType(ELContext, Object, Object)** - Method in class javax.servlet.jsp.el ScopedAttributeELResolver
If the base object is null, returns Object.class to indicate that any type is valid to set for a scoped attribute.

**getType()** - Method in interface javax.xml.registry.infomodel.EmailAddress
Gets the type for this object.

**getType()** - Method in interface javax.xml.registry.infomodel.PostalAddress
Returns the type of address (for example, "headquarters") as a
String.

**getType()** - Method in interface javax.xml.registry.infomodel.**TelephoneNumber**
The type of telephone number (for example, "fax").

**getType()** - Method in interface javax.xml.registry.infomodel.**User**
Gets the type for this User.

**getType()** - Method in interface javax.xml.registry.infomodel.**Query**
Gets the type of Query (for example, QUERY_TYPE_SQL).

**getTypeMapping(String)** - Method in interface javax.xml.rpc.encoding.**TypeMappingRegistry**
Returns the registered TypeMapping for the specified encodingStyle URI.

**getTypeMappingRegistry()** - Method in interface javax.xml.rpc.**Service**
Gets the TypeMappingRegistry for this Service object.

**getTypeName()** - Method in class javax.servlet.jsp.tagext.**TagAttributeInfo**
The type (as a String) of this attribute.

**getUID(Message)** - Method in interface javax.mail.**UIDFolder**
Get the UID for the specified message.

**getUIDValidity()** - Method in interface javax.mail.**UIDFolder**
Returns the UIDValidity value associated with this folder.

**getUnreadMessageCount()** - Method in exception javax.mail.**Folder**
Get the total number of unread messages in this Folder.

**getUnavailableSeconds()** - Method in exception javax.servlet.**UnavailableException**
Returns the number of seconds the servlet expects to be temporarily unavailable.

**getUnit()** - Method in interface javax.management.j2ee.statistics.**Statistic**
The unit of measurement for this Statistic.

**getUnmarshallerHandler()** - Method in interface javax.xml.bind.**Unmarshaller**
Get an unmarshaller handler object that can be used as a component in an XML pipeline.

**getUnreadMessageCount()** - Method in class javax.mail.**Folder**
Get the total number of unread messages in this Folder.

**getUpperBound()** - Method in interface javax.management.j2ee.statistics.**BoundaryStatistic**
The upper limit of the value of this attribute.

**getUpTime()** - Method in interface javax.management.j2ee.statistics.**JVMStats**
Amount of time the JVM has been running.
**getURI()** - Method in class `javax.servlet.jsp.tagext.TagLibraryInfo`
The value of the uri attribute from the taglib directive for this library.

**getURI()** - Method in interface `javax.xml.soap.Name`
Returns the URI of the namespace for the XML name that this `Name` object represents.

**getURL()** - Method in class `javax.activation.URLDataSource`
Return the URL used to create this DataSource.

**getURL()** - Method in class `javax.faces.component.UIGraphic`
Return the image URL for this `UIGraphic`.

**getURL()** - Method in class `javax.mail.URLName`
Constructs a URL from the URLName.

**getURL()** - Method in class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

**getURL()** - Method in interface `javax.xml.bind.ValidationEventLocator`
Return the name of the XML source as a URL if available

**getUrl()** - Method in interface `javax.xml.registry.infomodel.TelephoneNumber`
Gets the URL that can dial this number electronically.

**getUrl()** - Method in interface `javax.xml.registry.infomodel.User`
Gets the URL to the web page for this User.

**getURLName()** - Method in class `javax.mail.Folder`
Return a URLName representing this folder.

**getURLName()** - Method in class `javax.mail.Service`
Return a URLName representing this service.

**getUsageDescription()** - Method in interface `javax.xml.registry.infomodel.SpecificationLink`
Gets the description of usage parameters.

**getUsageParameters()** - Method in interface `javax.xml.registry.infomodel.SpecificationLink`
Gets any usage parameters.

**getUsemap()** - Method in class `javax.faces.component.html.HtmlGraphicImage`
Return the value of the `usemap` property.

**getUser()** - Method in interface `javax.xml.registry.infomodel.AuditableEvent`
Gets the User associated with this object.

**getUserFlags()** - Method in class `javax.mail.Flags`
Return all the user flags in this Flags object.
**getUserName()** - Method in class `javax.mail.PasswordAuthentication`

**getUsername()** - Method in class `javax.mail.URLName`
Returns the user name of this URLName.

**getUserName()** - Method in interface `javax.resource.cci.ConnectionMetaData`
Returns the user name for an active connection as known to the underlying EIS instance.

**getUserName()** - Method in interface `javax.resource.spi.ManagedConnectionMetaData`
Returns name of the user associated with the ManagedConnection instance.

**getUserName()** - Method in class `javax.resource.spi.security.PasswordCredential`
Returns the user name.

**getUserNamespaces(String)** - Method in class `javax.mail.Store`
Return a set of folders representing the namespaces for user.

**getUserPrincipal()** - Method in class `javax.faces.context.ExternalContext`
Return the Principal object containing the name of the current authenticated user, if any; otherwise, return null.

**getUserPrincipal()** - Method in interface `javax.servlet.http.HttpServletRequest`
Returns a java.security.Principal object containing the name of the current authenticated user.

**getUserPrincipal()** - Method in class `javax.servlet.http.HttpServletRequestWrapper`
The default behavior of this method is to return getUserPrincipal() on the wrapped request object.

**getUserPrincipal()** - Method in interface `javax.xml.rpc.server.ServletEndpointContext`
Returns a java.security.Principal instance that contains the name of the authenticated user for the current method invocation on the endpoint instance.

**getUserPrincipal()** - Method in interface `javax.xml.ws.WebServiceContext`
Returns the Principal that identifies the sender of the request currently being serviced.

**getUsers()** - Method in interface
javax.xml.registry.infomodel.**Organization**
- Gets the Collection of Users affiliated with this Organization.

**getUserTransaction()** - Method in interface **javax.ejb.EJBContext**
- Obtain the transaction demarcation interface.

**getUserVersion()** - Method in interface **javax.xml.registry.infomodel.Versionable**
- Gets the user-specified revision number for this version of the Versionable object.

**getUseTime()** - Method in interface **javax.management.j2ee.statistics.JCAConnectionStats**
- Returns the time spent using a connection

**getUseTime()** - Method in interface **javax.management.j2ee.statistics.JDBCConnectionStats**
- Time spent using a connection.

**getValidateURI()** - Method in interface **javax.xml.registry.infomodel.URIValidator**
- Gets whether to do URI validation for this object.

**getValidator()** - Method in interface **javax.faces.component.EditableValueHolder**
- Deprecated. **EditableValueHolder.getValidators()** should be used instead.

**getValidator()** - Method in class **javax.faces.component.UIInput**
- Deprecated. **UIInput.getValidators()** should be used instead.

**getValidatorIds()** - Method in class **javax.faces.application.Application**
- Return an Iterator over the set of currently registered validator ids for this Application.

**getValidatorMessage()** - Method in class **javax.faces.component.UIInput**
- If there has been a call to **UIInput.setRequiredMessage(java.lang.String)** on this instance, return the message.

**getValidators()** - Method in interface **javax.faces.component.EditableValueHolder**
- Return the set of registered **Validators** for this component instance.

**getValidators()** - Method in class **javax.faces.component.UIInput**
- Return the set of registered **Validators** for this **UIInput** instance.

**getValidSentAddresses()** - Method in class **javax.mail.event.TransportEvent**
- Return the addresses to which this message was sent succesfully.

**getValidSentAddresses()** - Method in exception
javax.mail.SendFailedException
Return the addresses to which this message was sent succesfully.

getValidUnsentAddresses() - Method in class javax.mail.event.TransportEvent
Return the addresses that are valid but to which this message was not sent.

getValidUnsentAddresses() - Method in exception javax.mail.SendFailedException
Return the addresses that are valid but to which this message was not sent.

getValue(ELContext, Object, Object) - Method in class javax.el.ArrayELResolver
If the base object is a Java language array, returns the value at the given index.

getValue(ELContext, Object, Object) - Method in class javax.el.BeanELResolver
If the base object is not null, returns the current value of the given property on this bean.

getValue(ELContext, Object, Object) - Method in class javax.el.CompositeELResolver
Attempts to resolve the given property object on the given base object by querying all component resolvers.

getValue(ELContext, Object, Object) - Method in class javax.el.ELResolver
Attempts to resolve the given property object on the given base object.

getValue(ELContext, Object, Object) - Method in class javax.el.ListELResolver
If the base object is a list, returns the value at the given index.

getValue(ELContext, Object, Object) - Method in class javax.el.MapELResolver
If the base object is a map, returns the value associated with the given key, as specified by the property argument.

getValue(ELContext, Object, Object) - Method in class javax.el.ResourceBundleELResolver
If the base object is an instance of ResourceBundle, the provided property will first be coerced to a String.

getValue(ELContext) - Method in class javax.el.ValueExpression
Evaluates the expression relative to the provided context, and
returns the resulting value.

**getValue()** - Method in class `javax.enterprise.deploy.shared.ActionType`
  Returns this enumeration value's integer value.

**getValue()** - Method in class `javax.enterprise.deploy.shared.CommandType`
  Returns this enumeration value's integer value.

**getValue()** - Method in class `javax.enterprise.deploy.shared.DConfigBeanVersionType`
  Returns this enumeration value's integer value.

**getValue()** - Method in class `javax.enterprise.deploy.shared.ModuleType`
  Returns this enumeration value's integer value.

**getValue()** - Method in class `javax.enterprise.deploy.shared.StateType`
  Returns this enumeration value's integer value.

**getValue()** - Method in class `javax.faces.component.UICommand`
  Returns the `value` property of the `UICommand`.  

**getValue()** - Method in class `javax.faces.component.UIData`
  Return the value of the `UIData`.  

**getValue()** - Method in class `javax.faces.component.UIGraphic`
  Returns the `value` property of the `UIGraphic`.  

**getValue()** - Method in class `javax.faces.component.UIOutput`

**getValue()** - Method in class `javax.faces.component.UIParameter`
  Returns the `value` property of the `UIParameter`.  

**getValue()** - Method in class `javax.faces.component.UISelectItem`
  Returns the `value` property of the `UISelectItem`.  

**getValue()** - Method in class `javax.faces.component.UISelectItems`
  Returns the `value` property of the `UISelectItems`.  

**getValue(Object, Object)** - Method in class `javax.faces.el.PropertyResolver`
  Deprecated. Return the value of the specified property from the specified base object.

**getValue(Object, int)** - Method in class `javax.faces.el.PropertyResolver`
  Deprecated. Return the value at the specified index of the specified base object.

**getValue(FacesContext)** - Method in class `javax.faces.el.ValueBinding`
  Deprecated. Return the value of the property represented by this `ValueBinding`, relative to the specified `FacesContext`.  

**getValue()** - Method in class *javax.faces.model.SelectItem*
Return the value of this item, to be delivered to the model if this item is selected by the user.

**getValue()** - Method in class *javax.mail.Header*
Returns the value of this header.

**getValue()** - Method in class *javax.mail.internet.HeaderTokenizer.Token*
Returns the value of the token just read.

**getValue()** - Method in class
javax.mail.internet.InternetHeaders.InternetHeader
Return the "value" part of the header line.

**getValue()** - Method in class *javax.servlet.http.Cookie*
Returns the value of the cookie.

**getValue(String)** - Method in interface *javax.servlet.http.HttpSession*
*Deprecated. As of Version 2.2, this method is replaced by HttpSession.getAttribute(java.lang.String).*

**getValue()** - Method in class *javax.servlet.http.HttpSessionBindingEvent*
Returns the value of the attribute that has been added, removed or replaced.

**getValue(ELContext, Object, Object)** - Method in class
javax.servlet.jsp.el.ImplicitObjectELResolver
If the base object is null, and the property matches the name of a JSP implicit object, returns the implicit object.

**getValue(ELContext, Object, Object)** - Method in class
javax.servlet.jsp.el.ScopedAttributeELResolver
If the base object is null, searches the page, request, session and application scopes for an attribute with the given name and returns it, or null if no attribute exists with the current name.

**getValue(String)** - Method in class *javax.servlet.jsp.tagext.TagSupport*
Get a the value associated with a key.

**getValue()** - Method in class *javax.servlet.ServletContextAttributeEvent*
Returns the value of the attribute that has been added, removed, or replaced.

**getValue()** - Method in class *javax.servlet.ServletRequestAttributeEvent*
Returns the value of the attribute that has been added, removed or replaced.

**getValue()** - Method in class *javax.xml.bind.JAXBElement*
Return the content model and attribute values for this element.

**getValue(Object)** - Static method in class
javax.xml.bind.JAXBIntrospector
Get the element value of a JAXB element.

**getValue()** - Method in interface
javax.xml.registry.infomodel.[Classification]

  Gets the taxonomy value for this Classification.

**getValue()** - Method in interface javax.xml.registry.infomodel.[Concept]

  Gets the value (usually a code in a taxonomy) associated with this Concept.

**getValue()** - Method in interface
javax.xml.registry.infomodel.[ExternalIdentifier]

  Gets the value of an ExternalIdentifier.

**getValue()** - Method in interface
javax.xml.registry.infomodel.[InternationalString]

  Gets the String value for the Locale returned by Locale.getDefault().

**getValue(Locale)** - Method in interface
javax.xml.registry.infomodel.[InternationalString]

  Gets the String value for the specified Locale.

**getValue()** - Method in interface
javax.xml.registry.infomodel.[LocalizedString]

  Get the String value for this object.

**getValue()** - Method in class javax.xml.soap.[MimeHeader]

  Returns the value of this MimeHeader object.

**getValue()** - Method in interface javax.xml.soap.[Node]

  Returns the value of this node if this is a Text node or the value of the immediate child of this node otherwise.

**getValue()** - Method in interface javax.xml.stream.events.[Attribute]

  Gets the normalized value of this attribute

**getValueBinding(String)** - Method in class
javax.faces.component.[UIComponent]

  Deprecated. This has been replaced by
  [UIComponent.getValueExpression(java.lang.String)].

**getValueBinding(String)** - Method in class
javax.faces.component.[UIComponentBase]

  Deprecated. This has been replaced by
  [UIComponent.getValueExpression(java.lang.String)].

**getValueBinding(String)** - Method in class
javax.faces.component.[UIGraphic]

  Deprecated. This has been replaced by
  [UIGraphic.getValueExpression(java.lang.String)].

**getValueBinding(String)** - Method in class
javax.faces.component.UISelectBoolean
   **Deprecated.** This has been replaced by
   UISelectBoolean.getValueExpression(java.lang.String).

**getValuExpression(String)** - Method in class
javax.faces.component.UISelectMany
   **Deprecated.** This has been replaced by
   UISelectMany.getValueExpression(java.lang.String).

**getValueChangeListener()** - Method in interface
javax.faces.component.EditableValueHolder
   **Deprecated.** Use EditableValueHolder.getValueChangeListeners() instead.

**getValueChangeListener()** - Method in class
javax.faces.component.UIInput

**getValueChangeListeners()** - Method in interface
javax.faces.component.EditableValueHolder
   Return the set of registered ValueChangeListener for this component instance.

**getValueChangeListeners()** - Method in class
javax.faces.component.UIInput
   Return the set of registered ValueChangeListeners for this UIInput instance.

**getValueExpression(String)** - Method in class
javax.faces.component.UIComponent
   Return the ValueExpression used to calculate the value for the specified attribute or property name, if any.

**getValueExpression(String)** - Method in class
javax.faces.component.UIGraphic
   Return any ValueExpression set for value if a ValueExpression for url is requested; otherwise, perform the default superclass processing for this method.

**getValueExpression(String)** - Method in class
javax.faces.component.UISelectBoolean
   Return any ValueExpression set for value if a ValueExpression for selected is requested; otherwise, perform the default superclass processing for this method.

**getValueExpression(String)** - Method in class
javax.faces.component.UISelectMany
   Return any ValueExpression set for value if a ValueExpression for
selectedIndex is requested; otherwise, perform the default superclass processing for this method.

**getValueNames()** - Method in interface javax.servlet.http.HttpSession

*Deprecated. As of Version 2.2, this method is replaced by HttpSession.getAttributeNames()*

**getValues()** - Method in class javax.servlet.jsp.tagext.TagSupport

Enumerate the keys for the values kept by this tag handler.

**getValues()** - Method in interface javax.xml.registry.infomodel.Slot

Gets the values for this Slot.

**getValueType()** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Gets the value type for this object.

**getVar()** - Method in class javax.faces.component.UIData

Return the request-scope attribute under which the data object for the current row will be exposed when iterating.

**getVariableInfo(TagData)** - Method in class javax.servlet.jsp.tagext.TagExtraInfo

Information on scripting variables defined by the tag associated with this TagExtraInfo instance.

**getVariableInfo(TagData)** - Method in class javax.servlet.jsp.tagext.TagInfo

Information on the scripting objects created by this tag at runtime.

**getVariableMapper()** - Method in class javax.el.ELContext

Retrieves the VariableMapper associated with this ELContext.

**getVariableResolver()** - Method in class javax.servlet.jsp.JspContext

*Deprecated. This has been replaced by Application.getELResolver().*

**getVariableResolver()** - Method in class javax.servlet.jsp.tagext.VariableInfo

*Deprecated. As of JSP 2.1, replaced by ELContext.getELResolver(), which can be obtained by jspContext.getELContext().getELResolver().*

**getVarName()** - Method in class javax.servlet.jsp.tagext.VariableInfo

Returns the name of the scripting variable.

**getVendor()** - Method in class javax.mail.Provider

Returns name of vendor associated with this implementation or null

** getVersion()** - Method in class javax.mail.Provider

Returns version of this implementation or null if no version

Returns the version of the protocol this cookie complies with.

getVersion() - Method in interface javax.xml.registry.CapabilityProfile
   Gets the JAXR specification version supported by the JAXR provider.

getVersion() - Method in interface
   javax.xml.stream.events.StartDocument
   Gets the version of XML of this XML stream

getVersion() - Method in class
   javax.xml.stream.util.StreamReaderDelegate

getVersion() - Method in interface javax.xml.stream.XMLStreamReader
   Get the xml version declared on the xml declaration Returns null if none was declared

getViewHandler() - Method in class javax.faces.application.Application
   Return the ViewHandler instance that will be utilized during the Restore View and Render Response phases of the request processing lifecycle.

getViewId() - Method in exception
   javax.faces.application.ViewExpiredException
   Return the view identifier of this exception, or null if the view identifier is nonexistent or unknown.

getViewId() - Method in class javax.faces.component.UIViewRoot
   Return the view identifier for this view.

getViewRoot() - Method in class javax.faces.context.FacesContext
   Return the root component that is associated with the this request.

getVisibleNamespacePrefixes() - Method in interface
   javax.xml.soap.SOAPElement
   Returns an Iterator over the namespace prefix Strings visible to this element.

getWaitingThreadCount() - Method in interface
   javax.management.j2ee.statistics.JCACollectionPoolStats
   The number of threads waiting for a connection

getWaitingThreadCount() - Method in interface
   javax.management.j2ee.statistics.JDBCCollectionPoolStats
   Number of threads waiting for a connection.

getWaitTime() - Method in interface
   javax.management.j2ee.statistics.JCAConnectionStats
   Returns the time spent waiting for a connection to be available
javax.management.j2ee.statistics.JDBCConnectionStats
Time spent waiting for a connection to be available.

getWarnClass() - Method in class
javax.faces.component.html.HtmlMessage
Return the value of the warnClass property.

getWarnClass() - Method in class
javax.faces.component.html.HtmlMessages
Return the value of the warnClass property.

getWarnings() - Method in interface javax.resource.cci.Interaction
Gets the first ResourceWarning from the chain of warnings associated with this Interaction instance.

getWarnStyle() - Method in class
javax.faces.component.html.HtmlMessage
Return the value of the warnStyle property.

getWarnStyle() - Method in class
javax.faces.component.html.HtmlMessages
Return the value of the warnStyle property.

getWebURL() - Method in interface
javax.enterprise.deploy.spi.TargetModuleID
If this TargetModuleID represents a web module retrieve the URL for it.

getWidth() - Method in class javax.faces.component.html.HtmlDataTable
Return the value of the width property.

getWidth() - Method in class
javax.faces.component.html.HtmlGraphicImage
Return the value of the width property.

getWidth() - Method in class
javax.faces.component.html.HtmlPanelGrid
Return the value of the width property.

getWork() - Method in class javax.resource.spi.work.WorkEvent
Return the work instance which is the cause of the event.

getWorkManager() - Method in interface
javax.resource.spi.BootstrapContext
Provides a handle to a WorkManager instance.

getWrapped() - Method in class
javax.faces.application.StateManagerWrapper

getWrapped() - Method in class
javax.faces.application.ViewHandlerWrapper
**getWrapped()** - Method in class `javax.faces.context.ResponseWriterWrapper`
javax.resource.spi.**ManagedConnection**
  Returns an `javax.transaction.xa.XAresource` instance.

**getXAResources(ActivationSpec[])** - Method in interface `javax.resource.spi. ResourceAdapter`
  This method is called by the application server during crash recovery.

**getXATerminator()** - Method in interface `javax.resource.spi. BootstrapContext`
  Provides a handle to a `XATerminator` instance.

**getXid()** - Method in class `javax.resource.spi.work. ExecutionContext`

**getXMLNode(Object)** - Method in class `javax.xml.bind. Binder`
  Gets the XML element associated with the given JAXB object.

**getXMLReader()** - Method in class `javax.xml.bind.helpers. AbstractUnmarshallerImpl`
  Obtains a configured XMLReader.

**getXMLReporter()** - Method in class `javax.xml.stream. XMLInputFactory`
  The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

**getXMLResolver()** - Method in class `javax.xml.stream. XMLInputFactory`
  The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

**getXpath()** - Method in interface `javax.enterprise.deploy.model. DDBean`
  Returns the original xpath string provided by the DConfigBean.

**getXpath()** - Method in interface `javax.enterprise.deploy.model. DDBeanRoot`
  Return the XPath for this standard bean.

**getXpaths()** - Method in interface `javax.enterprise.deploy.spi. DConfigBean`
  Return a list of XPaths designating the deployment descriptor information this DConfigBean requires.

**GT** - Static variable in class `javax.mail.search. ComparisonTerm`
**Handle** - Interface in *javax.ejb*
   The Handle interface is implemented by all EJB object handles.

**HandleDelegate** - Interface in *javax.ejb.spi*
   The HandleDelegate interface is implemented by the EJB container.

**handleEvent(ValidationEvent)** - Method in class *javax.xml.bind.helpers.DefaultValidationEventHandler*

**handleEvent(ValidationEvent)** - Method in class *javax.xml.bind.util.ValidationEventCollector*

**handleEvent(ValidationEvent)** - Method in interface *javax.xml.bind.ValidationEventHandler*
   Receive notification of a validation warning or error.

**handleFault(MessageContext)** - Method in class *javax.xml.rpc.handler.GenericHandler*
   The handleFault method processes the SOAP faults based on the SOAP message processing model.

**handleFault(MessageContext)** - Method in interface *javax.xml.rpc.handler.Handler*
   The handleFault method processes the SOAP faults based on the SOAP message processing model.

**handleFault(MessageContext)** - Method in interface *javax.xml.rpc.handler.HandlerChain*
   The handleFault method initiates the SOAP fault processing for this handler chain.

**handleFault(C)** - Method in interface *javax.xml.ws.handler.Handler*
   The handleFault method is invoked for fault message processing.

**handleMessage(C)** - Method in interface *javax.xml.ws.handler.Handler*
   The handleMessage method is invoked for normal processing of inbound and outbound messages.

**handleNavigation(FacesContext, String, String)** - Method in class *javax.faces.application.NavigationHandler*
   Perform navigation processing based on the state information in the specified FacesContext, plus the outcome string returned by an executed application action.
**handlePageException(Exception)** - Method in class javax.servlet.jsp.PageContext
   This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP.

**handlePageException(Throwable)** - Method in class javax.servlet.jsp.PageContext
   This method is intended to process an unhandled 'page' level exception by forwarding the exception to the specified error page for this JSP.

**handleProgressEvent(ProgressEvent)** - Method in interface javax.enterprise.deploy.spi.status.ProgressListener
   Invoked when a deployment progress event occurs.

**Handler** - Interface in javax.xml.rpc.handler
   The javax.xml.rpc.handler.Handler interface is required to be implemented by a SOAP message handler.

**Handler<C extends MessageContext>** - Interface in javax.xml.ws.handler
   The Handler interface is the base interface for JAX-WS handlers.

**HandlerChain** - Annotation Type in javax.jws
   Associates the Web Service with an externally defined handler chain.

**HandlerChain** - Interface in javax.xml.rpc.handler
   The javax.xml.rpc.handler.HandlerChain interface represents a list of handlers.

**handleRequest(MessageContext)** - Method in class javax.xml.rpc.handler.GenericHandler
   The handleRequest method processes the request SOAP message.

**handleRequest(MessageContext)** - Method in interface javax.xml.rpc.handler.Handler
   The handleRequest method processes the request message.

**handleRequest(MessageContext)** - Method in interface javax.xml.rpc.handler.HandlerChain
   The handleRequest method initiates the request processing for this handler chain.

**handleResponse(MessageContext)** - Method in class javax.xml.rpc.handler.GenericHandler
   The handleResponse method processes the response message.
The handleResponse method processes the response SOAP message.

**handleResponse(MessageContext)** - Method in interface javax.xml.rpc.handler.HandlerChain
The handleResponse method initiates the response processing for this handler chain.

**handleResponse(Response<T>)** - Method in interface javax.xml.ws.AsyncHandler
Called when the response to an asynchronous operation is available.

**HandlerInfo** - Class in javax.xml.rpc.handler
The javax.xml.rpc.handler.HandlerInfo represents information about a handler in the HandlerChain.

**HandlerInfo()** - Constructor for class javax.xml.rpc.handler.HandlerInfo
Default constructor

**HandlerInfo(Class, Map, QName[])** - Constructor for class javax.xml.rpc.handler.HandlerInfo
Constructor for HandlerInfo

**HandlerRegistry** - Interface in javax.xml.rpc.handler
The javax.xml.rpc.handler.HandlerRegistry provides support for the programmatic configuration of handlers in a HandlerRegistry.

**HandlerResolver** - Interface in javax.xml.ws.handler
HandlerResolver is an interface implemented by an application to get control over the handler chain set on proxy/dispatch objects at the time of their creation.

**handlesURI(String)** - Method in interface javax.enterprise.deploy.spi.factories.DeploymentFactory
Tests whether this factory can create a DeploymentManager object based on the specified URI.

**hasBinding()** - Method in class javax.faces.webapp.UIClassicTagBase
Return true if this component has a non-null binding attribute.

**hasBinding()** - Method in class javax.faces.webapp.UIClassicTag
Deprecated.

**hasDetail()** - Method in interface javax.xml.soap.SOAPFault
Returns true if this SOAPFault has a Detail subelement and false otherwise.
**hasDynamicAttributes()** - Method in class javax.servlet.jsp.tagext.TagInfo
Get dynamicAttributes associated with this TagInfo.

**hasEvents()** - Method in class javax.xml.bind.util.ValidationEventCollector
Returns true if this event collector contains at least one ValidationEvent.

**hasFault()** - Method in interface javax.xml.soap.SOAPBody
Indicates whether a SOAPFault object exists in this SOAPBody object.

**hashCode()** - Method in class javax.el.Expression
Returns the hash code for this Expression.

**hashCode()** - Method in class javax.faces.validator.DoubleRangeValidator

**hashCode()** - Method in class javax.faces.validator.LengthValidator

**hashCode()** - Method in class javax.faces.validator.LongRangeValidator

**hashCode()** - Method in class javax.mail.Flags
Compute a hash code for this Flags object.

**hashCode()** - Method in class javax.mail.internet.InternetAddress
Compute a hash code for the address.

**hashCode()** - Method in class javax.mail.internet.NewsAddress
Compute a hash code for the address.

**hashCode()** - Method in class javax.mail.search.AddressTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.AndTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.ComparisonTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.DateTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.FlagTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.HeaderTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.IntegerComparisonTerm
Compute a hash code for this object.

**hashCode()** - Method in class javax.mail.search.NotTerm

Compute a hashCode for this object.

**hashCode()** - Method in class `javax.mail.search.OrTerm`
Compute a hashCode for this object.

**hashCode()** - Method in class `javax.mail.search.RecipientStringTerm`
Compute a hashCode for this object.

**hashCode()** - Method in class `javax.mail.search.RecipientTerm`
Compute a hashCode for this object.

**hashCode()** - Method in class `javax.mail.search.StringTerm`
Compute a hashCode for this object.

**hashCode()** - Method in class `javax.mail.URLName`
Compute the hash code for this URLName.

**hashCode()** - Method in interface `javax.resource.cci.Record`
Returns the hash code for the Record instance.

**hashCode()** - Method in interface
`javax.resource.spi.ConnectionRequestInfo`.
Returns the hash code of the ConnectionRequestInfo.

**hashCode()** - Method in interface
`javax.resource.spi.ManagedConnectionFactory`.
Returns the hash code for the ManagedConnectionFactory.

**hashCode()** - Method in interface
`javax.resource.spi.security_GENERICCREDENTIAL`.
**Deprecated.** Returns the hash code for this GenericCredential

**hashCode()** - Method in class
`javax.resource.spi.security.PasswordCredential`.
Returns the hash code for this PasswordCredential.

**hashCode()** - Method in class
`javax.security.jacc.EJBMethodPermission`
Returns the hash code value for this EJBMethodPermission.

**hashCode()** - Method in class
`javax.security.jacc.EJBRoleRefPermission`
Returns the hash code value for this EJBRoleRefPermission.

**hashCode()** - Method in class
`javax.security.jacc.WebResourcePermission`
Returns the hash code value for this WebResourcePermission.

**hashCode()** - Method in class
`javax.security.jacc.WebRoleRefPermission`
Returns the hash code value for this WebRoleRefPermission.

**hashCode()** - Method in class
`javax.security.jacc.WebUserDataPermission`
Returns the hash code value for this WebUserDataPermission.

**hasName()** - Method in class
`javax.xml.stream.util.StreamReaderDelegate`
**hasName()** - Method in interface [javax.xml.stream.XMLStreamReader](https://docs.oracle.com/javase/8/docs/api/java/xml/stream/(XMLStreamReader))
returns true if the current event has a name (is a `START_ELEMENT` or `END_ELEMENT`) returns false otherwise

**hasNewMessages()** - Method in class [javax.mail.Folder](https://docs.oracle.com/javase/8/docs/api/javax/mail/Folder)
Returns true if this Folder has new messages since the last time this indication was reset.

**hasNext()** - Method in class [javax.xml.stream.util.EventReaderDelegate](https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/util/EventReaderDelegate)

**hasText()** - Method in class [javax.xml.stream.util.StreamReaderDelegate](https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/util/StreamReaderDelegate)

**hasText()** - Method in interface [javax.xml.stream.XMLStreamReader](https://docs.oracle.com/javase/8/docs/api/java/xml/stream/XMLStreamReader)
Check if there are more events.

**hasNext()** - Method in interface [javax.xml.stream.XMLStreamReader](https://docs.oracle.com/javase/8/docs/api/java/xml/stream/XMLStreamReader)
Returns true if there are more parsing events and false if there are no more events.

**hasNext()** - Method in class [javax.xml.stream.util.StreamReaderDelegate](https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/util/StreamReaderDelegate)

**hasNext()** - Method in interface [javax.xml.stream.XMLEventReader](https://docs.oracle.com/javase/8/docs/api/javax/xml/stream/XMLEventReader)
Check if there are more events.

**hasText()** - Method in interface [javax.xml.stream.XMLStreamReader](https://docs.oracle.com/javase/8/docs/api/java/xml/stream/XMLStreamReader)
Return true if the current event has text, false otherwise.
The following events have text: `CHARACTERS`, `DTD`, `ENTITY_REFERENCE`, `COMMENT`, `SPACE`

**Header** - Class in [javax.mail](https://docs.oracle.com/javase/8/docs/api/javax/mail)
The Header class stores a name/value pair to represent headers.

**Header(String, String)** - Constructor for class [javax.mail.Header](https://docs.oracle.com/javase/8/docs/api/javax/mail/Header)
Construct a Header object.

**headerName** - Variable in class [javax.mail.search.HeaderTerm](https://docs.oracle.com/javase/8/docs/api/javax/mail/search/HeaderTerm)
The name of the header.

**headers** - Variable in class [javax.mail.internet.InternetHeaders](https://docs.oracle.com/javase/8/docs/api/javax/mail/internet/InternetHeaders)
The actual list of Headers, including placeholder entries.

**headers** - Variable in class [javax.mail.internet.MimeBodyPart](https://docs.oracle.com/javase/8/docs/api/javax/mail/internet/MimeBodyPart)
The InternetHeaders object that stores all the headers of this body part.

**headers** - Variable in class [javax.mail.internet.MimeMessage](https://docs.oracle.com/javase/8/docs/api/javax/mail/internet/MimeMessage)
The InternetHeaders object that stores the header of this message.

**HeaderTerm** - Class in [javax.mail.search](https://docs.oracle.com/javase/8/docs/api/javax/mail/search)
This class implements comparisons for Message headers.

**HeaderTerm(String, String)** - Constructor for class [javax.mail.search.HeaderTerm](https://docs.oracle.com/javase/8/docs/api/javax/mail/search/HeaderTerm)
HeaderTokenizer - Class in javax.mail.internet
This class tokenizes RFC822 and MIME headers into the basic symbols specified by RFC822 and MIME.

HeaderTokenizer(String, String, boolean) - Constructor for class javax.mail.internet. HeaderTokenizer
Constructor that takes a rfc822 style header.

HeaderTokenizer(String, String) - Constructor for class javax.mail.internet. HeaderTokenizer
Constructor.

HeaderTokenizer(String) - Constructor for class javax.mail.internet. HeaderTokenizer
Constructor.

HeaderTokenizer.Token - Class in javax.mail.internet
The Token class represents tokens returned by the HeaderTokenizer.

HeaderTokenizer.Token(int, String) - Constructor for class javax.mail.internet. HeaderTokenizer.Token
Constructor.

HeuristicCommitException - Exception in javax.transaction
This exception is thrown by the rollback operation on a resource to report that a heuristic decision was made and that all relevant updates have been committed.

HeuristicCommitException() - Constructor for exception javax.transaction.HeuristicCommitException

HeuristicCommitException(String) - Constructor for exception javax.transaction.HeuristicCommitException

HeuristicMixedException - Exception in javax.transaction
This exception is thrown to report that a heuristic decision was made and that some relevant updates have been committed and others have been rolled back.

HeuristicMixedException() - Constructor for exception javax.transaction.HeuristicMixedException

HeuristicMixedException(String) - Constructor for exception javax.transaction.HeuristicMixedException

HeuristicRollbackException - Exception in javax.transaction
This exception is thrown by the commit operation to report that a heuristic decision was made and that all relevant updates have been rolled back.

**HeuristicRollbackException()** - Constructor for exception `javax.transaction.HeuristicRollbackException`

**HeuristicRollbackException(String)** - Constructor for exception `javax.transaction.HeuristicRollbackException`

**HexBinaryAdapter** - Class in `javax.xml.bind.annotation.adapters.XmlAdapter` for `xs:hexBinary`.

**HexBinaryAdapter()** - Constructor for class `javax.xml.bind.annotation.adapters.HexBinaryAdapter`

**Holder** - Interface in `javax.xml.rpc.holders`. The `javax.xml.rpc.holders.Holder` interface represents the base interface for both standard and generated `Holder` classes.

**Holder<T>** - Class in `javax.xml.ws`. Holds a value of type `T`.

**Holder()** - Constructor for class `javax.xml.ws.Holder`. Creates a new holder with a `null` value.

**Holder(T)** - Constructor for class `javax.xml.ws.Holder`. Create a new holder with the specified value.

**HOLDS_FOLDERS** - Static variable in class `javax.mail.Folder`. This folder can contain other folders.

**HOLDS_MESSAGES** - Static variable in class `javax.mail.Folder`. This folder can contain messages.

**HomeHandle** - Interface in `javax.ejb`. The `HomeHandle` interface is implemented by all home object handles.

**host** - Variable in class `javax.mail.internet.NewsAddress`.

**HTML_BASIC_RENDER_KIT** - Static variable in class `javax.faces.render.RenderKitFactory`. The render kit identifier of the default `RenderKit` instance for this JavaServer Faces implementation.

**HtmlColumn** - Class in `javax.faces.component.html`. Represents a column that will be rendered in an HTML table element.
**HtmlColumn()** - Constructor for class javax.faces.component.html.HtmlColumn

**HtmlCommandButton** - Class in javax.faces.component.html

  Represents an HTML input element for a button of type submit or reset.

**HtmlCommandButton()** - Constructor for class javax.faces.component.html.HtmlCommandButton

**HtmlCommandLink** - Class in javax.faces.component.html

  Represents an HTML a element for a hyperlink that acts like a submit button.

**HtmlCommandLink()** - Constructor for class javax.faces.component.html.HtmlCommandLink

**HtmlDataTable** - Class in javax.faces.component.html

  Represents a set of repeating data (segregated into columns by child UIColumn components) that will be rendered in an HTML table element.

**HtmlDataTable()** - Constructor for class javax.faces.component.html.HtmlDataTable

**HtmlForm** - Class in javax.faces.component.html

  Represents an HTML form element.

**HtmlForm()** - Constructor for class javax.faces.component.html.HtmlForm

**HtmlGraphicImage** - Class in javax.faces.component.html

  Represents an HTML img element, used to retrieve and render a graphical image.

**HtmlGraphicImage()** - Constructor for class javax.faces.component.html.HtmlGraphicImage

**HtmlInputHidden** - Class in javax.faces.component.html

  Represents an HTML input element of type hidden.

**HtmlInputHidden()** - Constructor for class javax.faces.component.html.HtmlInputHidden

**HtmlInputSecret** - Class in javax.faces.component.html
Represents an HTML input element of type password.

**HtmlInputSecret()** - Constructor for class javax.faces.component.html.HtmlInputSecret

**HtmlInputText** - Class in javax.faces.component.html

Represents an HTML input element of type text.

**HtmlInputText()** - Constructor for class javax.faces.component.html.HtmlInputText

**HtmlInputTextarea** - Class in javax.faces.component.html

Represents an HTML textarea element.

**HtmlInputTextarea()** - Constructor for class javax.faces.component.html.HtmlInputTextarea

**HtmlMessage** - Class in javax.faces.component.html

By default, the `rendererType` property must be set to "javax.faces.Message".

**HtmlMessage()** - Constructor for class javax.faces.component.html.HtmlMessage

**HtmlMessages** - Class in javax.faces.component.html

By default, the `rendererType` property must be set to "javax.faces.Messages".

**HtmlMessages()** - Constructor for class javax.faces.component.html.HtmlMessages

**HtmlOutputFormat** - Class in javax.faces.component.html

Represents a component that looks up a localized message in a resource bundle, optionally uses it as a `MessageFormat` pattern string and substitutes in parameter values from nested `UIParameter` components, and renders the result.

**HtmlOutputFormat()** - Constructor for class javax.faces.component.html.HtmlOutputFormat

**HtmlOutputLabel** - Class in javax.faces.component.html

Represents an HTML label element, used to define an accessible label for a corresponding input element.

**HtmlOutputLabel()** - Constructor for class javax.faces.component.html.HtmlOutputLabel
HtmlOutputLink - Class in javax.faces.component.html
Represents an HTML a (hyperlink) element that may be used to link to an arbitrary URL defined by the value property.
HtmlOutputLink() - Constructor for class
javax.faces.component.html.HtmlOutputLink

HtmlOutputText - Class in javax.faces.component.html
Renders the component value as text, optionally wrapping in a span element if I18N attributes, CSS styles or style classes are specified.
HtmlOutputText() - Constructor for class
javax.faces.component.html.HtmlOutputText

HtmlPanelGrid - Class in javax.faces.component.html
Renders child components in a table, starting a new row after the specified number of columns.
HtmlPanelGrid() - Constructor for class
javax.faces.component.html.HtmlPanelGrid

HtmlPanelGroup - Class in javax.faces.component.html
Causes all child components of this component to be rendered.
HtmlPanelGroup() - Constructor for class
javax.faces.component.html.HtmlPanelGroup

HtmlSelectBooleanCheckbox - Class in javax.faces.component.html
Represents an HTML input element of type checkbox.
HtmlSelectBooleanCheckbox() - Constructor for class
javax.faces.component.html.HtmlSelectBooleanCheckbox

HtmlSelectManyCheckbox - Class in javax.faces.component.html
Represents a multiple-selection component that is rendered as a set of HTML input elements of type checkbox.
HtmlSelectManyCheckbox() - Constructor for class
javax.faces.component.html.HtmlSelectManyCheckbox

HtmlSelectManyListbox - Class in javax.faces.component.html
Represents a multiple-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.
**HtmlSelectManyListbox()** - Constructor for class javax.faces.component.html.HtmlSelectManyListbox

**HtmlSelectManyMenu** - Class in javax.faces.component.html
Represents a multiple-selection component that is rendered as an HTML select element, showing a single available option at a time.

**HtmlSelectManyMenu()** - Constructor for class javax.faces.component.html.HtmlSelectManyMenu

**HtmlSelectOneListbox** - Class in javax.faces.component.html
Represents a single-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.

**HtmlSelectOneListbox()** - Constructor for class javax.faces.component.html.HtmlSelectOneListbox

**HtmlSelectOneMenu** - Class in javax.faces.component.html
Represents a single-selection component that is rendered as an HTML select element, showing a single available option at a time.

**HtmlSelectOneMenu()** - Constructor for class javax.faces.component.html.HtmlSelectOneMenu

**HtmlSelectOneRadio** - Class in javax.faces.component.html
Represents a single-selection component that is rendered as a set of HTML input elements of type radio.

**HtmlSelectOneRadio()** - Constructor for class javax.faces.component.html.HtmlSelectOneRadio

**HTTP_BINDING** - Static variable in interface javax.xml.ws.http.HTTPBinding
A constant representing the identity of the XML/HTTP binding.

**HTTP_REQUEST_HEADERS** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: HTTP request headers.

**HTTP_REQUEST_METHOD** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: HTTP request method.

**HTTP_RESPONSE_CODE** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: HTTP response status code.

**HTTP RESPONSE_HEADERS** - Static variable in interface javax.xml.ws.handler.MessageContext

  Standard property: HTTP response headers.

**HTTPBinding** - Interface in javax.xml.ws.http

  The HTTPBinding interface is an abstraction for the XML/HTTP binding.

**HTTPException** - Exception in javax.xml.ws.http

  The HTTPException exception represents a XML/HTTP fault.

**HTTPException(int)** - Constructor for exception javax.xml.ws.http.HttpException

  Constructor for the HTTPException

**HttpJspPage** - Interface in javax.servlet.jsp

  The HttpJspPage interface describes the interaction that a JSP Page Implementation Class must satisfy when using the HTTP protocol.

**HttpServlet** - Class in javax.servlet.http

  Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.

**HttpServlet()** - Constructor for class javax.servlet.http.HttpServlet

  Does nothing, because this is an abstract class.

**HttpServletRequest** - Interface in javax.servlet.http

  Extends the ServletRequest interface to provide request information for HTTP servlets.

**HttpServletRequestWrapper** - Class in javax.servlet.http

  Provides a convenient implementation of the HttpServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet.

**HttpServletRequestWrapper(HttpServletRequest)** - Constructor for class javax.servlet.http.HttpServletRequestWrapper

  Constructs a request object wrapping the given request.

**HttpServletRequestWrapper(HttpServletRequest)** - Interface in javax.servlet.http

  Extends the ServletResponse interface to provide HTTP-specific functionality in sending a response.

**HttpServletResponse** - Interface in javax.servlet.http

  Provides a convenient implementation of the HttpServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet.

**HttpServletResponseWrapper** - Class in javax.servlet.http

  Provides a convenient implementation of the HttpServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet.
Constructs a response adaptor wrapping the given response.

**HttpSession** - Interface in **javax.servlet.http**
Provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.

**HttpSessionActivationListener** - Interface in **javax.servlet.http**
Objects that are bound to a session may listen to container events notifying them that sessions will be passivated and that session will be activated.

**HttpSessionAttributeListener** - Interface in **javax.servlet.http**
This listener interface can be implemented in order to get notifications of changes to the attribute lists of sessions within this web application.

**HttpSessionBindingEvent** - Class in **javax.servlet.http**
Events of this type are either sent to an object that implements **HttpSessionBindingListener** when it is bound or unbound from a session, or to a **HttpSessionAttributeListener** that has been configured in the deployment descriptor when any attribute is bound, unbound or replaced in a session.

**HttpSessionBindingEvent(HttpSession, String)** - Constructor for class **javax.servlet.http.HttpSessionBindingEvent**
Constructs an event that notifies an object that it has been bound to or unbound from a session.

**HttpSessionBindingEvent(HttpSession, String, Object)** - Constructor for class **javax.servlet.http.HttpSessionBindingEvent**
Constructs an event that notifies an object that it has been bound to or unbound from a session.

**HttpSessionBindingListener** - Interface in **javax.servlet.http**
Causes an object to be notified when it is bound to or unbound from a session.

**HttpSessionContext** - Interface in **javax.servlet.http**
Deprecated. As of Java(tm) Servlet API 2.1 for security reasons, with no replacement. This interface will be removed in a future version of this API.

**HttpSessionEvent** - Class in **javax.servlet.http**
This is the class representing event notifications for changes to sessions within a web application.

**HttpSessionEvent(HttpSession)** - Constructor for class **javax.servlet.http.HttpSessionEvent**
Construct a session event from the given source.
**HttpSessionListener** - Interface in [javax.servlet.http](https://docs.oracle.com/javaee/7/api/javax/servlet/http/HttpSessionListener.html)
Implementations of this interface are notified of changes to the list of active sessions in a web application.

**HttpUtils** - Class in [javax.servlet.http](https://docs.oracle.com/javaee/7/api/javax/servlet/http/HttpUtils.html)
**Deprecated.** As of Java(tm) Servlet API 2.3. These methods were only useful with the default encoding and have been moved to the request interfaces.

**Deprecated.** Constructs an empty `HttpUtils` object.
Id - Annotation Type in javax.persistence
  Specifies the primary key property or field of an entity.

id - Variable in class javax.resource.spi.ConnectionEvent
  Type of the event

ID - Static variable in class javax.servlet.jsp.tagext.TagAttributeInfo
  "id" is wired in to be ID.

id - Variable in class javax.servlet.jsp.tagext.TagSupport
  The value of the id attribute of this tag; or null.

IdClass - Annotation Type in javax.persistence
  Specifies a composite primary key class that is mapped to multiple fields or properties of the entity.

ignoreCase - Variable in class javax.mail.search.StringTerm
  Ignore case when comparing?

IllegalStateException - Exception in javax.jms
  This exception is thrown when a method is invoked at an illegal or inappropriate time or if the provider is not in an appropriate state for the requested operation.

IllegalStateException(String, String) - Constructor for exception javax.jms.IllegalStateException
  Constructs an IllegalStateException with the specified reason and error code.

IllegalStateException(String) - Constructor for exception javax.jms.IllegalStateException
  Constructs a new instance with the specified reason.

IllegalStateException(throwable) - Constructor for exception javax.resource.spi.IllegalStateException
  Constructs a new instance with null as its detail message.
javax.resource.spi.**IllegalStateException**

Constructs a new throwable with the specified cause.

**IllegalStateException(String, Throwable)** - Constructor for exception

Constructs a new throwable with the specified detail message and cause.

**IllegalStateException(String, String)** - Constructor for exception

Constructs a new throwable with the specified detail message and an error code.

**IllegalWriteException** - Exception in **javax.mail**

The exception thrown when a write is attempted on a read-only attribute of any Messaging object.

**IllegalWriteException()** - Constructor for exception

Constructs a IllegalWriteException with no detail message.

**IllegalWriteException(String)** - Constructor for exception

Constructs a IllegalWriteException with the specified detail message.

**IMMEDIATE** - Static variable in interface

javax.resource.spi.work.**WorkManager**

A constant to indicate timeout duration.

**ImplicitObjectELResolver** - Class in **javax.servlet.jsp.el**

Defines variable resolution behavior for the EL implicit objects defined in the JSP specification.

**ImplicitObjectELResolver()** - Constructor for class

javax.servlet.jsp.el.**ImplicitObjectELResolver**

**implies(Permission)** - Method in class

javax.security.jacc.**EJBMethodPermission**

Determines if the argument Permission is "implied by" this EJBMethodPermission.

**implies(Permission)** - Method in class

javax.security.jacc.**EJBRoleRefPermission**

Determines if the argument Permission is "implied by" this EJBRoleRefPermission.

**implies(Permission)** - Method in class

javax.security.jacc.**WebResourcePermission**

Determines if the argument Permission is "implied by" this
WebResourcePermission.

**implies(Permission)** - Method in class javax.security.jacc.WebRoleRefPermission

Determine if the argument Permission is "implied by" this WebRoleRefPermission.

**implies(Permission)** - Method in class javax.security.jacc.WebUserDataPermission

Determine if the argument Permission is "implied by" this WebUserDataPermission.

**in** - Variable in class javax.mail.util.SharedFileInputStream

The file containing the data.

**IN** - Static variable in class javax.xml.rpc.ParameterMode

IN mode for parameter passing.

**INBOUND_MESSAGE_ATTACHMENTS** - Static variable in interface javax.xml.ws.handler.MessageContext

Standard property: Map of attachments to a message for the inbound message, key is the MIME Content-ID, value is a DataHandler.

**include(String)** - Method in class javax.servlet.jsp.PageContext

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread.

**include(String, boolean)** - Method in class javax.servlet.jsp.PageContext

Causes the resource specified to be processed as part of the current ServletRequest and ServletResponse being processed by the calling Thread.

**include(ServletRequest, ServletResponse)** - Method in interface javax.servlet.RequestDispatcher

Includes the content of a resource (servlet, JSP page, HTML file) in the response.

**INDEFINITE** - Static variable in interface javax.resource.spi.work.WorkManager

A constant to indicate timeout duration.

**IndexedRecord** - Interface in javax.resource.cci

IndexedRecord represents an ordered collection of record elements based on the java.util.List interface.

**info** - Variable in class javax.servlet.jsp.tagext.TagLibraryInfo

Information (documentation) for this TLD.
**Inheritance** - Annotation Type in `javax.persistence`  
Defines the inheritance strategy to be used for an entity class hierarchy.

**InheritanceType** - Enum in `javax.persistence`  
Defines inheritance strategy options.

**Init** - Annotation Type in `javax.ejb`  
Designates a method of a session bean that corresponds to the create method of an adapted Home interface or an adapted Local Home interface.

**init(ServletConfig)** - Method in class `javax.faces.webapp.FacesServlet`  
Acquire the factory instances we will require.

**init(FilterConfig)** - Method in interface `javax.servlet.Filter`  
Called by the web container to indicate to a filter that it is being placed into service.

**init(ServletConfig)** - Method in class `javax.servlet.GenericServlet`  
Called by the servlet container to indicate to a servlet that the servlet is being placed into service.

**init()** - Method in class `javax.servlet.GenericServlet`  
A convenience method which can be overridden so that there's no need to call `super.init(config)`.

**init(ServletConfig)** - Method in interface `javax.servlet.Servlet`  
Called by the servlet container to indicate to a servlet that the servlet is being placed into service.

**init(HandlerInfo)** - Method in class `javax.xml.rpc.handler.GenericHandler`  
The `init` method to enable the Handler instance to initialize itself.

**init(HandlerInfo)** - Method in interface `javax.xml.rpc.handler.Handler`  
The `init` method enables the Handler instance to initialize itself.

**init(Map)** - Method in interface `javax.xml.rpc.handler.HandlerChain`  
Initializes the configuration for a HandlerChain.

**init(Object)** - Method in interface `javax.xml.rpc.server.ServiceLifecycle`  
Used for initialization of a service endpoint.

**initCause(Throwable)** - Method in exception `javax.xml.registry.JAXRException`  
Initializes the `cause` of this throwable to the specified value.

**initCause(Throwable)** - Method in exception `javax.xml.soap.SOAPException`  
Initializes the `cause` field of this SOAPException object with the given Throwable object.

**initialize(Servlet, ServletRequest, ServletResponse, String, boolean,}
int, boolean) - Method in class javax.servlet.jsp.PageContext
   The initialize method is called to initialize an uninitialized PageContext so that it may be used by a JSP Implementation class to service an incoming request and response within it's _jspService() method.

InitParam - Annotation Type in javax.jws.soap
   Deprecated. As of JSR-181 2.0 with no replacement.

initView(FacesContext) - Method in class javax.faces.application.ViewHandler
   Initialize the view for the request processing lifecycle.

initView(FacesContext) - Method in class javax.faces.application.ViewHandlerWrapper
   The default behavior of this method is to call
   ViewHandler.initView(javax.faces.context.FacesContext) on the wrapped ViewHandler object.

INLINE - Static variable in interface javax.mail.Part
   This part should be presented inline.

INOUT - Static variable in class javax.xml.rpc.ParameterMode
   INOUT mode for parameter passing

insertsAreDetected(int) - Method in interface javax.resource.cci.ResultSetInfo
   Indicates whether or not a visible row insert can be detected by calling ResultSet.rowInserted.

inService() - Method in interface javax.security.jacc.PolicyConfiguration
   This method is used to determine if the policy context whose interface is this PolicyConfiguration Object is in the "inService" state.

inService(String) - Method in class javax.security.jacc.PolicyConfigurationFactory
   This method determines if the identified policy context exists with state "inService" in the Policy provider associated with the factory.

INTEGER_ID - Static variable in class javax.faces.convert.IntegerConverter
   The message identifier of the FacesMessage to be created if the conversion to Integer fails.

IntegerComparisonTerm - Class in javax.mail.search
   This class implements comparisons for integers.

IntegerComparisonTerm(int, int) - Constructor for class javax.mail.search.IntegerComparisonTerm
**IntegerConverter** - Class in `javax.faces.convert` Converter implementation for `java.lang.Integer` (and int primitive) values.

**IntegerConverter()** - Constructor for class `javax.faces.convert.IntegerConverter`

**IntegerWrapperHolder** - Class in `javax.xml.rpc.holders`

**IntegerWrapperHolder()** - Constructor for class `javax.xml.rpc.holders.IntegerWrapperHolder`

**IntegerWrapperHolder(Integer)** - Constructor for class `javax.xml.rpc.holders.IntegerWrapperHolder`

**Interaction** - Interface in `javax.resource.cci` The `javax.resource.cci.Interaction` enables a component to execute EIS functions.

**InteractionSpec** - Interface in `javax.resource.cci` An `InteractionSpec` holds properties for driving an Interaction with an EIS instance.

**Interceptors** - Annotation Type in `javax.interceptor` Declares an ordered list of interceptors for a class or method.

**INTERNAL** - Static variable in exception `javax.resource.spi.work.WorkException` Indicates an internal error condition.

**INTERNATIONAL_STRING** - Static variable in interface `javax.xml.registry.LifeCycleManager` This interface represents a String that has been internationalized into several Locales.

**InternationalString** - Interface in `javax.xml.registry.infomodel` This interface represents a String that has been internationalized into several Locales.

**InternetAddress** - Class in `javax.mail.internet` This class represents an Internet email address using the syntax of `RFC822`.

**InternetAddress()** - Constructor for class `javax.mail.internet.InternetAddress` Default constructor.

**InternetAddress(String)** - Constructor for class `javax.mail.internet.InternetAddress`
Constructor.

**InternetAddress(String, boolean)** - Constructor for class `javax.mail.internet.InternetAddress`  
Parse the given string and create an InternetAddress.

**InternetAddress(String, String)** - Constructor for class `javax.mail.internet.InternetAddress`  
Construct an InternetAddress given the address and personal name.

**InternetAddress(String, String, String)** - Constructor for class `javax.mail.internet.InternetAddress`  
Construct an InternetAddress given the address and personal name.

**InternetHeaders** - Class in `javax.mail.internet`  
InternetHeaders is a utility class that manages RFC822 style headers.

**InternetHeaders()** - Constructor for class `javax.mail.internet.InternetHeaders`  
Create an empty InternetHeaders object.

**InternetHeaders(InputStream)** - Constructor for class `javax.mail.internet.InternetHeaders`  
Read and parse the given RFC822 message stream till the blank line separating the header from the body.

**InternetHeaders.InternetHeader** - Class in `javax.mail.internet`  
An individual internet header.

**InternetHeaders.InternetHeader(String)** - Constructor for class `javax.mail.internet.InternetHeaders.InternetHeader`  
Constructor that takes a line and splits out the header name.

**InternetHeaders.InternetHeader(String, String)** - Constructor for class `javax.mail.internet.InternetHeaders.InternetHeader`  
Constructor that takes a header name and value.

**IntHolder** - Class in `javax.xml.rpc.holders`  

**IntHolder()** - Constructor for class `javax.xml.rpc.holders.IntHolder`

**IntHolder(int)** - Constructor for class `javax.xml.rpc.holders.IntHolder`

**invalid** - Variable in class `javax.mail.event.TransportEvent`

**invalid** - Variable in exception `javax.mail.SendFailedException`

**INVALID_MESSAGE_ID** - Static variable in class
javax.faces.component.UISelectMany
The message identifier of the FacesMessage to be created if a value not matching the available options is specified.

INVALID_MESSAGE_ID - Static variable in class javax.faces.component.UISelectOne
The message identifier of the FacesMessage to be created if a value not matching the available options is specified.

invalidate() - Method in interface javax.servlet.http.HttpSession
Invalidates this session then unbinds any objects bound to it.

InvalidClientIDException - Exception in javax.jms
This exception must be thrown when a client attempts to set a connection's client ID to a value that is rejected by a provider.

InvalidClientIDException(String, String) - Constructor for exception javax.jms.InvalidClientIDException
Constructs an InvalidClientIDException with the specified reason and error code.

InvalidClientIDException(String) - Constructor for exception javax.jms.InvalidClientIDException
Constructs an InvalidClientIDException with the specified reason.

InvalidDestinationException - Exception in javax.jms
This exception must be thrown when a destination either is not understood by a provider or is no longer valid.

InvalidDestinationException(String, String) - Constructor for exception javax.jms.InvalidDestinationException
Constructs an InvalidDestinationException with the specified reason and error code.

InvalidDestinationException(String) - Constructor for exception javax.jms.InvalidDestinationException
Constructs an InvalidDestinationException with the specified reason.

InvalidModuleException - Exception in javax.enterprise.deploy.spi.exceptions
This exception is to report an invalid J2EE deployment module type.

InvalidModuleException(String) - Constructor for exception javax.enterprise.deploy.spi.exceptions.InvalidModuleException
Creates a new InvalidModuleException object.

InvalidPropertyException - Exception in javax.resource.spi
This exception is thrown to indicate invalid configuration property settings.
InvalidPropertyException() - Constructor for exception
javax.resource.spi.InvalidPropertyException
Create a InvalidPropertyException.
InvalidPropertyException(String) - Constructor for exception
javax.resource.spi.InvalidPropertyException
Create a InvalidPropertyException.
InvalidPropertyException(Throwables) - Constructor for exception
javax.resource.spi.InvalidPropertyException
Constructs a new Throwable with the specified cause.
InvalidPropertyException(String, Throwable) - Constructor for exception
javax.resource.spi.InvalidPropertyException
Constructs a new Throwable with the specified detail message and cause.
InvalidPropertyException(String, String) - Constructor for exception
javax.resource.spi.InvalidPropertyException
Constructs a new Throwable with the specified detail message and an error code.
InvalidRequestException - Exception in javax.xml.registry
This exception is thrown when a JAXR client attempts to invoke an API method that is not valid for some reason.
InvalidRequestException() - Constructor for exception
javax.xml.registry.InvalidRequestException
Constructs a JAXRException object with no reason or embedded Throwable.
InvalidRequestException(String) - Constructor for exception
javax.xml.registry.InvalidRequestException
Constructs a JAXRException object with the given String as the reason for the exception being thrown.
InvalidRequestException(String, Throwable) - Constructor for exception
javax.xml.registry.InvalidRequestException
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.
InvalidRequestException(Throwables) - Constructor for exception
javax.xml.registry.InvalidRequestException
Constructs a JAXRException object initialized with the given Throwable object.
InvalidSelectorException - Exception in javax.jms
This exception must be thrown when a JMS client attempts to give a
provider a message selector with invalid syntax.

**InvalidSelectorException(String, String)** - Constructor for exception

```java
javax.jms.InvalidSelectorException
```

**InvalidSelectorException** - Constructor for exception

```java
javax.jms.InvalidSelectorException
```

This exception indicates that the request carried an invalid transaction context.

**InvalidTransactionException()** - Constructor for exception

```java
javax.transaction.InvalidTransactionException
```

**InvalidTransactionException(String)** - Constructor for exception

```java
javax.transaction.InvalidTransactionException
```

**InvocationContext** - Interface in `javax.interceptor`

Context information passed to AroundInvoke and Interceptor-class lifecycle callback methods.

**invoke(ELContext, Object[])** - Method in class

```java
javax.el.MethodExpression
```

If a String literal is specified as the expression, returns the String literal coerced to the expected return type of the method signature.

**invoke(FacesContext, Object[])** - Method in class

```java
javax.faces.el.MethodBinding
```

Deprecated. Return the return value (if any) resulting from a call to the method identified by this method binding expression, passing it the specified parameters, relative to the specified `FacesContext`.

**invoke(ObjectName, String, Object[], String[])** - Method in interface

```java
javax.management.j2ee.Management
```

Invokes an operation on a managed object.

**invoke(Writer)** - Method in class `javax.servlet.jsp.tagext.JspFragment`

Executes the fragment and directs all output to the given Writer, or the JspWriter returned by the getOut() method of the JspContext associated with the fragment if out is null.

**invoke(Object[])** - Method in interface `javax.xml.rpc.Call`

Invokes a specific operation using a synchronous request-response interaction mode.
**invoke(QName, Object[])** - Method in interface javax.xml.rpc.Call
Invokes a specific operation using a synchronous request-response interaction mode.

**invoke(T)** - Method in interface javax.xml.ws.Dispatch
Invoke a service operation synchronously.

**invoke(T)** - Method in interface javax.xml.ws.Provider
Invokes an operation according to the contents of the request message.

**INVOKE_APPLICATION** - Static variable in class javax.faces.event.PhaseId
Identifier that indicates an interest in events queued for the Invoke Application phase of the request processing lifecycle.

**invokeAsync(T)** - Method in interface javax.xml.ws.Dispatch
Invoke a service operation asynchronously.

**invokeAsync(T, AsyncHandler<T>)** - Method in interface javax.xml.ws.Dispatch
 invoke a service operation asynchronously.

**invokeContextCallback(FacesContext, UIComponent)** - Method in interface javax.faces.component.ContextCallback
This method will be called by an implementation of
UIComponent.invokeOnComponent(javax.faces.context.FacesContext, java.lang.String, javax.faces.component.ContextCallback) and must be passed the component with the clientId given as an argument to invokeOnComponent.

**invokeOnComponent(FacesContext, String, ContextCallback)** - Method in class javax.faces.component.UICOMPONENT
Starting at this component in the View hierarchy, search for a component with a clientId equal to the argument clientId and, if found, call the
ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UICOMPONENT) method on the argument callback, passing the current FacesContext and the found component as arguments.

**invokeOnComponent(FacesContext, String, ContextCallback)** - Method in class javax.faces.component.UICOMPONENTBase
Starting at this component in the View hierarchy, search for a component with a clientId equal to the argument clientId and, if found, call the
ContextCallback.invokeContextCallback(javax.faces.context.FacesContext, javax.faces.component.UIComponent) method on the argument callback, passing the current FacesContext and the found component as arguments.

invokeOnComponent(FacesContext, String, ContextCallback) - Method in class javax.faces.component.UIData
Override behavior from UIComponentBase.invokeOnComponent(javax.faces.context.FacesContext, java.lang.String, javax.faces.component.ContextCallback) to provide special care for positioning the data properly before finding the component and invoking the callback on it.

invokeOneWay(Object[]) - Method in interface javax.xml.rpc.Call
Invokes a remote method using the one-way interaction mode.

invokeOneWay(T) - Method in interface javax.xml.ws.Dispatch
Invokes a service operation using the one-way interaction mode.

IS_COALEScing - Static variable in class javax.xml.stream.XMLInputFactory
The property that requires the parser to coalesce adjacent character data sections

IS_NAMESPACE_AWARE - Static variable in class javax.xml.stream.XMLInputFactory
The property used to turn on/off namespace support, this is to support XML 1.0 documents, only the true setting must be supported

IS_REPAIRING_NAMESPACES - Static variable in class javax.xml.stream.XMLOutputFactory
Property used to set prefix defaulting on the output side

IS_REPLACING_ENTITY_REFERENCES - Static variable in class javax.xml.stream.XMLInputFactory
Requires the parser to replace internal entity references with their replacement text and report them as characters

IS_SUPPORTING_EXTERNAL_ENTITIES - Static variable in class javax.xml.stream.XMLInputFactory
The property that requires the parser to resolve external parsed entities

IS_VALIDATING - Static variable in class javax.xml.stream.XMLInputFactory
The property used to turn on/off implementation specific validation

isActive() - Method in interface javax.persistence.EntityTransaction
Indicate whether a transaction is in progress.

**isAddEvent()** - Method in class javax.enterprise.deploy.model.XPathEvent

Is this an add event?

**isAppropriateListener(FacesListener)** - Method in class javax.faces.event.ActionEvent

**isAppropriateListener(FacesListener)** - Method in class javax.faces.event.FacesEvent

Return true if this FacesListener is an instance of a listener class that this event supports.

**isAppropriateListener(FacesListener)** - Method in class javax.faces.event.ValueChangeEvent

**isAttribute()** - Method in interface javax.xml.stream.events.XMLEvent

A utility function to check if this event is an Attribute.

**isAttributeSpecified(int)** - Method in class javax.xml.stream.util.StreamReaderDelegate

Returns a boolean which indicates if this attribute was created by default.

**isAutoFlush()** - Method in class javax.servlet.jsp.JspWriter

This method indicates whether the JspWriter is autoFlushing.

**isAvailable()** - Method in exception javax.xml.registry.JAXRException

Returns true if a response is available, false otherwise.

**isAvailable()** - Method in interface javax.xml.registry.JAXRResponse

Returns true if a response is available, false otherwise.

**isCallerInRole(Identity)** - Method in interface javax.ejb.EJBContext

Deprecated. Use boolean isCallerInRole(String roleName) instead.

**isCallerInRole(String)** - Method in interface javax.ejb.EJBContext

Test if the caller has a given security role.

**isCancelSupported()** - Method in interface javax.enterprise.deploy.spi.status.ProgressObject

Tests whether the vendor supports a cancel operation for deployment activities.

**isCData()** - Method in interface javax.xml.stream.events.Characters

Returns true if this is a CData section.
**isChangeEvent()** - Method in class javax.enterprise.deploy.model.XpathEvent
Is this a change event?

**isCharacters()** - Method in interface javax.xml.stream.events.XMLEvent
A utility function to check if this event is Characters.

**isCharacters()** - Method in class javax.xml.stream.util.StreamReaderDelegate

**isCharacters()** - Method in interface javax.xml.stream.XMLStreamReader
Returns true if the cursor points to a character data event

**isClosed()** - Method in interface javax.xml.registry.Connection
Indicated whether this Connection has been closed or not.

**isComment()** - Method in interface javax.xml.soap.Text
Retrieves whether this Text object represents a comment.

**isCommitted()** - Method in interface javax.servlet.ServletResponse
Returns a boolean indicating if the response has been committed.

**isCommitted()** - Method in class javax.servlet.ServletResponseWrapper
The default behavior of this method is to return isCommitted() on the wrapped response object.

**isComplete()** - Method in class javax.mail.internet.MimeMultipart
Return true if the final boundary line for this multipart was seen.

**isCompleted()** - Method in interface javax.enterprise.deploy.spi.status.DeploymentStatus
A convenience method to report if the operation is in the completed state.

**isConfirmed()** - Method in interface javax.xml.registry.infomodel.Association
Determines whether an Association has been confirmed completely.

**isConfirmedBySourceOwner()** - Method in interface javax.xml.registry.infomodel.Association
Determines whether an Association has been confirmed by the owner of the source object.

**isConfirmedByTargetOwner()** - Method in interface javax.xml.registry.infomodel.Association
Determines whether an Association has been confirmed by the owner of the target object.

**isConnected()** - Method in class javax.mail.Service
Is this service currently connected?
**isDataFlavorSupported(DataFlavor)** - Method in class javax.activation.DataHandler

Returns whether the specified data flavor is supported for this object.

**isDConfigBeanVersionSupported(DConfigBeanVersionType)** - Method in interface javax.enterprise.deploy.spi.DeploymentManager

Returns 'true' if the configuration beans support the J2EE platform version specified.

**isDefaultNamespaceDeclaration()** - Method in interface javax.xml.stream.events.Namespace

Returns true if this attribute declares the default namespace

**isDeferredMethod()** - Method in class javax.servlet.jsp.tagext.TagAttributeInfo

Returns true if this attribute is to be passed a MethodExpression so that expression evaluation can be deferred.

**isDeferredValue()** - Method in class javax.servlet.jsp.tagext.TagAttributeInfo

Returns true if this attribute is to be passed a ValueExpression so that expression evaluation can be deferred.

**isDeliveryTransacted(Method)** - Method in interface javax.resource.spi.endpoint.MessageEndpointFactory

This is used to find out whether message deliveries to a target method on a message listener interface that is implemented by a message endpoint will be transacted or not.

**isDisabled()** - Method in class javax.faces.component.html.HtmlCommandButton

Return the value of the disabled property.

**isDisabled()** - Method in class javax.faces.component.html.HtmlCommandLink

Return the value of the disabled property.

**isDisabled()** - Method in class javax.faces.component.html.HtmlInputSecret

Return the value of the disabled property.

**isDisabled()** - Method in class javax.faces.component.html.HtmlInputText

Return the value of the disabled property.

**isDisabled()** - Method in class javax.faces.component.html.HtmlInputTextarea

Return the value of the disabled property.
javax.faces.component.html.HtmlOutputLink
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectManyListbox
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectManyMenu
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectOneListbox
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectOneMenu
   Return the value of the disabled property.
   isDisabled() - Method in class
javax.faces.component.html.HtmlSelectOneRadio
   Return the value of the disabled property.
   isDisabled() - Method in class javax.faces.model.SelectItem
   Return the disabled flag for this item, which should modify the rendered output to make this item unavailable for selection by the user if set to true.
   isElement(Object) - Method in class javax.xml.bind.JAXBIntrospector
   Return true iff object represents a JAXB element.
   isEmpty() - Method in class javax.activation.MimeTypeParameterList
   Determine whether or not this list is empty.
   isEndElement() - Method in interface
javax.xml.stream.events.XMLEvent
   A utility function to check if this event is an EndDocument.
   isEndElement() - Method in interface
javax.xml.stream.events.XMLEvent
   A utility function to check if this event is aEndElement.
   isEndElement() - Method in class
javax.xml.stream.util.StreamReaderDelegate
**isEndElement()** - Method in interface javax.xml.stream.XMLStreamReader

Returns true if the cursor points to an end tag (otherwise false)

**isEntityReference()** - Method in interface javax.xml.stream.events.XMLEvent

A utility function to check if this event is an EntityReference.

**isEscape()** - Method in class javax.faces.component.html.HtmlOutputFormat

Return the value of the escape property.

**isEscape()** - Method in class javax.faces.component.html.HtmlOutputLabel

Return the value of the escape property.

**isEscape()** - Method in class javax.faces.component.html.HtmlOutputText

Return the value of the escape property.

**isExpunged()** - Method in class javax.mail.Message

Checks whether this message is expunged.

**isExternal()** - Method in interface javax.xml.registry.infomodel.Classification

Returns true if this is an external classification.

**isExternal()** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Determines whether this ClassificationScheme is an external ClassificationScheme or an internal ClassificationScheme.

**isExtramural()** - Method in interface javax.xml.registry.infomodel.Association

Determines whether an Association is extramural or not.

**isFailed()** - Method in interface javax.enterprise.deploy.spi.status.DeploymentStatus

A convenience method to report if the operation is in the failed state.

**isFormattedOutput()** - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl

Convenience method for getting the formatted output flag.

**isFragment()** - Method in class javax.servlet.jsp.tagext.TagAttributeInfo

Whether this attribute is of type JspFragment.

**isFragment()** - Method in class
javax.xml.bind.helpers.**AbstractMarshallerImpl**
Convenience method for getting the fragment flag.

**isGlobalOnly()** - Method in class javax.faces.component.**UIMessages**
Return the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered.

**isGlobalScope()** - Method in class javax.xml.bind.**JAXBElement**
Returns true iff this xml element declaration is global.

**isGroup()** - Method in class javax.mail.internet.**InternetAddress**
Indicates whether this address is an RFC 822 group address.

**isGroupingUsed()** - Method in class javax.faces.convert.**NumberConverter**
Return true if getAsString should include grouping separators if necessary.

**isIdentical(EJBLocalObject)** - Method in interface javax.ejb.**EJBLocalObject**
Test if a given EJB local object is identical to the invoked EJB local object.

**isIdentical(EJBOBJECT)** - Method in interface javax.ejb.**EJBObject**
Test if a given EJB object is identical to the invoked EJB object.

**isIgnorableWhiteSpace()** - Method in interface javax.xml.stream.events.**Characters**
Return true if this is ignorableWhiteSpace.

**isImmediate()** - Method in interface javax.faces.component.**ActionSource**
Return a flag indicating that the default **ActionListener** provided by the JavaServer Faces implementation should be executed immediately (that is, during **Apply Request Values** phase of the request processing lifecycle), rather than waiting until the **Invoke Application** phase.

**isImmediate()** - Method in interface javax.faces.component.**EditableValueHolder**
Return the "immediate" state for this component.

**isImmediate()** - Method in class javax.faces.component.**UIComponent**

**isIntegerOnly()** - Method in class javax.faces.convert.**NumberConverter**
Return true if only the integer portion of the given value should be returned from getAsObject().
**isIsmap()** - Method in class java.util.LinkedHashMap. **HtmlGraphicImage**
Return the value of the ismap property.

**isItemDisabled()** - Method in class javax.faces.component.UISelectItem
Return the disabled setting for this selection item.

**isItemEscaped()** - Method in class javax.faces.component.UISelectItem
Return the escape setting for the label of this selection item.

**isLiteralText()** - Method in class javax.el.Expression
Returns whether this expression was created from only literal text.

**isLocaleSupported(Locale)** - Method in interface javax.enterprise.deploy.spi.DeploymentManager
Reports if this implementation supports the designated locale.

**isLocalValueSet()** - Method in interface javax.faces.component.EditableValueHolder
Return the "local value set" state for this component.

**isLocalValueSet()** - Method in class javax.faces.component.UIInput
Return the "local value set" state for this component.

**isMimeType(String)** - Method in class javax.mail.internet.MimeBodyPart
Is this Part of the specified MIME type?

**isMimeType(String)** - Method in class javax.mail.internet.MimeMessage
Is this Part of the specified MIME type?

**isMimeType(String)** - Method in interface javax.mail.Part
Is this Part of the specified MIME type?

**isMimeTypeEqual(String)** - Method in class javax.activation.ActivationDataFlavor
Is the string representation of the MIME type passed in equivalent to the MIME type of this DataFlavor.

**isMTOMEnabled()** - Method in interface javax.xml.ws.soap.SOAPBinding
Returns true if the use of MTOM is enabled.

**isNamespace()** - Method in interface javax.xml.stream.events.XMLEvent
A utility function to check if this event is a Namespace.

**isNew()** - Method in interface javax.servlet.http.HttpServletRequest
Returns true if the client does not yet know about the session or if the client chooses not to join the session.

**isNil()** - Method in class javax.xml.bind.JAXBElement
Returns true iff this element instance content model is nil.

**isOpaque()** - Method in interface javax.xml.registry.infomodel.ExtrinsicObject
Determines whether the ExtrinsicObject is opaque (not readable) by the registry operator.

**isOpen()** - Method in class `javax.mail.Folder`
Indicates whether this Folder is in the 'open' state.

**isOpen()** - Method in interface `javax.persistence.EntityManager`
Determine whether the EntityManager is open.

**isOpen()** - Method in interface `javax.persistence.EntityManagerFactory`
Indicates whether or not this factory is open.

**isParameterAndReturnSpecRequired(QName)** - Method in interface `javax.xml.rpc.Call`
Indicates whether `addParameter` and `setReturnType` methods are to be invoked to specify the parameter and return type specification for a specific operation.

**isPartialResponse()** - Method in interface `javax.xml.registry.BulkResponse`
Determines whether the response is a partial response due to large result set.

**isPermanent()** - Method in exception `javax.servlet.UnavailableException`
Returns a boolean indicating whether the servlet is permanently unavailable.

**isPostback(FacesContext)** - Method in class `javax.faces.render.ResponseStateManager`
Return true if the current request is a postback.

**isPrependId()** - Method in class `javax.faces.component.UIForm`

**isProcessingInstruction()** - Method in interface `javax.xml.stream.events.XMLEvent`
A utility function to check if this event is a ProcessingInstruction.

**isPropertyResolved()** - Method in class `javax.el.ELContext`
Returns whether an `ELResolver` has successfully resolved a given (base, property) pair.

**isPropertySupported(String)** - Method in class `javax.xml.stream.XMLInputFactory`
Query the set of properties that this factory supports.

**isPropertySupported(String)** - Method in class `javax.xml.stream.XMLOutputFactory`
Query the set of properties that this factory supports.

**isPublished()** - Method in class `javax.xml.ws.Endpoint`
Returns true if the endpoint is in the published state.
isReadOnly(ELContext, Object, Object) - Method in class javax.el.ArrayELResolver
If the base object is a Java language array, returns whether a call to

isReadOnly() - Method in class javax.el.BeanELResolver.BeanProperty

isReadOnly(ELContext, Object, Object) - Method in class javax.el.BeanELResolver
If the base object is not null, returns whether a call to

isReadOnly(ELContext, Object, Object) - Method in class javax.el.CompositeELResolver
For a given base and property, attempts to determine whether a call to

isReadOnly(ELContext, Object, Object) - Method in class javax.el.EELResolver
For a given base and property, attempts to determine whether a call to

isReadOnly(ELContext, Object, Object) - Method in class javax.el.ListELResolver
If the base object is a list, returns whether a call to

isReadOnly(ELContext, Object, Object) - Method in class javax.el.MapELResolver
If the base object is a map, returns whether a call to

isReadOnly(ELContext, Object, Object) - Method in class javax.el ResourceBundleELResolver
If the base object is not null and an instance of ResourceBundle, return true.

isReadOnly(ELContext) - Method in class javax.el.ValueExpression
Evaluates the expression relative to the provided context, and returns true if a call to
ValueExpression.setValue(javax.el.ELContext, java.lang.Object) will always fail.

**isReadonly()** - Method in class
javax.faces.component.html.HtmlCommandButton
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlInputSecret
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlInputText
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlInputTextarea
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
```
Return the value of the readonly property.
```

**isReadonly()** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
```
Return the value of the readonly property.
```

**isReadOnly(Object, Object)** - Method in class
javax.faces.el.PropertyResolver
```
Deprecated. Checks if the specified property is read-only.
```

**isReadOnly(Object, int)** - Method in class
isReadOnly(FacesContext) - Method in class javax.faces.el.ValueBinding
  Deprecated. Return true if the specified property of the specified property is known to be immutable; otherwise, return false.

isReadOnly(ELContext, Object, Object) - Method in class javax.servlet.jsp.el.ImplicitObjectELResolver
  If the base object is null, and the property matches the name of a JSP implicit object, returns true to indicate that implicit objects cannot be overwritten.

isReadOnly(ELContext, Object, Object) - Method in class javax.servlet.jsp.el.ScopedAttributeELResolver
  If the base object is null, returns false to indicate that scoped attributes are never read-only.

isRedeploySupported() - Method in interface javax.enterprise.deploy.spi.DeploymentManager
  This method designates whether this platform vendor provides application redeployment functionality.

isRedisplay() - Method in class javax.faces.component.html.HtmlInputSecret
  Return the value of the redisplay property.

isRegistered(ObjectName) - Method in interface javax.management.j2ee.Management
  Checks whether a managed object, identified by its object name, is already registered with the MEJB.

isRegistered(Class, QName) - Method in interface javax.xml.rpc.encoding.TypeMapping
  Checks whether or not type mapping between specified XML type and Java type is registered.

isRemoved() - Method in class javax.mail.event.MessageCountEvent
  Indicates whether this event is the result of an explicit expunge by this client, or due to an expunge from external sources.

isRemoveEvent() - Method in class javax.enterprise.deploy.model.XpathEvent
  Is this a remove event?

isRendered() - Method in class javax.faces.component.UICOMPONENT
  Return true if this component (and its children) should be rendered during the Render Response phase of the request processing
lifecycle.

**isRendered()** - Method in class javax.faces.component.UIComponentBase

**isRequestedSessionIdFromCookie()** - Method in interface javax.servlet.http.HttpServletRequest

Checks whether the requested session ID came in as a cookie.

**isRequestedSessionIdFromCookie()** - Method in class javax.servlet.http.HttpServletRequestWrapper

The default behavior of this method is to return isRequestedSessionIdFromCookie() on the wrapped request object.

**isRequestedSessionIdFromUrl()** - Method in interface javax.servlet.http.HttpServletRequest

Checks whether the requested session ID came in as part of the request URL.

**isRequestedSessionIdFromUrl()** - Method in interface javax.servlet.http.HttpServletRequestWrapper

Deprecated. As of Version 2.1 of the Java Servlet API, use HttpServletRequest.isRequestedSessionIdFromURL() instead.

**isRequestedSessionIdFromUrl()** - Method in class javax.servlet.http.HttpServletRequestWrapper

The default behavior of this method is to return isRequestedSessionIdFromUrl() on the wrapped request object.

**isRequestedSessionIdValid()** - Method in interface javax.servlet.http.HttpServletRequest

Checks whether the requested session ID is still valid.

**isRequestedSessionIdValid()** - Method in class javax.servlet.http.HttpServletRequestWrapper

The default behavior of this method is to return isRequestedSessionIdValid() on the wrapped request object.

**isRequired()** - Method in interface javax.faces.component.EdtiableValueHolder

Return the "required field" state for this component.

**isRequired()** - Method in class javax.faces.component.UIInput

Return the "required field" state for this component.
**isRequired()** - Method in class `javax.servlet.jsp.tagext.TagAttributeInfo`  
Whether this attribute is required.

**isRowAvailable()** - Method in class `javax.faces.component.UIData`  
Return a flag indicating whether there is `rowData` available at the current `rowIndex`.

**isRowAvailable()** - Method in class `javax.faces.model.ListDataModel`  
Return true if there is `wrappedData` available, and the current value of `rowIndex` is greater than or equal to zero, and less than the size of the list.

**isRowAvailable()** - Method in class `javax.faces.model.ResultDataModel`  
Return true if there is `wrappedData` available, and the current value of `rowIndex` is greater than or equal to zero, and less than the length of the array returned by calling `getRows()` on the underlying `Result`.

**isRowAvailable()** - Method in class `javax.faces.model.ResultSetDataModel`  
Return true if there is `wrappedData` available, and the result of calling `absolute()` on the underlying `ResultSet`, passing the current value of `rowIndex` plus one (to account for the fact that `ResultSet` uses one-relative indexing), returns true.

**isSameRM(XAResource)** - Method in interface `javax.transaction.xa.XAResource`  
This method is called to determine if the resource manager instance represented by the target object is the same as the resource manager instance represented by the parameter `xares`.

**isSavingStateInClient(FacesContext)** - Method in class `javax.faces.application.StateManager`
**isSavingStateInClient(FacesContext)** - Method in class javax.faces.application.StateManagerWrapper

The default behavior of this method is to call `StateManager.isSavingStateInClient(javax.faces.context.FacesContext)` on the wrapped `StateManager` object.

**isSecure()** - Method in interface javax.servlet.ServletRequest

Returns a boolean indicating whether this request was made using a secure channel, such as HTTPS.

**isSecure()** - Method in class javax.servlet.ServletRequestWrapper

The default behavior of this method is to return isSecure() on the wrapped request object.

**isSelected()** - Method in class javax.faces.component.UISelectBoolean

Return the local value of the selected state of this component.

**isSession()** - Method in interface javax.ejb.EJBMetaData

Test if the enterprise Bean's type is "session".

**isSet(Flags.Flag)** - Method in class javax.mail.internet.MimeMessage

Check whether the flag specified in the `flag` argument is set in this message.

**isSet(Flags.Flag)** - Method in class javax.mail.Message

Check whether the flag specified in the `flag` argument is set in this message.

**isShowDetail()** - Method in class javax.faces.component.UIMessage

Return the flag indicating whether the `detail` property of the associated message(s) should be displayed.

**isShowDetail()** - Method in class javax.faces.component.UIMessages

Return the flag indicating whether the `detail` property of the associated message(s) should be displayed.

**isShowSummary()** - Method in class javax.faces.component.UIMessage

Return the flag indicating whether the `summary` property of the associated message(s) should be displayed.

**isShowSummary()** - Method in class javax.faces.component.UIMessages

Return the flag indicating whether the `summary` property of the associated message(s) should be displayed.

**isSpecified()** - Method in interface javax.xml.stream.events.Attribute

A flag indicating whether this attribute was actually specified in the start-tag of its element, or was defaulted from the schema.

**isStandalone()** - Method in interface
javax.xml.stream.events.\texttt{StartDocument}
Returns if this XML is standalone
\texttt{isStandalone()} - Method in class
javax.xml.stream.util.\texttt{StreamReaderDelegate}

\texttt{isStandalone()} - Method in interface
javax.xml.stream.\texttt{XMLStreamReader}
Get the standalone declaration from the xml declaration
\texttt{isStartDocument()} - Method in interface
javax.xml.stream.events.\texttt{XMLEvent}
A utility function to check if this event is a StartDocument.
\texttt{isStartElement()} - Method in interface
javax.xml.stream.events.\texttt{XMLEvent}
A utility function to check if this event is a StartElement.
\texttt{isStartElement()} - Method in class
javax.xml.stream.util.\texttt{StreamReaderDelegate}

\texttt{isStartElement()} - Method in interface
javax.xml.stream.\texttt{XMLStreamReader}
Returns true if the cursor points to a start tag (otherwise false)
\texttt{isStatelessSession()} - Method in interface javax.ejb.\texttt{EJBMetaData}
Test if the enterprise Bean's type is "stateless session".
\texttt{isStopSupported()} - Method in interface
javax.enterprise.deploy.spi.status.\texttt{ProgressObject}
Tests whether the vendor supports a stop operation for deployment activities.
\texttt{isSubmitted()} - Method in class javax.faces.component.\texttt{UIForm}
Returns the current value of the \texttt{submitted} property.
\texttt{isSubscribed()} - Method in class javax.mail.\texttt{Folder}
Returns true if this Folder is subscribed.
\texttt{isSuppressed()} - Method in class javax.faces.webapp.\texttt{UIComponentTag}
Deprecated.
\texttt{isSynchronous()} - Method in interface javax.xml.registry.\texttt{Connection}
Indicates whether a client uses synchronous communication with JAXR provider or not.
\texttt{isTooltip()} - Method in class javax.faces.component.html.\texttt{HtmlMessage}
Return the value of the \texttt{tooltip} property.
\texttt{isTooltip()} - Method in class javax.faces.component.html.\texttt{HtmlMessages}
Return the value of the \texttt{tooltip} property.
**isTransactional()** - Method in interface javax.management.j2ee.statistics.JMSConnectionStats
  Returns the transactional state of this JMS connection.

**isTransient()** - Method in interface javax.faces.component.StateHolder
  If true, the Object implementing this interface must not participate in state saving or restoring.

**isTransient()** - Method in class javax.faces.component.UIComponentBase

**isTransient()** - Method in class javax.faces.events.MethodExpressionActionListener

**isTransient()** - Method in class javax.faces.events.MethodExpressionValueChangeListener

**isTransient()** - Method in class javax.faces.validator.DoubleRangeValidator

**isTransient()** - Method in class javax.faces.validator.LengthValidator

**isTypeSubstituted()** - Method in class javax.xml.bind.JAXBElement
  Returns true iff this xml element instance's value has a different type than xml element declaration's declared type.

**isUserInRole(String)** - Method in class javax.faces.context.ExternalContext
  Return true if the currently authenticated user is included in the specified role.

**isUserInRole(String)** - Method in interface
javax.servlet.http.HttpServletRequest
Returns a boolean indicating whether the authenticated user is included in the specified logical "role".

isUserInRole(String) - Method in class javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to return isUserInRole(String role) on the wrapped request object.

isUserInRole(String) - Method in interface javax.xml.rpc.server.ServletEndpointContext
Returns a boolean indicating whether the authenticated user for the current method invocation on the endpoint instance is included in the specified logical "role".

isUserInRole(String) - Method in interface javax.xml.ws.WebServiceContext
Returns a boolean indicating whether the authenticated user is included in the specified logical role.

isValid() - Method in interface javax.faces.component.EditableValueHolder
Return a flag indicating whether the local value of this component is valid (no conversion error has occurred).

isValid() - Method in class javax.faces.component.UIInput

isValid(TagData) - Method in class javax.servlet.jsp.tagext.TagExtraInfo
Translation-time validation of the attributes.

isValid(TagData) - Method in class javax.servlet.jsp.tagext.TagInfo
Translation-time validation of the attributes.

isValidating() - Method in class javax.xml.bind.helpers.AbstractUnmarshallerImpl
Indicates whether or not the Unmarshaller is configured to validate during unmarshal operations.

isValidating() - Method in interface javax.xml.bindUnmarshaller
Deprecated. since JAXB2.0, please see Unmarshaller.getSchema()

isValiding() - Method in interface javax.xml.bind.Unmarshaller

isValueReference(String) - Static method in class javax.faces.webapp.UIComponentTag
Deprecated. Return true if the specified value conforms to the syntax requirements of a value binding expression.

isWhiteSpace(char) - Static method in class javax.xml.bind.annotation.adapters.CollapsedStringAdapter
returns true if the specified char is a white space character.
isWhiteSpace() - Method in interface javax.xml.stream.events.Characters
Returns true if this set of Characters is all whitespace.

isWhiteSpace() - Method in class javax.xml.stream.util.StreamReaderDelegate

isWhiteSpace() - Method in interface javax.xml.stream.XMLStreamReader
Returns true if the cursor points to a character data event that consists of all whitespace.

isWhiteSpaceExceptSpace(char) - Static method in class javax.xml.bind.annotation.adapters.NormalizedStringAdapter
Returns true if the specified char is a white space character but not 0x20.

isXOPPackage() - Method in class javax.xml.bind.attachment.AttachmentMarshaller
Read-only property that returns true if JAXB marshaller should enable XOP creation.

isXOPPackage() - Method in class javax.xml.bind.attachment.AttachmentUnmarshaller
Read-only property that returns true if JAXB unmarshaller needs to perform XOP processing.

itemExists(String) - Method in interface javax.jms.MapMessage
Indicates whether an item exists in this MapMessage object.

IterationTag - Interface in javax.servlet.jsp.tagext
The IterationTag interface extends Tag by defining one additional method that controls the reevaluation of its body.
**J2eeApplicationObject** - Interface in `javax.enterprise.deploy.model`
J2eeApplicationObject is an interface that represents a J2EE application (EAR); it maintains a DeployableObject for each module in the archive.

**javaCharset(String)** - Static method in class `javax.mail.internet.MimeUtility`
Convert a MIME charset name into a valid Java charset name.

**JavaMailStats** - Interface in `javax.management.j2ee.statistics`
Specifies the statistics provided by a JavaMail resource.

**javax.activation** - package `javax.activation`
The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data.

**javax.annotation** - package `javax.annotation`

**javax.annotation.security** - package `javax.annotation.security`

**javax.ejb** - package `javax.ejb`
The `javax.ejb` package contains the Enterprise JavaBeans classes and interfaces that define the contracts between the enterprise bean and its clients and between the enterprise bean and the EJB container.

**javax.ejb.spi** - package `javax.ejb.spi`
The `javax.ejb.spi` package defines interfaces that are implemented by the EJB container.

**javax.el** - package `javax.el`
Provides the API for the **Unified Expression Language** shared by the JSP 2.1 and JSF 1.2 technologies.

**javax.enterprise.deploy.model** - package `javax.enterprise.deploy.model`
Provides Tool Vendor implementation classes.

**javax.enterprise.deploy.model.exceptions** - package `javax.enterprise.deploy.model.exceptions`
Provides Tool Vendor exception implementation classes.

**javax.enterprise.deploy.shared** - package `javax.enterprise.deploy.shared`
Provides shared objects for Tool Vendor and Product Vendor
implementation classes.

**javax.enterprise.deploy.shared.factories** - package javax.enterprise.deploy.shared.factories
Provides shared factory manager object for Tool Vendor and Product Vendor implementation classes.

**javax.enterprise.deploy.spi** - package javax.enterprise.deploy.spi
Provides J2EE Product Vendor implementation classes.

**javax.enterprise.deploy.spi.exceptions** - package javax.enterprise.deploy.spi.exceptions
Provides J2EE Product Vendor deployment exception implementation classes.

**javax.enterprise.deploy.spi.factories** - package javax.enterprise.deploy.spi.factories
Provides J2EE Product Vendor deployment factory implementation classes.

**javax.enterprise.deploy.spi.status** - package javax.enterprise.deploy.spi.status
Provides J2EE Product Vendor deployment status implementation classes.

**javax.faces** - package javax.faces
Top level classes for the JavaServer(tm) Faces API.

**javax.faces.application** - package javax.faces.application
APIs that are used to link an application's business logic objects to JavaServer Faces, as well as convenient pluggable mechanisms to manage the execution of an application that is based on JavaServer Faces.

**javax.faces.component** - package javax.faces.component
Fundamental APIs for user interface components.

**javax.faces.component.html** - package javax.faces.component.html
Specialized user interface component classes for HTML.

**javax.faces.context** - package javax.faces.context
Classes and interfaces defining per-request state information.

**javax.faces.convert** - package javax.faces.convert
Contains classes and interfaces defining converters.

**javax.faces.el** - package javax.faces.el
DEPRECATED Classes and interfaces for evaluating and processing reference expressions.

**javax.faces.event** - package javax.faces.event
Interfaces describing events and event listeners, and concrete event
implementation classes.

**javax.faces.lifecycle** - package javax.faces.lifecycle

Classes and interfaces defining lifecycle management for the JavaServer Faces implementation.

**javax.faces.model** - package javax.faces.model

Standard model data beans for JavaServer Faces.

**javax.faces.render** - package javax.faces.render

Classes and interfaces defining the rendering model.

**javax.faces.validator** - package javax.faces.validator

Interface defining the validator model, and concrete validator implementation classes.

**javax.faces.webapp** - package javax.faces.webapp

Classes required for integration of JavaServer Faces into web applications, including a standard servlet, base classes for JSP custom component tags, and concrete tag implementations for core tags.

**javax.interceptor** - package javax.interceptor

The javax.interceptor package contains classes and interfaces for use with EJB interceptors.

**javax.jms** - package javax.jms

The Java Message Service (JMS) API provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages.

**javax.jws** - package javax.jws

**javax.jws.soap** - package javax.jws.soap

**javax.mail** - package javax.mail

The JavaMail™ API provides classes that model a mail system.

**javax.mail.event** - package javax.mail.event

Listeners and events for the JavaMail API.

**javax.mail.internet** - package javax.mail.internet

Classes specific to Internet mail systems.

**javax.mail.search** - package javax.mail.search

Message search terms for the JavaMail API.

**javax.mail.util** - package javax.mail.util

Utility classes.

**javax.management.j2ee** - package javax.management.j2ee

Provides the J2EE Management Enterprise Bean component
(MEJB) interfaces.

**javax.management.j2ee.statistics** - package javax.management.j2ee.statistics

Provides the standard interfaces for accessing performance data from J2EE managed objects. Package Specification [JSR 77, J2EE Management](http://www.jcp.org/en/jsr/detail?id=77) Related Documentation For overviews, tutorials, examples, guides, and tool documentation, please see: J2EE Tools

**javax.persistence** - package javax.persistence

The `javax.persistence` package contains the classes and interfaces that define the contracts between a persistence provider and the managed classes and the clients of the Java Persistence API.

**javax.persistence.spi** - package javax.persistence.spi

The `javax.persistence.spi` package defines the classes and interfaces that are implemented by the persistence provider and the Java EE container for use by the container, provider, and/or Persistence bootstrap class in deployment and bootstrapping.

**javax.resource** - package javax.resource

The `javax.resource` package is the top-level package for the J2EE Connector API specification.

**javax.resource.cci** - package javax.resource.cci

The `javax.resource.cci` package contains API specification for the Common Client Interface (CCI).

**javax.resource.spi** - package javax.resource.spi

The `javax.resource.spi` package contains APIs for the system contracts defined in the J2EE Connector Architecture specification.

**javax.resource.spi.endpoint** - package javax.resource.spi.endpoint

This package contains system contracts for service endpoint interactions.

**javax.resource.spi.security** - package javax.resource.spi.security

The `javax.resource.spi.security` package contains APIs for the security management contract.

**javax.resource.spi.work** - package javax.resource.spi.work

This package contains APIs for the work management contract.

**javax.security.jacc** - package javax.security.jacc

This package contains the Java Authorization Contract for Containers API

**javax.servlet** - package javax.servlet

The `javax.servlet` package contains a number of classes and interfaces that describe and define the contracts between a servlet
class and the runtime environment provided for an instance of such a class by a conforming servlet container.

**javax.servlet.http** - package javax.servlet.http
The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.

**javax.servlet.jsp** - package javax.servlet.jsp
Classes and interfaces for the Core JSP 2.1 API.

**javax.servlet.jsp.el** - package javax.servlet.jsp.el
Provides the ELResolver classes that define the object resolution rules that must be supported by a JSP container with the new unified Expression Language.

**javax.servlet.jsp.tagext** - package javax.servlet.jsp.tagext
Classes and interfaces for the definition of JavaServer Pages Tag Libraries.

**javax.transaction** - package javax.transaction
Provides the API that defines the contract between the transaction manager and the various parties involved in a distributed transaction namely: resource manager, application, and application server.

**javax.transaction.xa** - package javax.transaction.xa
Provides the API that defines the contract between the transaction manager and the resource manager, which allows the transaction manager to enlist and delist resource objects (supplied by the resource manager driver) in JTA transactions.

**javax.xml.bind** - package javax.xml.bind
Provides a runtime binding framework for client applications including unmarshalling, marshalling, and validation capabilities.

**javax.xml.bind.annotation** - package javax.xml.bind.annotation
Defines annotations for customizing Java program elements to XML Schema mapping.

**javax.xml.bind.annotation.adapters** - package javax.xml.bind.annotation.adapters
**XmlAdapter** and its spec-defined sub-classes to allow arbitrary Java classes to be used with JAXB.

**javax.xml.bind.attachment** - package javax.xml.bind.attachment
This package is implemented by a MIME-based package processor that enables the interpretation and creation of optimized binary data
within an MIME-based package format.

**javax.xml.bind.helpers** - package `javax.xml.bind.helpers`  
**JAXB Provider Use Only:** Provides partial default implementations for some of the `javax.xml.bind` interfaces.

**javax.xml.bind.util** - package `javax.xml.bind.util`  
Useful client utility classes.

**javax.xml.registry** - package `javax.xml.registry`  
This package and its sub-packages describe the API classes and interfaces for the JAXR API.

**javax.xml.registry.infomodel** - package `javax.xml.registry.infomodel`  
This package describes the information model for the JAXR API.

**javax.xml.rpc** - package `javax.xml.rpc`  
This package contains the core JAX-RPC APIs for the client programming model.

**javax.xml.rpc.encoding** - package `javax.xml.rpc.encoding`  
This package defines APIs for the extensible type mapping framework.

**javax.xml.rpc.handler** - package `javax.xml.rpc.handler`  
This package defines APIs for SOAP Message Handlers

**javax.xml.rpc.handler.soap** - package `javax.xml.rpc.handler.soap`  
This package defines APIs for SOAP Message Handlers

**javax.xml.rpc.holders** - package `javax.xml.rpc.holders`  
This package contains the standard Java Holder classes.

**javax.xml.rpc.server** - package `javax.xml.rpc.server`  
This package defines APIs for the servlet based JAX-RPC endpoint model.

**javax.xml.rpc.soap** - package `javax.xml.rpc.soap`  
This package defines APIs specific to the SOAP binding.

**javax.xml.soap** - package `javax.xml.soap`  
Provides the API for creating and building SOAP messages.

**javax.xml.stream** - package `javax.xml.stream`  

**javax.xml.stream.events** - package `javax.xml.stream.events`  

**javax.xml.stream.util** - package `javax.xml.stream.util`  

**javax.xml.ws** - package `javax.xml.ws`  
This package contains the core JAX-WS APIs.

**javax.xml.ws.handler** - package `javax.xml.ws.handler`
This package defines APIs for message handlers.

**javax.xml.ws.handler.soap** - package javax.xml.ws.handler.soap
This package defines APIs for SOAP message handlers.

This package defines APIs specific to the HTTP binding.

**javax.xml.ws.soap** - package javax.xml.ws.soap
This package defines APIs specific to the SOAP binding.

**javax.xml.ws.spi** - package javax.xml.ws.spi
This package defines SPIs for JAX-WS 2.0.

**JAXB_CONTEXT_FACTORY** - Static variable in class javax.xml.bind.JAXBContext
The name of the property that contains the name of the class capable of creating new JAXBContext objects.

**JAXB_ENCODING** - Static variable in interface javax.xml.bind.Marshaller
The name of the property used to specify the output encoding in the marshalled XML data.

**JAXB_FORMATTED_OUTPUT** - Static variable in interface javax.xml.bind.Marshaller
The name of the property used to specify whether or not the marshalled XML data is formatted with linefeeds and indentation.

**JAXB_FRAGMENT** - Static variable in interface javax.xml.bind.Marshaller
The name of the property used to specify whether or not the marshaller will generate document level events (ie calling startDocument or endDocument).

**JAXB_NO_NAMESPACE_SCHEMA_LOCATION** - Static variable in interface javax.xml.bind.Marshaller
The name of the property used to specify the xsi:noNamespaceSchemaLocation attribute value to place in the marshalled XML output.

**JAXB_SCHEMA_LOCATION** - Static variable in interface javax.xml.bind.Marshaller
The name of the property used to specify the xsi:schemaLocation attribute value to place in the marshalled XML output.

**JAXBContext** - Class in javax.xml.bind
The JAXBContext class provides the client's entry point to the JAXB API.

**JAXBContext()** - Constructor for class javax.xml.bind.JAXBContext
JAXBElement<T> - Class in javax.xml.bind
   JAXB representation of an Xml Element.
JAXBElement(QName, Class<T>, Class, T) - Constructor for class
   javax.xml.bind.JAXBElement
   Construct an xml element instance.
JAXBElement(QName, Class<T>, T) - Constructor for class
   javax.xml.bind.JAXBElement
   Construct an xml element instance.
JAXBElement.GlobalScope - Class in javax.xml.bind
   Designates global scope for an xml element.
JAXBElement.GlobalScope() - Constructor for class
   javax.xml.bind.JAXBElement.GlobalScope
JAXBException - Exception in javax.xml.bind
   This is the root exception class for all JAXB exceptions.
JAXBException(String) - Constructor for exception
   javax.xml.bind.JAXBException
   Construct a JAXBException with the specified detail message.
JAXBException(String, String) - Constructor for exception
   javax.xml.bind.JAXBException
   Construct a JAXBException with the specified detail message and
   vendor specific errorCode.
JAXBException(Throwabler) - Constructor for exception
   javax.xml.bind.JAXBException
   Construct a JAXBException with a linkedException.
JAXBException(String, Throwable) - Constructor for exception
   javax.xml.bind.JAXBException
   Construct a JAXBException with the specified detail message and
   linkedException.
JAXBException(String, String, Throwable) - Constructor for exception
   javax.xml.bind.JAXBException
   Construct a JAXBException with the specified detail message,
   vendor specific errorCode, and linkedException.
JAXBIntrospector - Class in javax.xml.bind
   Provide access to JAXB xml binding data for a JAXB object.
JAXBIntrospector() - Constructor for class
   javax.xml.bind.JAXBIntrospector
JAXBResult - Class in javax.xml.bind.util
JAXP Result implementation that unmarshals a JAXB object.

JAXBResult(JAXBContext) - Constructor for class javax.xml.bind.util.JAXBResult
Creates a new instance that uses the specified JAXBContext to unmarshal.

JAXBResult(Unmarshaller) - Constructor for class javax.xml.bind.util.JAXBResult
Creates a new instance that uses the specified Unmarshaller to unmarshal an object.

JAXBSource - Class in javax.xml.bind.util
JAXP Source implementation that marshals a JAXB-generated object.

JAXBSource(JAXBContext, Object) - Constructor for class javax.xml.bind.util.JAXBSource
Creates a new Source for the given content object.

JAXBSource(Marshaller, Object) - Constructor for class javax.xml.bind.util.JAXBSource
Creates a new Source for the given content object.

JAXRException - Exception in javax.xml.registry
Signals that a JAXR exception has occurred.

JAXRException() - Constructor for exception javax.xml.registry.JAXRException
Constructs a JAXRException object with no reason or embedded Throwable.

JAXRException(String) - Constructor for exception javax.xml.registry.JAXRException
Constructs a JAXRException object with the given String as the reason for the exception being thrown.

JAXRException(String, Throwable) - Constructor for exception javax.xml.registry.JAXRException
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

JAXRException(Throwerable) - Constructor for exception javax.xml.registry.JAXRException
Constructs a JAXRException object initialized with the given Throwable object.

JAXRPCException - Exception in javax.xml.rpc
The javax.xml.rpc.JAXRPCException is thrown from the core JAX-
RPC APIs to indicate an exception related to the JAX-RPC runtime mechanisms.

**JAXRPCException()** - Constructor for exception

```java
javax.xml.rpc.JAXRPCException
```

Constructs a new exception with null as its detail message.

**JAXRPCException(String)** - Constructor for exception

```java
javax.xml.rpc.JAXRPCException
```

Constructs a new exception with the specified detail message.

**JAXRPCException(String, Throwable)** - Constructor for exception

```java
javax.xml.rpc.JAXRPCException
```

Constructs a new exception with the specified detail message and cause.

**JAXRPCException(Throwables)** - Constructor for exception

```java
javax.xml.rpc.JAXRPCException
```

Constructs a new JAXRPCException with the specified cause and a detail message of (cause==null ?

**JAXRResponse** - Interface in `javax.xml.registry`

A JAXR requests' response.

**JAXWSPROVIDER_PROPERTY** - Static variable in class

```java
javax.xml.ws.spi.Provider
```

A constant representing the property used to lookup the name of a Provider implementation class.

**JCAConnectionPoolStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JCA Connection Pool

**JCAConnectionStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JCA connection

**JCASTats** - Interface in `javax.management.j2ee.statistics`

Specifies statistics provided by a JCA resource

**JDBCConnectionPoolStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JDBC connection pool.

**JDBCConnectionStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by all (pooled and non-pooled) JDBC connections.

**JDBCSStats** - Interface in `javax.management.j2ee.statistics`

Statistics provided by a JDBC resource

**JMSConnectionStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS connection

**JMSConsumerStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS message consumer

**JMSEndpointStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS message producer or a JMS message consumer.

**JMSEException** - Exception in `javax.jms`
This is the root class of all JMS API exceptions.

**JMSEException(String, String)** - Constructor for exception

```java
javax.jms.JMSEException
```

Constructs a JMSEException with the specified reason and error code.

**JMSEException(String)** - Constructor for exception

```java
javax.jms.JMSEException
```

Constructs a JMSEException with the specified reason and with the error code defaulting to null.

**JMSProducerStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS message producer

**JMSSecurityException** - Exception in `javax.jms`

This exception must be thrown when a provider rejects a user name/password submitted by a client.

**JMSSecurityException(String, String)** - Constructor for exception

```java
javax.jms.JMSSecurityException
```

Constructs a JMSSecurityException with the specified reason and error code.

**JMSSecurityException(String)** - Constructor for exception

```java
javax.jms.JMSSecurityException
```

Constructs a JMSSecurityException with the specified reason.

**JMSSessionStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS session.

**JMSStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a JMS Resource

**JoinColumn** - Annotation Type in `javax.persistence`

Is used to specify a mapped column for joining an entity association.

**JoinColumns** - Annotation Type in `javax.persistence`

 Defines mapping for the composite foreign keys.

**JoinColumn** - Annotation Type in `javax.persistence`

 This annotation is used in the mapping of associations.

**joinTransaction()** - Method in interface

```java
javax.persistence.EntityManager
```

Indicate to the EntityManager that a JTA transaction is active.

**JspApplicationContext** - Interface in `javax.servlet.jsp`

Stores application-scoped information relevant to JSP containers.

**JspContext** - Class in `javax.servlet.jsp`

JspContext serves as the base class for the PageContext class and abstracts all information that is not specific to servlets.

**JspContext()** - Constructor for class `javax.servlet.jsp.JspContext`

Sole constructor.

**jspDestroy()** - Method in interface `javax.servlet.jsp.JspPage`

The jspDestroy() method is invoked when the JSP page is about to be destroyed.

**JspEngineInfo** - Class in `javax.servlet.jsp`

The JspEngineInfo is an abstract class that provides information on the current JSP engine.

**JspEngineInfo()** - Constructor for class
javax.servlet.jsp.

**JspEngineInfo**

Sole constructor.

**JspException** - Exception in javax.servlet.jsp

A generic exception known to the JSP engine; uncaught JspExceptions will result in an invocation of the errorpage machinery.

**JspException()** - Constructor for exception

Construct a JspException.

**JspException(String)** - Constructor for exception

Constructs a new JSP exception with the specified message.

**JspException(String, Throwable)** - Constructor for exception

Constructs a new JspException with the specified detail message and cause.

**JspException(Throwabla)** - Constructor for exception

Constructs a new JspException with the specified cause.

**JspFactory** - Class in javax.servlet.jsp

The JspFactory is an abstract class that defines a number of factory methods available to a JSP page at runtime for the purposes of creating instances of various interfaces and classes used to support the JSP implementation.

**JspFactory()** - Constructor for class javax.servlet.jsp.JspFactory

Sole constructor.

**JspFragment** - Class in javax.servlet.jsp.tagext

Encapsulates a portion of JSP code in an object that can be invoked as many times as needed.

**JspFragment()** - Constructor for class

jersey.servlet.jsp.tagext.JspFragment

**JspIdConsumer** - Interface in javax.servlet.jsp.tagext

This interface indicates to the container that a tag handler wishes to be provided with a compiler generated ID.

**jspInit()** - Method in interface javax.servlet.jsp.JspPage

The jspInit() method is invoked when the JSP page is initialized.

**JspPage** - Interface in javax.servlet.jsp

The JspPage interface describes the generic interaction that a JSP Page Implementation class must satisfy; pages that use the HTTP protocol are described by the HttpJspPage interface.

**JspTag** - Interface in javax.servlet.jsp.tagext

Serves as a base class for Tag and SimpleTag.

**JspTagException** - Exception in javax.servlet.jsp

Exception to be used by a Tag Handler to indicate some unrecoverable error.

**JspTagException(String)** - Constructor for exception

Constructs a new JspTagException with the specified message.
**JspTagException()** - Constructor for exception

*javax.servlet.jsp.JspTagException*

Constructs a new JspTagException with no message.

**JspTagException(String, Throwable)** - Constructor for exception

*javax.servlet.jsp.JspTagException*

Constructs a new JspTagException when the JSP Tag needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation, including a description message.

**JspTagException(Throwable)** - Constructor for exception

*javax.servlet.jsp.JspTagException*

Constructs a new JSP Tag exception when the JSP Tag needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation.

**jspversion** - Variable in class

*javax.servlet.jsp.tagext.TagLibraryInfo*

The version of the JSP specification this tag library is written to.

**JspWriter** - Class in *javax.servlet.jsp*

The actions and template data in a JSP page is written using the JspWriter object that is referenced by the implicit variable out which is initialized automatically using methods in the PageContext object.

**JspWriter(int, boolean)** - Constructor for class

*javax.servlet.jsp.JspWriter*

Protected constructor.

**JTAStats** - Interface in *javax.management.j2ee.statistics*

Specifies the statistics provided by a JTA resource.

**JVMStats** - Interface in *javax.management.j2ee.statistics*

Specifies the statistics provided by a Java VM.
Key - Interface in `javax.xml.registry.infomodel`
   Represents a unique key that identifies a RegistryObject.
KEY - Static variable in interface `javax.xml.registry.LifeCycleManager`
LASTUID - Static variable in interface javax.mail.UIDFolder
This is a special value that can be used as the end parameter in getMessagesByUID(start, end), to denote the UID of the last message in the folder.

LazyAssociatableConnectionManager - Interface in javax.resource.spi
This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation.

LazyEnlist(ManagedConnection) - Method in interface javax.resource.spi.LazyEnlistableConnectionManager
This method is called by a resource adapter (that is capable of lazy transaction enlistment optimization) in order to lazily enlist a connection object with a XA transaction.

LazyEnlistableConnectionManager - Interface in javax.resource.spi
This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation.

LazyEnlistableManagedConnection - Interface in javax.resource.spi
This is a mix-in interface that may be optionally implemented by a ManagedConnection implementation.

LE - Static variable in class javax.mail.search.ComparisonTerm

LengthValidator - Class in javax.faces.validator
LengthValidator is a Validator that checks the number of characters in the String representation of the value of the associated component.

LengthValidator() - Constructor for class javax.faces.validator.LengthValidator
Construct a Validator with no preconfigured limits.

LengthValidator(int) - Constructor for class javax.faces.validator.LengthValidator
Construct a Validator with the specified preconfigured limit.

LengthValidator(int, int) - Constructor for class javax.faces.validator.LengthValidator
Construct a Validator with the specified preconfigured limits.

Lifecycle - Class in javax.faces.lifecycle
Lifecycle manages the processing of the entire lifecycle of a particular JavaServer Faces request.

Lifecycle() - Constructor for class javax.faces.lifecycle.Lifecycle

LIFECYCLE_FACTORY - Static variable in class javax.faces.FactoryFinder
The property name for the LifecycleFactory class name.

LIFECYCLE_ID_ATTR - Static variable in class javax.faces.webapp.FacesServlet
Context initialization parameter name for the lifecycle identifier of the Lifecycle instance to be utilized.

LifecycleFactory - Class in javax.faces.lifecycle
**LifecycleFactory** is a factory object that creates (if needed) and returns **Lifecycle** instances.

**LifecycleFactory()** - Constructor for class
javax.faces.lifecycle.LifecycleFactory

**LifeCycleManager** - Interface in javax.xml.registry
The LifeCycleManager interface is the main interface in the API for managing life cycle operations on objects defined by the information model.

**limit** - Variable in class javax.mail.Quota.Resource
The usage limit for the resource.

**linkConfiguration(PolicyConfiguration)** - Method in interface
javax.security.jacc.PolicyConfiguration
Creates a relationship between this configuration and another such that they share the same principal-to-role mappings.

**list(String)** - Method in class javax.mail.Folder
Returns a list of Folders belonging to this Folder's namespace that match the specified pattern.

**list()** - Method in class javax.mail.Folder
Convenience method that returns the list of folders under this Folder.

**ListDataModel** - Class in javax.faces.model
**ListDataModel** is a convenience implementation of **DataModel** that wraps an List of Java objects.

**ListDataModel()** - Constructor for class
javax.faces.model.ListDataModel
Construct a new ListDataModel with no specified wrapped data.

**ListDataModel(List)** - Constructor for class
javax.faces.model.ListDataModel
Construct a new ListDataModel wrapping the specified list.

**ListELResolver** - Class in javax.el
Defines property resolution behavior on instances of **List**.

**ListELResolver()** - Constructor for class javax.el.ListELResolver
Creates a new read/write ListELResolver.

**ListELResolver(boolean)** - Constructor for class
javax.el.ListELResolver
Creates a new ListELResolver whose read-only status is determined by the given parameter.

**ListenerRegistration** - Interface in javax.management.j2ee
ListenerRegistration defines the methods which clients of the MEJB use to add and remove event listeners.

**listSubscribed(String)** - Method in class javax.mail.Folder
Returns a list of subscribed Folders belonging to this Folder's namespace that match the specified pattern.

**listSubscribed()** - Method in class javax.mail.Folder
Convenience method that returns the list of subscribed folders under this Folder.

**load(InputStream)** - Method in class
javax.mail.internet.InternetHeaders
Read and parse the given RFC822 message stream till the blank
line separating the header from the body.

**loadService(Class)** - Method in class `javax.xml.rpc.ServiceFactory`
Create an instance of the generated service implementation class for a given service interface, if available.

**loadService(URL, Class, Properties)** - Method in class `javax.xml.rpc.ServiceFactory`
Create an instance of the generated service implementation class for a given service interface, if available.

**loadService(URL, QName, Properties)** - Method in class `javax.xml.rpc.ServiceFactory`
Create an instance of the generated service implementation class for a given service, if available.

**Lob** - Annotation Type in `javax.persistence`
Specifies that a persistent property or field should be persisted as a large object to a database-supported large object type.

**Local** - Annotation Type in `javax.ejb`
When used on the bean class, declares the local business interface(s) for a session bean.

**LOCAL_TRANSACTION_COMMITTED** - Static variable in class `javax.resource.spi.ConnectionEvent`
Event notification that a Resource Manager Local Transaction was committed on the connection

**LOCAL_TRANSACTION_ROLLEDBACK** - Static variable in class `javax.resource.spi.ConnectionEvent`
Event notification that a Resource Manager Local Transaction was rolled back on the connection

**LOCAL_TRANSACTION_STARTED** - Static variable in class `javax.resource.spi.ConnectionEvent`
Event notification that a Resource Manager Local Transaction was started on the connection

**LocalHome** - Annotation Type in `javax.ejb`
Declares the Local Home or adapted Local Home interface for a session bean.

**LOCALIZED_STRING** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**LocalizedMessage** - Interface in `javax.xml.registry.infomodel`
This interface is used as a simple wrapper interface that associates a String with its Locale.

**LocalTransaction** - Interface in `javax.resource.cci`
The LocalTransaction defines a transaction demarcation interface for resource manager local transactions.

**LocalTransaction** - Interface in `javax.resource.spi`
LocalTransaction interface provides support for transactions that are managed internal to an EIS resource manager, and do not require an external transaction manager.

**localTransactionCommitted(ConnectionEvent)** - Method in interface `javax.resource.spi.ConnectionEventListener`
Notifies that a Resource Manager Local Transaction was
committed on the ManagedConnection instance.

LocalTransactionException - Exception in javax.resource.spi
A LocalTransactionException represents various error conditions related to the local transaction management contract.

LocalTransactionException() - Constructor for exception
javax.resource.spi.LocalTransactionException
Constructs a new instance with null as its detail message.

LocalTransactionException(String) - Constructor for exception
javax.resource.spi.LocalTransactionException
Constructs a new instance with the specified detail message.

LocalTransactionException(Throwable) - Constructor for exception
javax.resource.spi.LocalTransactionException
Constructs a new throwable with the specified cause.

LocalTransactionException(String, Throwable) - Constructor for exception
javax.resource.spi.LocalTransactionException
Constructs a new throwable with the specified detail message and cause.

LocalTransactionRolledback(ConnectionEvent) - Method in interface
javax.resource.spi.ConnectionEventListener
Notifies that a Resource Manager Local Transaction was rolled back on the ManagedConnection instance.

LocalTransactionStarted(ConnectionEvent) - Method in interface
javax.resource.spi.ConnectionEventListener
Notifies that a Resource Manager Local Transaction was started on the ManagedConnection instance.

Location - Interface in javax.xml.stream
Provides information on the location of an event.

location - Variable in exception
javax.xml.stream.XMLStreamException

lock(Object, LockModeType) - Method in interface
javax.persistence.EntityManager
Set the lock mode for an entity object contained in the persistence context.

LockModeType - Enum in javax.persistence
Lock modes that can be specified by means of the EntityManager.lock() method.

log(String) - Method in class javax.faces.context.ExternalContext
Log the specified message to the application object.

log(String, Throwable) - Method in class
javax.faces.context.ExternalContext
Log the specified message and exception to the application object.

log - Static variable in class
javax.faces.webapp.UICOMPONENTTAGBASE
**log(String)** - Method in class `javax.servlet.GenericServlet`

Writes the specified message to a servlet log file, prepended by the servlet's name.

**log(String, Throwable)** - Method in class `javax.servlet.GenericServlet`

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file, prepended by the servlet's name.

**log(String)** - Method in interface `javax.servlet.ServletContext`

Writes the specified message to a servlet log file, usually an event log.

**log(Exception, String)** - Method in interface `javax.servlet.ServletContext`

Deprecated. As of Java Servlet API 2.1, use `ServletContext.log(String message, Throwable throwable)` instead.

This method was originally defined to write an exception's stack trace and an explanatory error message to the servlet log file.

**log(String, Throwable)** - Method in interface `javax.servlet.ServletContext`

Writes an explanatory message and a stack trace for a given Throwable exception to the servlet log file.

**LogicalHandler**<C> extends `LogicalMessageContext` - Interface in `javax.xml.ws.handler`

The LogicalHandler extends Handler to provide typesafety for the message context parameter.

**LogicalMessage** - Interface in `javax.xml.ws`

The LogicalMessage interface represents a protocol agnostic XML message and contains methods that provide access to the payload of the message.

**LogicalMessageContext** - Interface in `javax.xml.ws.handler`

The LogicalMessageContext interface extends MessageContext to provide access to a the contained message as a protocol neutral LogicalMessage

**LONG_ID** - Static variable in class `javax.faces.convert.LongConverter`

The message identifier of the `FacesMessage` to be created if the conversion to Long fails.

**LongConverter** - Class in `javax.faces.convert`

Converter implementation for java.lang.Long (and long primitive) values.

**LongConverter()** - Constructor for class `javax.faces.convert.LongConverter`

**LongHolder** - Class in `javax.xml.rpc.holders`

**LongHolder()** - Constructor for class
javax.xml.rpc.holders.LongHolder

LongHolder(long) - Constructor for class javax.xml.rpc.holders.LongHolder

LongRangeValidator - Class in javax.faces.validator
   LongRangeValidator is a Validator that checks the value of the corresponding component against specified minimum and maximum values.
LongRangeValidator() - Constructor for class javax.faces.validator.LongRangeValidator
   Construct a Validator with no preconfigured limits.
LongRangeValidator(long) - Constructor for class javax.faces.validator.LongRangeValidator
   Construct a Validator with the specified preconfigured limit.
LongRangeValidator(long, long) - Constructor for class javax.faces.validator.LongRangeValidator
   Construct a Validator with the specified preconfigured limits.
LongWrapperHolder - Class in javax.xml.rpc.holders
LongWrapperHolder() - Constructor for class javax.xml.rpc.holders.LongWrapperHolder
LongWrapperHolder(Long) - Constructor for class javax.xml.rpc.holders.LongWrapperHolder

lookup(String) - Method in interface javax.ejb.EJBContext
   Lookup a resource within the component's private naming context.
LT - Static variable in class javax.mail.search.ComparisonTerm
**MailcapCommandMap** - Class in `javax.activation`
MailcapCommandMap extends the CommandMap abstract class.

**MailcapCommandMap()** - Constructor for class
`javax.activation.MailcapCommandMap`
The default Constructor.

**MailcapCommandMap(String)** - Constructor for class
`javax.activation.MailcapCommandMap`
Constructor that allows the caller to specify the path of a mailcap file.

**MailcapCommandMap(InputStream)** - Constructor for class
`javax.activation.MailcapCommandMap`
Constructor that allows the caller to specify an InputStream containing a mailcap file.

**MailDateFormat** - Class in `javax.mail.internet`

**MailDateFormat()** - Constructor for class
`javax.mail.internet.MailDateFormat`

**MailEvent** - Class in `javax.mail.event`
Common base class for mail events, defining the dispatch method.

**MailEvent(Object)** - Constructor for class
`javax.mail.event.MailEvent`

**makeRegistrySpecificRequest(String)** - Method in interface
`javax.xml.registry.RegistryService`
Takes a String that is an XML request in a registry-specific format, sends the request to the registry, and returns a String that is the registry-specific XML response.

**ManagedConnection** - Interface in `javax.resource.spi`
ManagedConnection instance represents a physical connection to the underlying EIS.

**ManagedConnectionFactory** - Interface in `javax.resource.spi`
ManagedConnectionFactory instance is a factory of both ManagedConnection and EIS-specific connection factory instances.

**ManagedConnectionMetaData** - Interface in `javax.resource.spi`
The ManagedConnectionMetaData interface provides information about the underlying EIS instance associated with a ManagedConnection instance.

**Management** - Interface in `javax.management.j2ee`
The Management interface provides the APIs to navigate and manipulate managed objects.

**ManagementHome** - Interface in `javax.management.j2ee`
The required home interface for the J2EE Management EJB
component (MEJB).

ManyToMany - Annotation Type in javax.persistence
Defines a many-valued association with many-to-many multiplicity.

ManyToOne - Annotation Type in javax.persistence
This annotation defines a single-valued association to another entity class that has many-to-one multiplicity.

MapELResolver - Class in javax.el
Defines property resolution behavior on instances of Map.
MapELResolver() - Constructor for class javax.el.MapELResolver
Creates a new read/write MapELResolver.
MapELResolver(boolean) - Constructor for class javax.el.MapELResolver
Creates a new MapELResolver whose read-only status is determined by the given parameter.

MapKey - Annotation Type in javax.persistence
Is used to specify the map key for associations of type Map.

MapMessage - Interface in javax.jms
A MapMessage object is used to send a set of name-value pairs.

MappedRecord - Interface in javax.resource.cci
The interface javax.resource.cci.MappedRecord is used for key-value map based representation of record elements.

MappedSuperclass - Annotation Type in javax.persistence
Designates a class whose mapping information is applied to the entities that inherit from it.

mark(int) - Method in class javax.mail.util.SharedFileInputStream
See the general contract of the mark method of InputStream.
markSupported() - Method in class javax.mail.util.SharedFileInputStream
Tests if this input stream supports the mark and reset methods.
marshal(String) - Method in class javax.xml.bind.annotation.adapters.CollapsedStringAdapter
No-op.
marshal(byte[]) - Method in class javax.xml.bind.annotation.adapters.HexBinaryAdapter

marshal(String) - Method in class javax.xml.bind.annotation.adapters.NormalizedStringAdapter
No-op.
marshal(BoundType) - Method in class javax.xml.bind.annotation.adapters.XmlAdapter
Convert a bound type to a value type.
marshal(ElementT, ValidationEventHandler) - Method in interface javax.xml.bind.annotation.DomHandler
This method is called when a JAXB provider needs to marshal an element to XML.
marshal(Element, ValidationEventHandler) - Method in class javax.xml.bind.annotation.W3CDomHandler

marshal(Object, XmlNode) - Method in class javax.xml.bind.Binder
Marshal a JAXB object tree to a new XML document.

**marshal(Object, OutputStream)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, Writer)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, ContentHandler)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, Node)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, XMLEventWriter)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, XMLStreamWriter)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl

**marshal(Object, Result)** - Method in interface
developed as the implementation

**MarshalException** - Exception in javax.xml.bind
This exception indicates that an error has occurred while performing a marshal operation that the provider is unable to recover from.

**MarshalException(String)** - Constructor for exception
javax.xml.bind.**MarshalException**

Construct a MarshalException with the specified detail message.

**MarshalException(String, String)** - Constructor for exception

javax.xml.bind.**MarshalException**

Construct a MarshalException with the specified detail message and vendor specific errorCode.

**MarshalException(Throwable)** - Constructor for exception

javax.xml.bind.**MarshalException**

Construct a MarshalException with a linkedException.

**MarshalException(String, Throwable)** - Constructor for exception

javax.xml.bind.**MarshalException**

Construct a MarshalException with the specified detail message and linkedException.

**MarshalException(String, String, Throwable)** - Constructor for exception

javax.xml.bind.**MarshalException**

Construct a MarshalException with the specified detail message, vendor specific errorCode, and linkedException.

**Marshaller** - Interface in **javax.xml.bind**

The Marshaller class is responsible for governing the process of serializing Java content trees back into XML data.

**Marshaller.Listener** - Class in **javax.xml.bind**

Register an instance of an implementation of this class with a **Marshaller** to externally listen for marshal events.

**Marshaller.Listener()** - Constructor for class

javax.xml.bind.**Marshaller.Listener**

**match(MimeType)** - Method in class java.activation.**MimeType**

Determine if the primary and sub type of this object is the same as what is in the given type.

**match(String)** - Method in class java.activation.**MimeType**

Determine if the primary and sub type of this object is the same as the content type described in rawdata.

**match(ContentType)** - Method in class javax.mail.internet.**ContentType**

Match with the specified ContentType object.

**match(String)** - Method in class javax.mail.internet.**ContentType**

Match with the specified content-type string.

**match(SearchTerm)** - Method in class javax.mail.**Message**

Apply the specified Search criterion to this message.

**match(Address)** - Method in class javax.mail.search.**AddressStringTerm**

Check whether the address pattern specified in the constructor is a substring of the string representation of the given Address object.

**match(Address)** - Method in class javax.mail.search.**AddressTerm**

Match against the argument Address.

**match(Message)** - Method in class javax.mail.search.**AndTerm**

The AND operation.

**match(Message)** - Method in class javax.mail.search.**BodyTerm**
The match method.

**match(Date)** - Method in class **javax.mail.search.DateTerm**
The date comparison method.

**match(Message)** - Method in class **javax.mail.search.FlagTerm**
The comparison method.

**match(Message)** - Method in class **javax.mail.search.FromStringTerm**
Check whether the address string specified in the constructor is a substring of the From address of this Message.

**match(Message)** - Method in class **javax.mail.search.FromTerm**
The address comparator.

**match(Message)** - Method in class **javax.mail.search.HeaderTerm**
The header match method.

**match(int)** - Method in class **javax.mail.search.IntegerComparisonTerm**

**match(Message)** - Method in class **javax.mail.search.MessageIDTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.MessageNumberTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.NotTerm**

**match(Message)** - Method in class **javax.mail.search.OrTerm**
The OR operation.

**match(Message)** - Method in class **javax.mail.search.ReceivedDateTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.RecipientStringTerm**
Check whether the address specified in the constructor is a substring of the recipient address of this Message.

**match(Message)** - Method in class **javax.mail.search.RecipientTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.SearchTerm**
This method applies a specific match criterion to the given message and returns the result.

**match(Message)** - Method in class **javax.mail.search.SentDateTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.SizeTerm**
The match method.

**match(String)** - Method in class **javax.mail.search.StringTerm**
The match method.

**match(Message)** - Method in class **javax.mail.search.SubjectTerm**
The match method.

**matchManagedConnections(Set, Subject, ConnectionRequestInfo)** - Method in interface **javax.resource.spi.ManagedConnectionFactory**
Returns a matched connection from the candidate set of connections.

**MAXBQUALSIZE** - Static variable in interface **javax.transaction.xa.Xid**
Maximum number of bytes returned by getBqual.
**MAXGTRIDSIZE** - Static variable in interface javax.transaction.xa.Xid
   Maximum number of bytes returned by getGtrid.

**MAXIMUM MESSAGE ID** - Static variable in class javax.faces.validator.DoubleRangeValidator
   The message identifier of the FacesMessage to be created if the maximum value check fails.

**MAXIMUM MESSAGE ID** - Static variable in class javax.faces.validator.LengthValidator
   The message identifier of the FacesMessage to be created if the maximum length check fails.

**MAXIMUM MESSAGE ID** - Static variable in class javax.faces.validator.LongRangeValidator
   The message identifier of the FacesMessage to be created if the maximum value check fails.

**merge(T)** - Method in interface javax.persistence.EntityManager
   Merge the state of the given entity into the current persistence context.

**Message** - Interface in javax.jms
   The Message interface is the root interface of all JMS messages.

**message** - Variable in class javax.mail.event.StoreEvent
   The message text to be presented to the user.

**Message** - Class in javax.mail
   This class models an email message.

**Message()** - Constructor for class javax.mail.Message
   No-arg version of the constructor.

**Message(Folder, int)** - Constructor for class javax.mail.Message
   Constructor that takes a Folder and a message number.

**Message(Session)** - Constructor for class javax.mail.Message
   Constructor that takes a Session.

**Message.RecipientType** - Class in javax.mail
   This inner class defines the types of recipients allowed by the Message class.

**Message.RecipientType(String)** - Constructor for class javax.mail.Message.RecipientType
   Constructor for use by subclasses.

**MESSAGE_DELIVERED** - Static variable in class javax.mail.event.TransportEvent
   Message has been successfully delivered to all recipients by the transport firing this event. validSent[] contains all the addresses this transport sent to successfully. validUnsent[] and invalid[] should be null.

**MESSAGE_NOT_DELIVERED** - Static variable in class javax.mail.event.TransportEvent
   Message was not sent for some reason. validSent[] should be null.

**MESSAGE_OUTBOUND_PROPERTY** - Static variable in interface javax.xml.ws.handler.MessageContext
   Standard property: message direction, true for outbound...
messages, false for inbound.

**MESSAGE_PARTIALLY_DELIVERED** - Static variable in class javax.mail.event.TransportEvent
Message was successfully sent to some recipients but not to all.

**MessageAware** - Interface in javax.mail
An interface optionally implemented by DataSources to supply information to a DataContentHandler about the message context in which the data content object is operating.

**messageChanged(MessageChangedEvent)** - Method in interface javax.mail.event.MessageChangedListener
Invoked when a message is changed.

**MessageChangedEvent** - Class in javax.mail.event
This class models Message change events.

**MessageChangedEvent(Object, int, Message)** - Constructor for class javax.mail.event.MessageChangedEvent
Constructor.

**MessageChangedListener** - Interface in javax.mail.event
This is the Listener interface for MessageChanged events

**MessageConsumer** - Interface in javax.jms
A client uses a MessageConsumer object to receive messages from a destination.

**MessageContext** - Class in javax.mail
The context in which a piece of Message content is contained.

**MessageContext(Part)** - Constructor for class javax.mail.MessageContext
Create a MessageContext object describing the context of the given Part.

**MessageContext** - Interface in javax.xml.rpc.handler
The interface MessageContext abstracts the message context that is processed by a handler in the handle method.

**MessageContext** - Interface in javax.xml.ws.handler
The interface MessageContext abstracts the message context that is processed by a handler in the handle method.

**MessageContext.Scope** - Enum in javax.xml.ws.handler
Property scope.

**MessageCountAdapter** - Class in javax.mail.event
The adapter which receives MessageCount events.

**MessageCountAdapter()** - Constructor for class javax.mail.event.MessageCountAdapter

**MessageCountEvent** - Class in javax.mail.event
This class notifies changes in the number of messages in a folder.

**MessageCountEvent(Folder, int, boolean, Message[])** - Constructor for class javax.mail.event.MessageCountEvent
Constructor.

**MessageCountListener** - Interface in javax.mail.event
This is the Listener interface for MessageCount events.

**messageDelivered(TransportEvent)** - Method in class
javax.mail.event.TransportAdapter

messageDelivered(TransportEvent) - Method in interface javax.mail.event.TransportListener
Invoked when a Message is successfully delivered.

MessageDriven - Annotation Type in javax.ejb
Component-defining annotation for a message driven bean.

MessageDrivenBean - Interface in javax.ejb
The MessageDrivenBean interface is implemented by every message-driven enterprise Bean class.

MessageDrivenBeanStats - Interface in javax.management.j2ee.statistics
Specifies the statistics provided by a message driven bean.

MessageDrivenContext - Interface in javax.ejb
The MessageDrivenContext interface provides access to the runtime message-driven context that the container provides for a message-driven enterprise Bean instance.

MessageEndpoint - Interface in javax.resource.spi.endpoint
This defines a contract for a message endpoint.

MessageEndpointFactory - Interface in javax.resource.spi.endpoint
This serves as a factory for creating message endpoints.

MessageEOFException - Exception in javax.jms
This exception must be thrown when an unexpected end of stream has been reached when a StreamMessage or BytesMessage is being read.

MessageEOFException(String, String) - Constructor for exception javax.jms.MessageEOFException
Constructs a MessageEOFException with the specified reason and error code.

MessageEOFException(String) - Constructor for exception javax.jms.MessageEOFException
Constructs a MessageEOFException with the specified reason.

MessageFactory - Class in javax.xml.soap
A factory for creating SOAPMessage objects.
MessageFactory() - Constructor for class javax.xml.soap.MessageFactory

MessageFormatException - Exception in javax.jms
This exception must be thrown when a JMS client attempts to use a data type not supported by a message or attempts to read data in a message as the wrong type.

MessageFormatException(String, String) - Constructor for exception javax.jms.MessageFormatException
Constructs a MessageFormatException with the specified reason and error code.

MessageFormatException(String) - Constructor for exception javax.jms.MessageFormatException
Constructs a MessageFormatException with the specified reason.

MessageIDTerm - Class in javax.mail.search
This term models the RFC822 "MessageId" - a message-id for
Internet messages that is supposed to be unique per message.

**MessageIDTerm(String)** - Constructor for class javax.mail.search.MessageIDTerm

**MessageListener** - Interface in javax.jms

A MessageListener object is used to receive asynchronously delivered messages.

**MessageListener** - Interface in javax.resource.cci

This serves as a request-response message listener type that message endpoints (message-driven beans) may implement.

**messageNotDelivered(TransportEvent)** - Method in class javax.mail.event.TransportAdapter

**MessageNotReadableException** - Exception in javax.jms

This exception must be thrown when a JMS client attempts to read a write-only message.

**MessageNotReadableException(String, String)** - Constructor for exception javax.jms.MessageNotReadableException

Constructs a MessageNotReadableException with the specified reason and error code.

**MessageNotReadableException(String)** - Constructor for exception javax.jms.MessageNotReadableException

Constructs a MessageNotReadableException with the specified reason.

**MessageNotWriteableException** - Exception in javax.jms

This exception must be thrown when a JMS client attempts to write to a read-only message.

**MessageNotWriteableException(String, String)** - Constructor for exception javax.jms.MessageNotWriteableException

Constructs a MessageNotWriteableException with the specified reason and error code.

**MessageNotWriteableException(String)** - Constructor for exception javax.jms.MessageNotWriteableException

Constructs a MessageNotWriteableException with the specified reason.

**MessageNumberTerm** - Class in javax.mail.search

This class implements comparisons for Message numbers.

**MessageNumberTerm(int)** - Constructor for class javax.mail.search.MessageNumberTerm

**messagePartiallyDelivered(TransportEvent)** - Method in class javax.mail.event.TransportAdapter

**MessageProducer** - Interface in javax.jms
A client uses a MessageProducer object to send messages to a destination.

**MessageRemovedException** - Exception in `javax.mail`
- The exception thrown when an invalid method is invoked on an expunged Message.

**MessageRemovedException()** - Constructor for exception `javax.mail.Message RemovedException`
- Constructs a MessageRemovedException with no detail message.

**MessageRemovedException(String)** - Constructor for exception `javax.mail.Message RemovedException`
- Constructs a MessageRemovedException with the specified detail message.

**messagesAdded(MessageCountEvent)** - Method in class `javax.mail.event.MessageCountAdapter`

**messagesAdded(MessageCountEvent)** - Method in interface `javax.mail.event.MessageCountListener`
- Invoked when messages are added into a folder.

**messagesRemoved(MessageCountEvent)** - Method in class `javax.mail.event.MessageCountAdapter`

**messagesRemoved(MessageCountEvent)** - Method in interface `javax.mail.event.MessageCountListener`
- Invoked when messages are removed (expunged) from a folder.

**MessagingException** - Exception in `javax.mail`
- The base class for all exceptions thrown by the Messaging classes.

**MessagingException()** - Constructor for exception `javax.mail.MessagingException`
- Constructs a MessagingException with no detail message.

**MessagingException(String)** - Constructor for exception `javax.mail.MessagingException`
- Constructs a MessagingException with the specified detail message.

**MessagingException(String, Exception)** - Constructor for exception `javax.mail.MessagingException`
- Constructs a MessagingException with the specified Exception and detail message.

**MethodBinding** - Class in `javax.faces.el`
- Deprecated. This has been replaced by `MethodExpression`.

**MethodBinding()** - Constructor for class `javax.faces.el.MethodBinding`
- Deprecated.

**MethodExpression** - Class in `javax.el`
- An Expression that refers to a method on an object.

**MethodExpression()** - Constructor for class `javax.el.MethodExpression`

**MethodExpressionActionListener** - Class in `javax.faces.event`
- `MethodExpressionActionListener` is an `ActionListener` that wraps
a MethodExpression.
MethodExpressionActionListener() - Constructor for class javax.faces.event.MethodExpressionActionListener
MethodExpressionActionListener(MethodExpression) - Constructor for class javax.faces.event.MethodExpressionActionListener
  Construct a ValueChangeListener that contains a MethodExpression.
MethodExpressionValidator - Class in javax.faces.validator
  MethodExpressionValidator is a Validator that wraps a MethodExpression, and it performs validation by executing a method on an object identified by the MethodExpression.
MethodExpressionValidator() - Constructor for class javax.faces.validator.MethodExpressionValidator
MethodExpressionValidator(MethodExpression) - Constructor for class javax.faces.validator.MethodExpressionValidator
  Construct a Validator that contains a MethodExpression.
MethodExpressionValueChangeListener - Class in javax.faces.event
  MethodExpressionValueChangeListener is a ValueChangeListener that wraps a MethodExpression.
MethodExpressionValueChangeListener() - Constructor for class javax.faces.event.MethodExpressionValueChangeListener
MethodExpressionValueChangeListener(MethodExpression) - Constructor for class javax.faces.event.MethodExpressionValueChangeListener
  Construct a ValueChangeListener that contains a MethodExpression.
MethodInfo - Class in javax.el
  Holds information about a method that a MethodExpression evaluated to.
MethodInfo(String, Class<?>, Class<?>[]) - Constructor for class javax.el.MethodInfo
  Creates a new instance of MethodInfo with the given information.
MethodNotFoundException - Exception in javax.el
  Thrown when a method could not be found while evaluating a MethodExpression.
MethodNotFoundException() - Constructor for exception javax.el.MethodNotFoundException
  Creates a MethodNotFoundException with no detail message.
MethodNotFoundException(String) - Constructor for exception javax.el.MethodNotFoundException
  Creates a MethodNotFoundException with the provided detail message.
MethodNotFoundException(Throwables) - Constructor for exception javax.el.MethodNotFoundException
  Creates a MethodNotFoundException with the given root cause.
MethodNotFoundException(String, Throwable) - Constructor for exception javax.el.MethodNotFoundException
Creates a MethodNotFoundException with the given detail message and root cause.

**MethodNotFoundException** - Exception in javax.faces.el

*Deprecated.* *This has been replaced by MethodNotFoundException.*

**MethodNotFoundException()** - Constructor for exception javax.faces.el.MethodNotFoundException

*Deprecated.* Construct a new exception with no detail message or root cause.

**MethodNotFoundException(String)** - Constructor for exception javax.faces.el.MethodNotFoundException

*Deprecated.* Construct a new exception with the specified detail message and no root cause.

**MethodNotFoundException(Throwable)** - Constructor for exception javax.faces.el.MethodNotFoundException

*Deprecated.* Construct a new exception with the specified root cause.

**MethodNotFoundException(String, Throwable)** - Constructor for exception javax.faces.el.MethodNotFoundException

*Deprecated.* Construct a new exception with the specified detail message and root cause.

**MethodNotSupportedException** - Exception in javax.mail

The exception thrown when a method is not supported by the implementation

**MethodNotSupportedException()** - Constructor for exception javax.mail.MethodNotSupportedException

Constructs a MethodNotSupportedException with no detail message.

**MethodNotSupportedException(String)** - Constructor for exception javax.mail.MethodNotSupportedException

Constructs a MethodNotSupportedException with the specified detail message.

**MIME** - Static variable in class javax.mail.internet.HeaderTokenizer

MIME specials

**MimeBodyPart** - Class in javax.mail.internet

This class represents a MIME body part.

**MimeBodyPart()** - Constructor for class javax.mail.internet.MimeBodyPart

An empty MimeBodyPart object is created.

**MimeBodyPart(InputStream)** - Constructor for class javax.mail.internet.MimeBodyPart

Constructs a MimeBodyPart by reading and parsing the data from the specified input stream.

**MimeBodyPart(InternetHeaders, byte[])** - Constructor for class javax.mail.internet.MimeBodyPart

Constructs a MimeBodyPart using the given header and content bytes.

**mimeCharset(String)** - Static method in class javax.mail.internet.MimeUtility

Convert a Java charset into its MIME charset name.

**MimeHeader** - Class in javax.xml.soap
An object that stores a MIME header name and its value.

**MimeHeader(String, String)** - Constructor for class javax.xml.soap.MimeHeader
Constructs a MimeHeader object initialized with the given name and value.

**MimeHeaders** - Class in javax.xml.soap
A container for MimeHeader objects, which represent the MIME headers present in a MIME part of a message.

**MimeHeaders()** - Constructor for class javax.xml.soap.MimeHeaders
Constructs a default MimeHeaders object initialized with an empty Vector object.

**MimeMessage** - Class in javax.mail.internet
This class represents a MIME style email message.

**MimeMessage(Session)** - Constructor for class javax.mail.internet.MimeMessage
Default constructor.

**MimeMessage(Session, InputStream)** - Constructor for class javax.mail.internet.MimeMessage
Constructs a MimeMessage by reading and parsing the data from the specified MIME InputStream.

**MimeMessage(MimeMessage)** - Constructor for class javax.mail.internet.MimeMessage
Constructs a new MimeMessage with content initialized from the source MimeMessage.

**MimeMessage(Folder, int)** - Constructor for class javax.mail.internet.MimeMessage
Constructs an empty MimeMessage object with the given Folder and message number.

**MimeMessage(Folder, InputStream, int)** - Constructor for class javax.mail.internet.MimeMessage
Constructs a MimeMessage by reading and parsing the data from the specified MIME InputStream.

**MimeMessage(Folder, InternetHeaders, byte[], int)** - Constructor for class javax.mail.internet.MimeMessage
Constructs a MimeMessage from the given InternetHeaders object and content.

**MimeMessage.RecipientType** - Class in javax.mail.internet
This inner class extends the javax.mail.Message.RecipientType class to add additional RecipientTypes.

**MimeMessage.RecipientType(String)** - Constructor for class javax.mail.internet.MimeMessage.RecipientType

**MimeMultipart** - Class in javax.mail.internet
The MimeMultipart class is an implementation of the abstract Multipart class that uses MIME conventions for the multipart data.

**MimeMultipart()** - Constructor for class javax.mail.internet.MimeMultipart
Default constructor.

**MimeMultipart(String)** - Constructor for class
javax.mail.internet. **MimeMultipart**
Construct a MimeMultipart object of the given subtype.

**MimeMultipart(DataSource)** - Constructor for class
Constructs a MimeMultipart object and its bodyparts from the given DataSource.

**MimePart** - Interface in *javax.mail.internet*
The MimePart interface models an **Entity** as defined by MIME (RFC2045, Section 2.4).

**MimePartDataSource** - Class in *javax.mail.internet*
A utility class that implements a DataSource out of a MimePart.

**MimePartDataSource(MimePart)** - Constructor for class
Constructs a DataSource from a MimePart.

**MimeType** - Class in *javax.activation*
A Multipurpose Internet Mail Extension (MIME) type, as defined in RFC 2045 and 2046.

**MimeType()** - Constructor for class *javax.activation.MimeType*
Default constructor.

**MimeType(String)** - Constructor for class *javax.activation.MimeType*
Constructor that builds a MimeType from a String.

**MimeType(String, String)** - Constructor for class *javax.activation.MimeType*
Constructor that builds a MimeType with the given primary and sub type but has an empty parameter list.

**MimeTypeParameterList** - Class in *javax.activation*
A parameter list of a MimeType as defined in RFC 2045 and 2046.

**MimeTypeParameterList()** - Constructor for class
Default constructor.

**MimeTypeParameterList(String)** - Constructor for class
Constructs a new MimeTypeParameterList with the passed in data.

**MimeTypeParseException** - Exception in *javax.activation*
A class to encapsulate MimeType parsing related exceptions.

**MimeTypeParseException()** - Constructor for exception
Constructs a MimeTypeParseException with no specified detail message.

**MimeTypeParseException(String)** - Constructor for exception
Constructs a MimeTypeParseException with the specified detail message.

**MimetypesFileTypeMap** - Class in *javax.activation*
This class extends FileTypeMap and provides data typing of files via their file extension.

**MimetypesFileTypeMap()** - Constructor for class
The default constructor.

**MimetypesFileTypeMap(String)** - Constructor for class
javax.activation.MimetypesFileTypeMap
Construct a MimetypesFileTypeMap with programmatic entries added from the named file.

MimetypesFileTypeMap(InputStream) - Constructor for class javax.activation.MimetypesFileTypeMap
Construct a MimetypesFileTypeMap with programmatic entries added from the InputStream.

MimeUtility - Class in javax.mail.internet
This is a utility class that provides various MIME related functionality.

MINIMUM_MESSAGE_ID - Static variable in class javax.mail.validator.DoubleRangeValidator
The message identifier of the FacesMessage to be created if the minimum value check fails.

MINIMUM_MESSAGE_ID - Static variable in class javax.mail.validator.LengthValidator
The message identifier of the FacesMessage to be created if the minimum length check fails.

MINIMUM_MESSAGE_ID - Static variable in class javax.mail.validator.LongRangeValidator
The message identifier of the FacesMessage to be created if the minimum value check fails.

mode - Variable in class javax.mail.Folder
The open mode of this folder.

modified - Variable in class javax.mail.internet.MimeMessage
A flag indicating whether the message has been modified.

ModuleType - Class in javax.enterprise.deploy.shared
Class ModuleTypes defines enumeration values for the J2EE module types.

ModuleType(int) - Constructor for class javax.enterprise.deploy.shared.ModuleType
Construct a new enumeration value with the given integer value.

msg - Variable in class javax.mail.event.MessageChangedEvent
The message that changed.

msgnum - Variable in class javax.mail.Message
The number of this message within its folder, or zero if the message was not retrieved from a folder.

msgs - Variable in class javax.mail.event.MessageCountEvent
The messages.

Multipart - Class in javax.mail
Multipart is a container that holds multiple body parts.

Multipart() - Constructor for class javax.mail.Multipart
Default constructor.

MultipartDataSource - Interface in javax.mail
MultipartDataSource is a DataSource that contains body parts.
name - Variable in class javax.mail.
- The name of the header.
name - Variable in class javax.mail.
- The name of the resource.
name - Variable in class javax.xml.bind.
- xml element tag name
Name - Interface in javax.xml.soap
- A representation of an XML name.
NamedNativeQueries - Annotation Type in javax.persistence
- Is used to specify an array of native SQL named queries.
NamedNativeQuery - Annotation Type in javax.persistence
- Is used to specify a native SQL named query.
NamedQuery - Annotation Type in javax.persistence
- Specifies an array of named Java Persistence query language queries.
Namespace - Interface in javax.xml.stream.events
- An interface that contains information about a namespace.
NAMESPACE - Static variable in interface
javax.xml.stream.XMLStreamConstants
- Indicates the event is a namespace declaration
NamespaceConstants - Class in javax.xml.rpc
- Constants used in JAX-RPC for namespace prefixes and URIs
NamespaceConstants() - Constructor for class
javax.xml.rpc.NamespaceConstants

NamingContainer - Interface in javax.faces.component
- NamingContainer is an interface that must be implemented by any
  UIComponent that wants to be a naming container.
NavigationHandler - Class in javax.faces.application
- A NavigationHandler is passed the outcome string returned by an
  application action invoked for this application, and will use
  this (along with related state information) to choose the view
  to be displayed next.
NavigationHandler() - Constructor for class
javax.faces.application.NavigationHandler

NE - Static variable in class javax.mail.search.
- ComparisonTerm
NESTED - Static variable in class
javax.servlet.jsp.tagext.VariableInfo
- Scope information that scripting variable is visible only
  within the start/end tags.
nested - Variable in exception javax.xml.stream.
- XMLStreamException
`newFolder` - Variable in class `javax.mail.event.FolderEvent`
The folder that represents the new name, in case of a RENAMED event.

`newInstance()` - Static method in class `javax.el.ExpressionFactory`
Creates a new instance of a ExpressionFactory.

`newInstance(Properties)` - Static method in class `javax.el.ExpressionFactory`
Create a new instance of a ExpressionFactory, with optional properties.

`newInstance()` - Static method in class `javax.mail.event.FolderEvent`
The folder that represents the new name, in case of a RENAMED event.

`newInstance(String)` - Static method in class `javax.xml.bind.JAXBContext`
Obtain a new instance of a JAXBContext class.

`newInstance(String, ClassLoader)` - Static method in class `javax.xml.bind.JAXBContext`
Obtain a new instance of a JAXBContext class.

`newInstance(String, ClassLoader, Map<String, ?>>)` - Static method in class `javax.xml.bind.JAXBContext`
Obtain a new instance of a JAXBContext class.

`newInstance(Class...)` - Static method in class `javax.xml.bind.JAXBContext`
Obtain a new instance of a JAXBContext class.

`newInstance(Class[], Map<String, ?>>)` - Static method in class `javax.xml.bind.JAXBContext`
Obtain a new instance of a JAXBContext class.

`newInstance()` - Static method in class `javax.xml.registry.ConnectionFactory`
Creates a defaultConnectionFactory object.

`newInstance()` - Static method in class `javax.xml.rpc.ServiceFactory`
Gets an instance of the ServiceFactory Only one copy of a factory exists and is returned to the application each time this method is called.

`newInstance()` - Static method in class `javax.xml.soap.MessageFactory`
Creates a new MessageFactory object that is an instance of the default implementation (SOAP 1.1), This method uses the following ordered lookup procedure to determine the MessageFactory implementation class to load: Use the javax.xml.soap.MessageFactory system property.

`newInstance(String)` - Static method in class `javax.xml.soap.MessageFactory`
Creates a new MessageFactory object that is an instance of the specified implementation.

`newInstance()` - Static method in class `javax.xml.soap.SOAPConnectionFactory`
Creates an instance of the default SOAPConnectionFactory object.

`newInstance()` - Static method in class `javax.xml.soap.SOAPElementFactory`
Deprecated. Creates a new instance of SOAPElementFactory.
newInstance() - Static method in class javax.xml.soap.SOAFactory
Creates a new SOAPFactory object that is an instance of the default implementation (SOAP 1.1). This method uses the following ordered lookup procedure to determine the SOAPFactory implementation class to load: Use the javax.xml.soap.SOAFactory system property.

newInstance(String) - Static method in class javax.xml.soap.SOAFactory
Creates a new SOAPFactory object that is an instance of the specified implementation, this method uses the SAAJMetaFactory to locate the implementation class and create the SOAPFactory instance.

newInstance() - Method in interface javax.xml.stream.util.XMLEventAllocator
This method creates an instance of the XMLEventAllocator.

newInstance() - Static method in class javax.xml.stream.XMLEventFactory
Create a new instance of the factory.

newInstance(String, ClassLoader) - Static method in class javax.xml.stream.XMLEventFactory
Create a new instance of the factory.

newInstance() - Static method in class javax.xml.stream.XMLInputFactory
Create a new instance of the factory.

newInstance(String, ClassLoader) - Static method in class javax.xml.stream.XMLInputFactory
Create a new instance of the factory.

newInstance() - Static method in class javax.xml.stream.XMLOutputFactory
Create a new instance of the factory.

newInstance(String, ClassLoader) - Static method in class javax.xml.stream.XMLOutputFactory
Create a new instance of the factory.

newLine() - Method in class javax.servlet.jsp.JspWriter
Write a line separator.

newMessageFactory(String) - Method in class javax.xml.soap.SAAJMetaFactory
Creates a MessageFactory object for the given String protocol.

NewsAddress - Class in javax.mail.internet
This class models an RFC1036 newsgroup address.

NewsAddress() - Constructor for class javax.mail.internet.NewsAddress
Default constructor.

NewsAddress(String) - Constructor for class javax.mail.internet.NewsAddress
Construct a NewsAddress with the given newsgroup.

NewsAddress(String, String) - Constructor for class javax.mail.internet.NewsAddress
Construct a NewsAddress with the given newsgroup and host.

newsgroup - Variable in class javax.mail.internet.NewsAddress
**NEWSGROUPS** - Static variable in class
javax.mail.internet.MimeMessage.RecipientType
The "Newsgroup" (Usenet news) recipients.

**newSOAPFactory(String)** - Method in class
javax.xml.soap.SAAJMetaFactory
Creates a SOAPFactory object for the given String protocol.

**newStream(long, long)** - Method in interface
javax.mail.internet.SharedInputStream
Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).

**newStream(long, long)** - Method in class
javax.mail.util.SharedByteArrayInputStream
Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).

**newStream(long, long)** - Method in class
javax.mail.util.SharedFileInputStream
Return a new InputStream representing a subset of the data from this InputStream, starting at start (inclusive) up to end (exclusive).

**next()** - Method in class javax.mail.internet.HeaderTokenizer
Parses the next token from this String.

**next()** - Method in class javax.xml.stream.util.EventReaderDelegate

**next()** - Method in interface javax.xml.stream.XMLStreamReader
Get the next parsing event - a processor may return all contiguous character data in a single chunk, or it may split it into several chunks.

**nextEvent()** - Method in interface javax.xml.stream.XMLEventReader
Get the next XMLEvent

**nextTag()** - Method in class
javax.xml.stream.util.EventReaderDelegate

**nextTag()** - Method in class
javax.xml.stream.util.StreamReaderDelegate

**nextTag()** - Method in interface javax.xml.stream.XMLEventReader
Skips any insignificant space events until a START_ELEMENT or END_ELEMENT is reached.

**nextTag()** - Method in interface javax.xml.stream.XMLStreamReader
Skips any white space (isWhiteSpace() returns true), COMMENT, or PROCESSING_INSTRUCTION, until a START_ELEMENT or END_ELEMENT is reached.
**nil** - Variable in class javax.xml.bind.JAXBElement
  true iff the xml element instance has xsi:nil="true".

**NO_BUFFER** - Static variable in class javax.servlet.jsp.JspWriter
  Constant indicating that the Writer is not buffering output.

**Node** - Interface in javax.xml.soap
  A representation of a node (element) in an XML document.

**NON_PERSISTENT** - Static variable in interface javax.jms.DeliveryMode
  This is the lowest-overhead delivery mode because it does not require that the message be logged to stable storage.

**NonUniqueResultException** - Exception in javax.persistence
  Thrown by the persistence provider when getSingleResult() is executed on a query and there is more than one result from the query.

**NonUniqueResultException()** - Constructor for exception javax.persistence.NonUniqueResultException
  Constructs a new NonUniqueResultException exception with null as its detail message.

**NonUniqueResultException(String)** - Constructor for exception javax.persistence.NonUniqueResultException
  Constructs a new NonUniqueResultException exception with the specified detail message.

**NoResultException** - Exception in javax.persistence
  Thrown by the persistence provider when getSingleResult() is executed on a query and there is no result to return.

**NoResultException()** - Constructor for exception javax.persistence.NoResultException
  Constructs a new NoResultException exception with null as its detail message.

**NoResultException(String)** - Constructor for exception javax.persistence.NoResultException
  Constructs a new NoResultException exception with the specified detail message.

**NormalizedStringAdapter** - Class in javax.xml.bind.annotation.adapters
  XmlAdapter to handle xs:normalizedString.

**NormalizedStringAdapterFactory()** - Constructor for class javax.xml.bind.annotation.adapters.NormalizedStringAdapter

**normalizeMimeType(String)** - Method in class javax.activation.ActivationDataFlavor
  Deprecated.

**normalizeMimeTypeParameter(String, String)** - Method in class javax.activation.ActivationDataFlavor
  Deprecated.

**NoSuchEJBException** - Exception in javax.ejb
  A NoSuchEJBException is thrown if an attempt is made to invoke a method on an object that no longer exists.

**NoSuchEJBException()** - Constructor for exception javax.ejb.NoSuchEJBException
Constructs a NoSuchEJBException with no detail message.

**NoSuchEJBException(String)** - Constructor for exception

`javax.ejb.NoSuchEJBException`

Constructs a NoSuchEJBException with the specified detail message.

**NoSuchEJBException(String, Exception)** - Constructor for exception

`javax.ejb.NoSuchEJBException`

Constructs a NoSuchEJBException with the specified detail message and a nested exception.

**NoSuchEntityException** - Exception in `javax.ejb`

The NoSuchEntityException exception is thrown by an Entity Bean instance to its container to report that the invoked business method or callback method could not be completed because of the underlying entity was removed from the database.

**NoSuchEntityException()** - Constructor for exception

`javax.ejb.NoSuchEntityException`

Constructs a NoSuchEntityException with no detail message.

**NoSuchEntityException(String)** - Constructor for exception

`javax.ejb.NoSuchEntityException`

Constructs a NoSuchEntityException with the specified detailed message.

**NoSuchEntityException(Exception)** - Constructor for exception

`javax.ejb.NoSuchEntityException`

Constructs a NoSuchEntityException that embeds the originally thrown exception.

**NoSuchObjectLocalException** - Exception in `javax.ejb`

A NoSuchObjectLocalException is thrown if an attempt is made to invoke a method on an object that no longer exists.

**NoSuchObjectLocalException()** - Constructor for exception

`javax.ejb.NoSuchObjectLocalException`

Constructs a NoSuchObjectLocalException with no detail message.

**NoSuchObjectLocalException(String)** - Constructor for exception

`javax.ejb.NoSuchObjectLocalException`

Constructs a NoSuchObjectLocalException with the specified detail message.

**NoSuchObjectLocalException(String, Exception)** - Constructor for exception

`javax.ejb.NoSuchObjectLocalException`

Constructs a NoSuchObjectLocalException with the specified detail message and a nested exception.

**NoSuchProviderException** - Exception in `javax.mail`

This exception is thrown when Session attempts to instantiate a Provider that doesn't exist.

**NoSuchProviderException()** - Constructor for exception

`javax.mail.NoSuchProviderException`

Constructor.

**NoSuchProviderException(String)** - Constructor for exception

`javax.mail.NoSuchProviderException`

Constructor.

**NOT_IN_RANGE_MESSAGE_ID** - Static variable in class

`javax.faces.validator.DoubleRangeValidator`
The message identifier of the `FacesMessage` to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set.

**NOT IN RANGE MESSAGE ID** - Static variable in class `javax.faces.validator.LongRangeValidator`

The message identifier of the `FacesMessage` to be created if the maximum or minimum value check fails, and both the maximum and minimum values for this validator have been set.

**NOT IN RANGE MESSAGE ID** - Static variable in interface `javax.faces.validator.Validator`

*Deprecated.* Use `DoubleRangeValidator.NOT_IN_RANGE_MESSAGE_ID` or `LongRangeValidator.NOT_IN_RANGE_MESSAGE_ID` instead.

**NOTATION DECLARATION** - Static variable in interface `javax.xml.stream.XMLStreamConstants`

Indicates a Notation Declaration - Interface in `javax.xml.stream.events`

An interface for handling Notation Declarations. Receive notification of a notation declaration event.

**NOTICE** - Static variable in class `javax.mail.event.StoreEvent`

Indicates that this message is a NOTICE.

**NotIdentifiableEvent** - Interface in `javax.xml.bind`

This event indicates that a problem was encountered resolving an ID/IDREF.

**NotIdentifiableEventImpl** - Class in `javax.xml.bind.helpers`

Default implementation of the NotIdentifiableEvent interface.

**NotIdentifiableEventImpl(int, String, ValidationEventLocator)** - Constructor for class `javax.xml.bind.helpers.NotIdentifiableEventImpl`

Create a new NotIdentifiableEventImpl.

**NotIdentifiableEventImpl(int, String, ValidationEventLocator, Throwable)** - Constructor for class `javax.xml.bind.helpers.NotIdentifiableEventImpl`

Create a new NotIdentifiableEventImpl.

**notification(StoreEvent)** - Method in interface `javax.mail.event.StoreListener`

Invoked when the Store generates a notification event.

**notifyConnectionListeners(int)** - Method in class `javax.mail.Folder`

Notify all ConnectionListeners.

**notifyConnectionListeners(int)** - Method in class `javax.mail.Service`

Notify all ConnectionListeners.

**notifyDDChange(XpathEvent)** - Method in interface `javax.enterprise.deploy.spi.DConfigBean`

A notification that the DDBean provided in the event has changed and this bean or its child beans need to reevaluate themselves.

**notifyFolderListeners(int)** - Method in class `javax.mail.Folder`

Notify all FolderListeners registered on this Folder and this folder's Store.

**notifyFolderListeners(int, Folder)** - Method in class `javax.mail.Store`
Notify all FolderListeners.

**notifyFolderRenamedListeners(Folder)** - Method in class
javax.mail.Folder

Notify all FolderListeners registered on this Folder and this folder's Store about the renaming of this folder.

**notifyFolderRenamedListeners(Folder, Folder)** - Method in class
javax.mail.Store

Notify all FolderListeners about the renaming of a folder.

**notifyMessageAddedListeners(Message[])** - Method in class
javax.mail.Folder

Notify all MessageCountListeners about the addition of messages into this folder.

**notifyMessageChangedListeners(int, Message)** - Method in class
javax.mail.Folder

Notify all MessageChangedListeners.

**notifyMessageRemovedListeners(boolean, Message[])** - Method in class
javax.mail.Folder

Notify all MessageCountListeners about the removal of messages from this folder.

**notifyStoreListeners(int, String)** - Method in class
javax.mail.Store

Notify all StoreListeners.

**notifyTransportListeners(int, Address[], Address[], Address[], Message)** - Method in class javax.mail.Transport

Notify all TransportListeners.

**NotSupportedException** - Exception in javax.resource

A NotSupportedException is thrown to indicate that callee (resource adapter or application server for system contracts) cannot execute an operation because the operation is not a supported feature.

**NotSupportedException()** - Constructor for exception
javax.resource.NotSupportedException

Constructs a new instance with null as its detail message.

**NotSupportedException(String)** - Constructor for exception
javax.resource.NotSupportedException

Constructs a new instance with the specified detail message.

**NotSupportedException(ThrowVariable)** - Constructor for exception
javax.resource.NotSupportedException

Constructs a new throwable with the specified cause.

**NotSupportedException(String, Throwable)** - Constructor for exception
javax.resource.NotSupportedException

Constructs a new throwable with the specified detail message and cause.

**NotSupportedException(String, String)** - Constructor for exception
javax.resource.NotSupportedException

Constructs a new throwable with the specified detail message and error code.

**NotSupportedException** - Exception in javax.transaction

NotSupportedException exception indicates that the request cannot be executed because the operation is not a supported
NotSupportedException() - Constructor for exception
javax.transaction.NotSupportedException

NotSupportedException(String) - Constructor for exception
javax.transaction.NotSupportedException

NotTerm - Class in javax.mail.search
This class implements the logical NEGATION operator.
NotTerm(SearchTerm) - Constructor for class
javax.mail.search.NotTerm

NSPREFIX_SCHEMA_XSD - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace prefix for XML schema XSD
NSPREFIX_SCHEMA_XSI - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace prefix for XML Schema XSI
NSPREFIX_SOAP_ENCODING - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace prefix for SOAP Encoding
NSPREFIX_SOAP_ENVELOPE - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace prefix for SOAP Envelope
NSURI_SCHEMA_XSD - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace URI for XML Schema XSD
NSURI_SCHEMA_XSI - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace URI for XML Schema XSI
NSURI_SOAP_ENCODING - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace URI for SOAP 1.1 Encoding
NSURI_SOAP_ENVELOPE - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace URI for SOAP 1.1 Envelope
NSURI_SOAP_NEXT_ACTOR - Static variable in class
javax.xml.rpc.NamespaceConstants
  Namespace URI for SOAP 1.1 next actor role
number - Variable in class javax.mail.search.IntegerComparisonTerm
  The number.
NUMBER_ID - Static variable in class
javax.faces.convert.NumberConverter
  The message identifier of the FacesMessage to be created if the conversion to Number fails.
NumberConverter - Class in javax.faces.convert
  Converter implementation for java.lang.Number values.
NumberConverter() - Constructor for class
javax.faces.convert.NumberConverter
**ObjectHolder** - Class in *javax.xml.rpc.holders*

**ObjectHolder()** - Constructor for class
*javax.xml.rpc.holders.ObjectHolder*

**ObjectHolder(Object)** - Constructor for class
*javax.xml.rpc.holders.ObjectHolder*

**ObjectMessage** - Interface in *javax.jms*

An ObjectMessage object is used to send a message that contains a serializable object in the Java programming language ("Java object").

**ObjectNotFoundException** - Exception in *javax.ejb*

The ObjectNotFoundException exception is thrown by a finder method to indicate that the specified EJB object does not exist.

**ObjectNotFoundException()** - Constructor for exception
*javax.ejb.ObjectNotFoundException*

Constructs an ObjectNotFoundException with no detail message.

**ObjectNotFoundException(String)** - Constructor for exception
*javax.ejb.ObjectNotFoundException*

Constructs an ObjectNotFoundException with the specified detail message.

**OneToMany** - Annotation Type in *javax.persistence*

Defines a many-valued association with one-to-many multiplicity.

**OneToOne** - Annotation Type in *javax.persistence*

This annotation defines a single-valued association to another entity that has one-to-one multiplicity.

**Oneway** - Annotation Type in *javax.jws*

Indicates that the given @WebMethod has only an input message and no output.

**onException(JMSEException)** - Method in interface
*javax.jms.ExceptionListener*

Notifies user of a JMS exception.

**onMessage(Message)** - Method in interface *javax.jms.MessageListener*

Passes a message to the listener.

**onMessage(Record)** - Method in interface
*javax.resource.cci.MessageListener*

This method allows an EIS to call a message endpoint using a request-response style communication.

**open(int)** - Method in class *javax.mail.Folder*

Open this Folder.

**opened(ConnectionEvent)** - Method in class
*javax.mail.event.ConnectionAdapter*
**OPENED** - Static variable in class `javax.mail.event.ConnectionEvent`
A connection was opened.

**opened(ConnectionEvent)** - Method in interface `javax.mail.event.ConnectionListener`
Invoked when a Store/Folder/Transport is opened.

**OPERATION_STYLE_PROPERTY** - Static variable in interface `javax.xml.rpc.Call`
Standard property for operation style.

**OperationUnsupportedException** - Exception in `javax.enterprise.deploy.spi.exceptions`
This exception is to report that the method called is not supported by this implementation.

**OperationUnsupportedException(String)** - Constructor for exception `javax.enterprise.deploy.spi.exceptions.OperationUnsupportedException`
Creates a new OperationUnsupportedException object.

**OPERATOR SLOT** - Static variable in interface `javax.xml.registry.infomodel.Slot`
Name for pre-defined Slot used in Organization and ClassificationScheme by JAXR UDDI provider.

**OptimisticLockException** - Exception in `javax.persistence`
Thrown by the persistence provider when an optimistic locking conflict occurs.

**OptimisticLockException()** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with null as its detail message.

**OptimisticLockException(String)** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with the specified detail message.

**OptimisticLockException(String, Throwable)** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with the specified detail message and cause.

**OptimisticLockException(Throwables)** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with the specified cause.

**OptimisticLockException(Object)** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with the specified entity.

**OptimisticLockException(String, Throwable, Object)** - Constructor for exception `javax.persistence.OptimisticLockException`
Constructs a new OptimisticLockException exception with the specified detail message, cause, and entity.

**OR_ALL_KEYS** - Static variable in interface `javax.xml.registry.FindQualifier`

**OR LIKE KEYS** - Static variable in interface
javax.xml.registry.FindQualifier

**OrderBy** - Annotation Type in [javax.persistence](https://javaee.oracle.com/javaee-8/docs/api/javax/persistence/package-summary.html)
This annotation specifies the ordering of the elements of a collection valued association at the point when the association is retrieved.

Organization instances provide information on organizations such as a Submitting Organization.

**ORGANIZATION** - Static variable in interface javax.xml.registry.[LifeCycleManager](https://javaee.oracle.com/javaee-8/docs/api/javax/xml/registry/LifeCycleManager.html)

**OrTerm** - Class in [javax.mail.search](https://javaee.oracle.com/javaee-8/docs/api/javax/mail/search/package-summary.html)
This class implements the logical OR operator on individual SearchTerms.

**OrTerm(SearchTerm, SearchTerm)** - Constructor for class javax.mail.search.[OrTerm](https://javaee.oracle.com/javaee-8/docs/api/javax/mail/search/OrTerm.html)
Constructor that takes two operands.

**OrTerm(SearchTerm[])** - Constructor for class javax.mail.search.[OrTerm](https://javaee.oracle.com/javaee-8/docs/api/javax/mail/search/OrTerm.html)
Constructor that takes an array of SearchTerms.

**othersDeletesAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)
Indicates whether deletes made by others are visible.

**othersInsertsAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)
Indicates whether inserts made by others are visible.

**othersUpdatesAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)
Indicates whether updates made by others are visible.

**OUT** - Static variable in class javax.servlet.jsp.[PageContext](https://javaee.oracle.com/javaee-8/docs/api/javax/servlet/jsp/PageContext.html)
Name used to store current JspWriter in PageContext name table.

**OUT** - Static variable in class javax.xml.rpc.[ParameterMode](https://javaee.oracle.com/javaee-8/docs/api/javax/xml/rpc/ParameterMode.html)
OUT mode for parameter passing

**OUTBOUND_MESSAGE_ATTACHMENTS** - Static variable in interface javax.xml.ws.handler.[MessageContext](https://javaee.oracle.com/javaee-8/docs/api/javax/xml/ws/handler/MessageContext.html)
Standard property: Map of attachments to a message for the outbound message, key is the MIME Content-ID, value is a DataHandler.

**ownDeletesAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)

**ownInsertsAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)

**ownUpdatesAreVisible(int)** - Method in interface javax.resource.cci.[ResultSetInfo](https://javaee.oracle.com/javaee-8/docs/api/javax/resource/cci/ResultSetInfo.html)
PAGE - Static variable in class javax.servlet.jsp.PageContext
Name used to store the Servlet in this PageContext's nametables.

PAGE_SCOPE - Static variable in class javax.servlet.jsp.PageContext
Page scope: (this is the default) the named reference remains available in this PageContext until the return from the current Servlet.service() invocation.

pageContext - Variable in class javax.faces.webapp.UIComponentClassicTagBase
The JSP PageContext for the page we are embedded in.

PageContext - Class in javax.servlet.jsp
PageContext extends JspContext to provide useful context information for when JSP technology is used in a Servlet environment.

PageContext() - Constructor for class javax.servlet.jsp.PageContext
Sole constructor.

PAGECONTEXT - Static variable in class javax.servlet.jsp.PageContext
Name used to store this PageContext in it's own name table.

pageContext - Variable in class javax.servlet.jsp.tagext.TagSupport
The PageContext.

PageData - Class in javax.servlet.jsp.tagext
Translation-time information on a JSP page.

PageData() - Constructor for class javax.servlet.jsp.tagext.PageData
Sole constructor.

ParameterList - Class in javax.mail.internet
This class holds MIME parameters (attribute-value pairs).

ParameterList() - Constructor for class javax.mail.internet.ParameterList
No-arg Constructor.

ParameterList(String) - Constructor for class javax.mail.internet.ParameterList
Constructor that takes a parameter-list string.

ParameterMode - Class in javax.xml.rpc
The javax.xml.rpc.ParameterMode is a type-safe enumeration for parameter mode.

parent - Variable in class javax.mail.BodyPart
The Multipart object containing this BodyPart, if known.

parent - Variable in class javax.mail.Multipart
The Part containing this Multipart, if known.

parse(String) - Method in class javax.activation.MimeTypeParameterList
A routine for parsing the parameter list out of a String.

parse(String) - Static method in class javax.mail.internet.InternetAddress
Parse the given comma separated sequence of addresses into InternetAddress objects.

**parse(String, boolean)** - Static method in class
javax.mail.internet.InternetAddress
Parse the given sequence of addresses into InternetAddress objects.

**parse(String, ParsePosition)** - Method in class
javax.mail.internet.MailDateFormat
Parses the given date in the format specified by draft-ietf-drums-msg-fmt-08 in the current TimeZone.

**parse(InputStream)** - Method in class
javax.mail.internet.MimeMessage
Parse the InputStream setting the headers and content fields appropriately.

**parse()** - Method in class javax.mail.internet.MimeMultipart
Parse the InputStream from our DataSource, constructing the appropriate MimeBodyParts.

**parse(String)** - Static method in class
javax.mail.internet.NewsAddress
Parse the given comma separated sequence of newsgroup into NewsAddress objects.

**parseAnySimpleType(String)** - Static method in class
javax.xml.bind.DatatypeConverter
Return a string containing the lexical representation of the simple type.

**parseAnySimpleType(String)** - Method in interface
javax.xml.bind.DatatypeConverterInterface
Return a string containing the lexical representation of the simple type.

**parseBase64Binary(String)** - Static method in class
javax.xml.bind.DatatypeConverter
Converts the string argument into an array of bytes.

**parseBase64Binary(String)** - Method in interface
javax.xml.bind.DatatypeConverterInterface
Converts the string argument into an array of bytes.

**parseBoolean(String)** - Static method in class
javax.xml.bind.DatatypeConverter
Converts the string argument into a boolean value.

**parseBoolean(String)** - Method in interface
javax.xml.bind.DatatypeConverterInterface
Converts the string argument into a boolean value.

**parseByte(String)** - Static method in class
javax.xml.bind.DatatypeConverter
Converts the string argument into a byte value.

**parseByte(String)** - Method in interface
javax.xml.bind.DatatypeConverterInterface
Converts the string argument into a byte value.

**ParseConversionEvent** - Interface in javax.xml.bind
This event indicates that a problem was encountered while converting a string from the XML data into a value of the
target Java data type.

**ParseConversionEventImpl** - Class in [javax.xml.bind.helpers](https://docs.oracle.com/javaee/8/api/javax/xml/bind/helpers/index.html)

Default implementation of the ParseConversionEvent interface.

**ParseConversionEventImpl(int, String, ValidationEventLocator)** - Constructor for class

javax.xml.bind.helpers.ParseConversionEventImpl

Create a new ParseConversionEventImpl.

**ParseConversionEventImpl(int, String, ValidationEventLocator, Throwable)** - Constructor for class

javax.xml.bind.helpers.ParseConversionEventImpl

Create a new ParseConversionEventImpl.

**parsed** - Variable in class [javax.mail.internet.MimeMultipart](https://docs.oracle.com/javase/8/docs/api/javax/mail/internet/MimeMultipart.html)

Have we parsed the data from our InputStream yet?

**parseDate(String)** - Static method in class

javax.xml.bind.DatatypeConverter

Converts the string argument into a Calendar value.

**parseDate(String)** - Method in interface

javax.xml.bind.DatatypeConverterInterface

Converts the string argument into a Calendar value.

**parseDateTime(String)** - Static method in class

javax.xml.bind.DatatypeConverter

Converts the string argument into a Calendar value.

**parseDateTime(String)** - Method in interface

javax.xml.bind.DatatypeConverterInterface

Converts the string argument into a Calendar value.

**parseDecimal(String)** - Static method in class

javax.xml.bind.DatatypeConverter

Converts the string argument into a BigDecimal value.

**parseDecimal(String)** - Method in interface

javax.xml.bind.DatatypeConverterInterface

Converts the string argument into a BigDecimal value.

**parseDouble(String)** - Static method in class

javax.xml.bind.DatatypeConverter

Converts the string argument into a double value.

**parseDouble(String)** - Method in interface

javax.xml.bind.DatatypeConverterInterface

Converts the string argument into a double value.

**ParseException** - Exception in [javax.mail.internet](https://docs.oracle.com/javase/8/docs/api/javax/mail/internet/index.html)

The exception thrown due to an error in parsing RFC822 or MIME headers

**ParseException()** - Constructor for exception

javax.mail.internet.ParseException

Constructs a ParseException with no detail message.

**ParseException(String)** - Constructor for exception

javax.mail.internet.ParseException

Constructs a ParseException with the specified detail message.

**parseExpression(String, Class, FunctionMapper)** - Method in class

javax.servlet.jsp.el.ExpressionEvaluator

Deprecated. Prepare an expression for later evaluation.

**parseFloat(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into a float value.

**parseFloat(String)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into a float value.

**parseHeader(String, boolean)** - Static method in class
javax.mail.internet.**InternetAddress**
Parse the given sequence of addresses into InternetAddress objects.

**parseHexBinary(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into an array of bytes.

**parseHexBinary(String)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into an array of bytes.

**parseInt(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into an int value.

**parseInt(String)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into an int value.

**parseInteger(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into a BigInteger value.

**parseInteger(String)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into a BigInteger value.

**parseLong(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into a long value.

**parseLong(String)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into a long value.

**parsePostData(int, ServletInputStream)** - Static method in class
javax.servlet.http.**HttpUtils**
*Deprecated.* Parses data from an HTML form that the client sends to the server using the HTTP POST method and the application/x-www-form-urlencoded MIME type.

**parseQName(String, NamespaceContext)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into a byte value.

**parseQName(String, NamespaceContext)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts the string argument into a QName value.

**parseQueryString(String)** - Static method in class
javax.servlet.http.**HttpUtils**
*Deprecated.* Parses a query string passed from the client to the server and builds a HashTable object with key-value pairs.

**parseShort(String)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts the string argument into a short value.

**parseShort(String)** - Method in interface javax.xml.bind.DatatypeConverterInterface
Converts the string argument into a short value.

**parseString(String)** - Method in interface javax.xml.bind.URLName
Method which does all of the work of parsing the string.

**parseString(String)** - Static method in class javax.xml.bind.DatatypeConverter
Convert the lexical XSD string argument into a String value.

**parseString(String)** - Static method in class javax.xml.bind.DatatypeConverterInterface
Convert the string argument into a string.

**parseTime(String)** - Static method in class javax.xml.bind.DatatypeConverter
Converts the string argument into a Calendar value.

**parseTime(String)** - Method in interface javax.xml.bind.DatatypeConverterInterface
Converts the string argument into a Calendar value.

**parseUnsignedInt(String)** - Static method in class javax.xml.bind.DatatypeConverter
Converts the string argument into a long value.

**parseUnsignedInt(String)** - Method in interface javax.xml.bind.DatatypeConverterInterface
Converts the string argument into a long value.

**parseUnsignedShort(String)** - Static method in class javax.xml.bind.DatatypeConverter
Converts the string argument into an int value.

**parseUnsignedShort(String)** - Method in interface javax.xml.bind.DatatypeConverterInterface
Converts the string argument into an int value.

**part** - Variable in class javax.mail.internet.MimePartDataSource
The MimePart that provides the data for this DataSource.

**Part** - Interface in javax.mail
The Part interface is the common base interface for Messages and BodyParts.

**parts** - Variable in class javax.mail.Multipart
Vector of BodyPart objects.

**PASSWORD_PROPERTY** - Static variable in interface java.xml.rpc.Call
Standard property: Password for authentication Type: java.lang.String

**PASSWORD_PROPERTY** - Static variable in interface java.xml.rpc.Stub
Standard property: Password for authentication.

**PASSWORD_PROPERTY** - Static variable in interface javax.xml.ws.BindingProvider
Standard property: Password for authentication.

**PasswordAuthentication** - Class in javax.mail
The class PasswordAuthentication is a data holder that is used by Authenticator.

**PasswordAuthentication(String, String)** - Constructor for class javax.mail.PasswordAuthentication
Initialize a new PasswordAuthentication

**PasswordCredential** - Class in *javax.resource.spi.security*
The class PasswordCredential acts as a holder for username and password.

**PasswordCredential(String, char[])** - Constructor for class
*javax.resource.spi.security.PasswordCredential*
Creates a new PasswordCredential object from the given user name and password.

**PATH_INFO** - Static variable in interface
*javax.xml.ws.handler.MessageContext*
Standard property: Request Path Info Type: String

**pattern** - Variable in class *javax.mail.search.StringTerm*
The pattern.

**PATTERN_ID** - Static variable in class
*javax.faces.convert.NumberConverter*
The message identifier of the *FacesMessage* to be created if the conversion to Number fails.

**peek()** - Method in class *javax.mail.internet.HeaderTokenizer*
Peek at the next token, without actually removing the token from the parse stream.

**peek()** - Method in class *javax.xml.stream.util.EventReaderDelegate*
Check the next XMLEvent without reading it from the stream.

**PERCENT_ID** - Static variable in class
*javax.faces.convert.NumberConverter*
The message identifier of the *FacesMessage* to be created if the conversion to Number fails.

**PermitAll** - Annotation Type in *javax.annotation.security*
Specifies that all security roles are allowed to invoke the specified method(s) i.e that the specified method(s) are "unchecked".

**persist(Object)** - Method in interface
*javax.persistence.EntityManager*
Make an entity instance managed and persistent.

**Persistence** - Class in *javax.persistence*
Bootstrap class that is used to obtain an *EntityManagerFactory*.

**Persistence()** - Constructor for class *javax.persistence.Persistence*

**PERSISTENCE_PROVIDER** - Static variable in class
*javax.persistence.Persistence*

**PersistenceContext** - Annotation Type in *javax.persistence*
Expresses a dependency on an *EntityManager* persistence context.

**PersistenceContexts** - Annotation Type in *javax.persistence*
Declares one or more *PersistenceContext* annotations.

**PersistenceContextType** - Enum in *javax.persistence*
Specifies whether a transaction-scoped or extended persistence context is to be used in *PersistenceContext*.

**PersistenceException** - Exception in *javax.persistence*
Thrown by the persistence provider when a problem occurs.

**PersistenceException()** - Constructor for exception
```
java.lang.Exception
```
Constructs a new PersistenceException exception with null as its detail message.

**PersistenceException(String)** - Constructor for exception
```
java.lang.Exception
```
Constructs a new PersistenceException exception with the specified detail message.

**PersistenceException(String, Throwable)** - Constructor for exception
```
java.lang.Exception
```
Constructs a new PersistenceException exception with the specified detail message and cause.

**PersistenceException(Throwables)** - Constructor for exception
```
java.lang.Exception
```
Constructs a new PersistenceException exception with the specified cause.

**PersistenceProperty** - Annotation Type in `javax.persistence`
Describes a single container or persistence provider property.

**PersistenceProvider** - Interface in `javax.persistence.spi`
Interface implemented by a persistence provider.

**PersistenceUnit** - Annotation Type in `javax.persistence`
Expresses a dependency on an `EntityManagerFactory`.

**PersistenceUnitInfo** - Interface in `javax.persistence.spi`
Interface implemented by the container and used by the persistence provider when creating an `EntityManagerFactory`.

**PersistenceUnits** - Annotation Type in `javax.persistence`
Declares one or more `PersistenceUnit` annotations.

**PersistenceUnitTransactionType** - Enum in `javax.persistence.spi`
This enum class defines whether the entity managers created by the `EntityManagerFactory` will be JTA or resource-local entity managers.

**PERSISTENT** - Static variable in interface `javax.jms.DeliveryMode`
This delivery mode instructs the JMS provider to log the message to stable storage as part of the client's send operation.

**PERSON_NAME** - Static variable in interface
```
javax.xml.registry.LifeCycleManager
```

**personal** - Variable in class `javax.mail.internet.InternetAddress`
The personal name.

**PersonName** - Interface in `javax.xml.registry.infomodel`
Represents a person's name.

**PhaseEvent** - Class in `javax.faces.event`
`PhaseEvent` represents the beginning or ending of processing for a particular phase of the request processing lifecycle, for the request encapsulated by the specified `FacesContext`.

**PhaseEvent(FacesContext, PhaseId, Lifecycle)** - Constructor for class `javax.faces.event.PhaseEvent`
Construct a new event object from the specified parameters.
PhaseId - Class in javax.faces.event
Typesafe enumeration of the legal values that may be returned by the getPhaseId() method of the FacesEvent interface.

PhaseListener - Interface in javax.faces.event
An interface implemented by objects that wish to be notified at the beginning and ending of processing for each standard phase of the request processing lifecycle.

PolicyConfiguration - Interface in javax.security.jacc
The methods of this interface are used by containers to create policy statements in a Policy provider.

PolicyConfigurationFactory - Class in javax.security.jacc
Abstract factory and finder class for obtaining the instance of the class that implements the PolicyConfigurationFactory of a provider.

PolicyConfigurationFactory() - Constructor for class javax.security.jacc.PolicyConfigurationFactory

PolicyContext - Class in javax.security.jacc
This utility class is used by containers to communicate policy context identifiers and other policy relevant context to Policy providers.

PolicyContextException - Exception in javax.security.jacc
This checked exception is thrown by implementations of the javax.security.jacc.PolicyConfiguration Interface, the javax.security.jacc.PolicyConfigurationFactory abstract class, the javax.security.jacc.PolicyContext utility class, and implementations of the javax.security.jacc.PolicyContextException Interface.

PolicyContextException() - Constructor for exception javax.security.jacc.PolicyContextException
Constructs a new PolicyContextException with null as its detail message.

PolicyContextException(String) - Constructor for exception javax.security.jacc.PolicyContextException
Constructs a new PolicyContextException with the specified detail message

PolicyContextException(String, Throwable) - Constructor for exception javax.security.jacc.PolicyContextException
Constructs a new PolicyContextException with the specified detail message and cause.

PolicyContextException(Throwables) - Constructor for exception javax.security.jacc.PolicyContextException
Constructs a new PolicyContextException with the specified cause.

PolicyContextHandler - Interface in javax.security.jacc
This interface defines the methods that must be implemented by handlers that are to be registered and activated by the PolicyContext class.

popBody() - Method in class javax.servlet.jsp.JspContext
Return the previous JspWriter "out" saved by the matching
pushBody(), and update the value of the "out" attribute in the page scope attribute namespace of the JspContext.

**PortInfo** - Interface in [javax.xml.ws.handler](https://docs.oracle.com/javase/10/docs/api/javax/xml/ws/handler/PortInfo.html)
The PortInfo interface is used by a HandlerResolver to query information about the port it is being asked to create a handler chain for.

**pos** - Variable in exception [javax.mail.internet.AddressException](https://docs.oracle.com/javase/10/docs/api/javax/mail/internet/AddressException.html)
The index in the string where the error occurred, or -1 if not known.

**PostActivate** - Annotation Type in [javax.ejb](https://docs.oracle.com/javase/10/docs/api/javax/ejb/)
Designates a method to receive a callback after a stateful session bean has been activated.

**POSTAL_ADDRESS** - Static variable in interface [javax.xml.registry.LifeCycleManager](https://docs.oracle.com/javase/10/docs/api/javax/xml/registry/LifeCycleManager.html)

**PostalAddress** - Interface in [javax.xml.registry.infomodel](https://docs.oracle.com/javase/10/docs/api/javax/xml/registry/infomodel/PostalAddress.html)
PostalAddress is a simple re-usable entity class that defines attributes of a postal Address.

**PostConstruct** - Annotation Type in [javax.annotation](https://docs.oracle.com/javase/10/docs/api/javax/annotation/PostConstruct.html)
The PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.

**PostLoad** - Annotation Type in [javax.persistence](https://docs.oracle.com/javase/10/docs/api/javax/persistence/)
Is used to specify callback methods for the corresponding lifecycle event.

**PostPersist** - Annotation Type in [javax.persistence](https://docs.oracle.com/javase/10/docs/api/javax/persistence/)
Is used to specify callback methods for the corresponding lifecycle event.

**PostRemove** - Annotation Type in [javax.persistence](https://docs.oracle.com/javase/10/docs/api/javax/persistence/)
Is used to specify callback methods for the corresponding lifecycle event.

**PostUpdate** - Annotation Type in [javax.persistence](https://docs.oracle.com/javase/10/docs/api/javax/persistence/)
Is used to specify callback methods for the corresponding lifecycle event.

**PreDestroy** - Annotation Type in [javax.annotation](https://docs.oracle.com/javase/10/docs/api/javax/annotation/PreDestroy.html)
The PreDestroy annotation is used on methods as a callback notification to signal that the instance is in the process of being removed by the container.

**PreencodedMimeBodyPart** - Class in [javax.mail.internet](https://docs.oracle.com/javase/10/docs/api/javax/mail/internet/PreencodedMimeBodyPart.html)
A MimeBodyPart that handles data that has already been encoded.

**PreencodedMimeBodyPart(String)** - Constructor for class [javax.mail.internet.PreencodedMimeBodyPart](https://docs.oracle.com/javase/10/docs/api/javax/mail/internet/PreencodedMimeBodyPart.html)
Create a PreencodedMimeBodyPart that assumes the data is encoded using the specified encoding.

**prefix** - Variable in class [javax.servlet.jsp.tagext.TagLibraryInfo](https://docs.oracle.com/javase/10/docs/api/javax/servlet/jsp/tagext/TagLibraryInfo.html)
The prefix assigned to this taglib from the taglib directive.

**prepare(Xid)** - Method in interface [javax.resource.spi.XATerminator](https://docs.oracle.com/javase/10/docs/api/javax/route/spi/XATerminator.html)
Ask the resource manager to prepare for a transaction commit of the transaction specified in xid.

**prepare(Xid)** - Method in interface [javax.transaction.xa.XAResource](https://docs.oracle.com/javase/10/docs/api/javax/transaction/xa/XAResource.html)
Ask the resource manager to prepare for a transaction commit of the transaction specified in xid.

**PrePassivate** - Annotation Type in *javax.ejb*
Designates a method to receive a callback before a stateful session bean is passivated.

**PrePersist** - Annotation Type in *javax.persistence*
Is used to specify callback methods for the corresponding lifecycle event.

**PreRemove** - Annotation Type in *javax.persistence*
Is used to specify callback methods for the corresponding lifecycle event.

**PreUpdate** - Annotation Type in *javax.persistence*
Is used to specify callback methods for the corresponding lifecycle event.

**PrimaryKeyJoinColumn** - Annotation Type in *javax.persistence*
This annotation specifies a primary key column that is used as a foreign key to join to another table.

**PrimaryKeyJoinColumn** - Annotation Type in *javax.persistence*
This annotation groups `PrimaryKeyJoinColumn` annotations.

**print(boolean)** - Method in class *javax.servlet.jsp.JspWriter*
Print a boolean value.

**print(char)** - Method in class *javax.servlet.jsp.JspWriter*
Print a character.

**print(int)** - Method in class *javax.servlet.jsp.JspWriter*
Print an integer.

**print(long)** - Method in class *javax.servlet.jsp.JspWriter*
Print a long integer.

**print(float)** - Method in class *javax.servlet.jsp.JspWriter*
Print a floating-point number.

**print(double)** - Method in class *javax.servlet.jsp.JspWriter*
Print a double-precision floating-point number.

**print(char[])** - Method in class *javax.servlet.jsp.JspWriter*
Print an array of characters.

**print(String)** - Method in class *javax.servlet.jsp.JspWriter*
Print a string.

**print(Object)** - Method in class *javax.servlet.jsp.JspWriter*
Print an object.

**print(String)** - Method in class *javax.servlet.ServletOutputStream*
Writes a String to the client, without a carriage return-line feed (CRLF) character at the end.

**print(boolean)** - Method in class *javax.servlet.ServletOutputStream*
Writes a boolean value to the client, with no carriage return-line feed (CRLF) character at the end.

**print(char)** - Method in class *javax.servlet.ServletOutputStream*
Writes a character to the client, with no carriage return-line feed (CRLF) at the end.

**print(int)** - Method in class *javax.servlet.ServletOutputStream*
Writes an int to the client, with no carriage return-line feed (CRLF) at the end.

**print(long)** - Method in class *javax.servlet.ServletOutputStream*
Writes a long value to the client, with no carriage return-line feed (CRLF) at the end.

**print(float)** - Method in class `javax.servlet.ServletOutputStream`
Writes a float value to the client, with no carriage return-line feed (CRLF) at the end.

**print(double)** - Method in class `javax.servlet.ServletOutputStream`
Writes a double value to the client, with no carriage return-line feed (CRLF) at the end.

**printAnySimpleType(String)** - Static method in class `javax.xml.bind.DatatypeConverter`
Converts a string value into a string.

**printAnySimpleType(String)** - Method in interface `javax.xml.bind.DatatypeConverterInterface`
Converts a string value into a string.

**printBase64Binary(byte[])** - Static method in class `javax.xml.bind.DatatypeConverter`
Converts an array of bytes into a string.

**printBase64Binary(byte[])** - Method in interface `javax.xml.bind.DatatypeConverterInterface`
Converts an array of bytes into a string.

**printBoolean(boolean)** - Static method in class `javax.xml.bind.DatatypeConverter`
Converts a boolean value into a string.

**printBoolean(boolean)** - Method in interface `javax.xml.bind.DatatypeConverterInterface`
Converts a boolean value into a string.

**printByte(byte)** - Static method in class `javax.xml.bind.DatatypeConverter`
Converts a byte value into a string.

**printByte(byte)** - Method in interface `javax.xml.bind.DatatypeConverterInterface`
Converts a byte value into a string.

**PrintConversionEvent** - Interface in `javax.xml.bind`
This event indicates that a problem was encountered while converting data from the Java content tree into its lexical representation.

**PrintConversionEventImpl** - Class in `javax.xml.bind.helpers`
Default implementation of the PrintConversionEvent interface.

**PrintConversionEventImpl(int, String, ValidationEventLocator)** - Constructor for class `javax.xml.bind.helpers.PrintConversionEventImpl`
Create a new PrintConversionEventImpl.

**PrintConversionEventImpl(int, String, ValidationEventLocator, Throwable)** - Constructor for class `javax.xml.bind.helpers.PrintConversionEventImpl`
Create a new PrintConversionEventImpl.

**printDate(Calendar)** - Static method in class `javax.xml.bind.DatatypeConverter`
Converts a Calendar value into a string.

**printDate(Calendar)** - Method in interface
javax.xml.bind.**DatatypeConverter**
Converts a Calendar value into a string.

**printDateTime(Calendar)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts a Calendar value into a string.

**printDateTime(Calendar)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts a Calendar value into a string.

**printDecimal(BigDecimal)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts a BigDecimal value into a string.

**printDecimal(BigDecimal)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts a BigDecimal value into a string.

**printDouble(double)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts a double value into a string.

**printDouble(double)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts a double value into a string.

**printFloat(float)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts a float value into a string.

**printFloat(float)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts a float value into a string.

**printHexBinary(byte[])** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts an array of bytes into a string.

**printHexBinary(byte[])** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts an array of bytes into a string.

**printInt(int)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts an int value into a string.

**printInt(int)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts an int value into a string.

**printInteger(BigInteger)** - Static method in class
javax.xml.bind.**DatatypeConverter**
Converts a BigInteger value into a string.

**printInteger(BigInteger)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**
Converts a BigInteger value into a string.

**println()** - Method in class javax.servlet.jsp.**JspWriter**
Terminate the current line by writing the line separator string.

**println(boolean)** - Method in class javax.servlet.jsp.**JspWriter**
Print a boolean value and then terminate the line.

**println(char)** - Method in class javax.servlet.jsp.**JspWriter**
Print a character and then terminate the line.

**println(int)** - Method in class **javax.servlet.jsp.JspWriter**
Print an integer and then terminate the line.

**println(long)** - Method in class **javax.servlet.jsp.JspWriter**
Print a long integer and then terminate the line.

**println(float)** - Method in class **javax.servlet.jsp.JspWriter**
Print a floating-point number and then terminate the line.

**println(double)** - Method in class **javax.servlet.jsp.JspWriter**
Print a double-precision floating-point number and then terminate the line.

**println(char[])** - Method in class **javax.servlet.jsp.JspWriter**
Print an array of characters and then terminate the line.

**println(String)** - Method in class **javax.servlet.jsp.JspWriter**
Print a String and then terminate the line.

**println(Object)** - Method in class **javax.servlet.jsp.JspWriter**
Print an Object and then terminate the line.

**println()** - Method in class **javax.servlet.ServletOutputStream**
Writes a carriage return-line feed (CRLF) to the client.

**println(String)** - Method in class **javax.servlet.ServletOutputStream**
Writes a String to the client, followed by a carriage return-line feed (CRLF).

**println(boolean)** - Method in class **javax.servlet.ServletOutputStream**
Writes a boolean value to the client, followed by a carriage return-line feed (CRLF).

**println(char)** - Method in class **javax.servlet.ServletOutputStream**
Writes a character to the client, followed by a carriage return-line feed (CRLF).

**println(int)** - Method in class **javax.servlet.ServletOutputStream**
Writes an int to the client, followed by a carriage return-line feed (CRLF).

**println(long)** - Method in class **javax.servlet.ServletOutputStream**
Writes a long value to the client, followed by a carriage return-line feed (CRLF).

**println(float)** - Method in class **javax.servlet.ServletOutputStream**
Writes a float value to the client, followed by a carriage return-line feed (CRLF).

**println(double)** - Method in class **javax.servlet.ServletOutputStream**
Writes a double value to the client, followed by a carriage return-line feed (CRLF).

**printLong(long)** - Static method in class **javax.xml.bind.DatatypeConverter**
Converts a long value into a string.

**printLong(long)** - Method in interface **javax.xml.bind.DatatypeConverterInterface**
Converts a long value into a string.

**printQName(QName, NamespaceContext)** - Static method in class **javax.xml.bind.DatatypeConverter**
Converts a QName instance into a string.

**printQName(QName, NamespaceContext)** - Method in interface
javax.xml.bind.**DatatypeConverterInterface**

Converts a QName instance into a string.

**printShort(short)** - Static method in class

javax.xml.bind.**DatatypeConverter**

Converts a short value into a string.

**printShort(short)** - Method in interface

javax.xml.bind.**DatatypeConverterInterface**

Converts a short value into a string.

**printStackTrace(PrintStream)** - Method in exception

javax.ejb.**EJBException**

Prints the composite message and the embedded stack trace to the specified stream ps.

**printStackTrace()** - Method in exception javax.ejb.**EJBException**

Prints the composite message to System.err.

**printStackTrace(PrintWriter)** - Method in exception

javax.ejb.**EJBException**

Prints the composite message and the embedded stack trace to the specified print writer pw.

**printStackTrace(PrintStream)** - Method in exception

javax.xml.bind.**JAXBException**

Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to the PrintStream.

**printStackTrace()** - Method in exception

javax.xml.bind.**JAXBException**

Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to System.err.

**printStackTrace(PrintWriter)** - Method in exception

javax.xml.bind.**JAXBException**

Prints this JAXBException and its stack trace (including the stack trace of the linkedException if it is non-null) to the PrintWriter.

**printStackTrace(PrintStream)** - Method in exception

javax.xml.bind.**TypeConstraintException**

Prints this TypeConstraintException and its stack trace (including the stack trace of the linkedException if it is non-null) to the PrintStream.

**printStackTrace()** - Method in exception

javax.xml.bind.**TypeConstraintException**

Prints this TypeConstraintException and its stack trace (including the stack trace of the linkedException if it is non-null) to System.err.

**printString(String)** - Static method in class

javax.xml.bind.**DatatypeConverter**

Converts the string argument into a string.

**printString(String)** - Method in interface

javax.xml.bind.**DatatypeConverterInterface**

Converts the string argument into a string.

**printTime(Calendar)** - Static method in class
javax.xml.bind.**DatatypeConverter**  
Converts a Calendar value into a string.

**printTime(Calendar)** - Method in interface

javax.xml.bind.**DatatypeConverterInterface**  
Converts a Calendar value into a string.

**printUnsignedInt(long)** - Static method in class

javax.xml.bind.**DatatypeConverter**  
Converts a long value into a string.

**printUnsignedInt(long)** - Method in interface

javax.xml.bind.**DatatypeConverterInterface**  
Converts a long value into a string.

**printUnsignedShort(int)** - Static method in class

javax.xml.bind.**DatatypeConverter**  
Converts an int value into a string.

**printUnsignedShort(int)** - Method in interface

javax.xml.bind.**DatatypeConverterInterface**  
Converts an int value into a string.

**proceed()** - Method in interface **javax.interceptor.InvocationContext**  
Proceed to the next entry in the interceptor chain.

**PROCESS_VALIDATIONS** - Static variable in class

javax.faces.event.**PhaseId**  
Identifier that indicates an interest in events queued for the **Process Validations** phase of the request processing lifecycle.

**processAction(ActionEvent)** - Method in interface

javax.faces.event.**ActionListener**  
Invoked when the action described by the specified **ActionEvent** occurs.

**processAction(ActionEvent)** - Method in class

javax.faces.event.**MethodExpressionActionListener**

**processApplication(FacesContext)** - Method in class

javax.faces.component.**UIViewRoot**  
Broadcast any events that have been queued for the **Invoke Application** phase of the request processing lifecycle and to clear out any events for later phases if the event processing for this phase caused **FacesContext.renderResponse()** or **FacesContext.responseComplete()** to be called.

**processDecodes(FacesContext)** - Method in class

javax.faces.component.**UIComponent**  
Perform the component tree processing required by the **Apply Request Values** phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

**processDecodes(FacesContext)** - Method in class

javax.faces.component.**UIData**  
Override the default **UIComponentBase.processDecodes(javax.faces.context.FacesContext)**
processing to perform the following steps.

**processDecodes(FacesContext)** - Method in class javax.faces.component.UIForm

Override

**UIComponent.processDecodes(javax.faces.context.FacesContext)** to ensure that the form is decoded **before** its children.

**processDecodes(FacesContext)** - Method in class javax.faces.component.UIInput

Specialized decode behavior on top of that provided by the superclass.

**processDecodes(FacesContext)** - Method in class javax.faces.component.UIViewRoot

Override the default

**UIComponentBase.processDecodes(javax.faces.context.FacesContext)** behavior to broadcast any queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused **FacesContext.renderResponse()** or **FacesContext.responseComplete()** to be called.

**PROCESSING_INSTRUCTION** - Static variable in interface javax.xml.stream.XMLStreamConstants

Indicates an event is a processing instruction

**ProcessingInstruction** - Interface in javax.xml.stream.events

An interface that describes the data found in processing instructions

**processListener(FacesListener)** - Method in class javax.faces.event.ActionEvent

Broadcast this **FacesEvent** to the specified **FacesListener**, by whatever mechanism is appropriate.

**processListener(FacesListener)** - Method in class javax.faces.event.FacesEvent

Broadcast this **FacesEvent** to the specified **FacesListener**, by whatever mechanism is appropriate.

**processListener(FacesListener)** - Method in class javax.faces.event.ValueChangeEvent

Perform the component tree processing required by the **Restore View** phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

**processRestoreState(FacesContext, Object)** - Method in class javax.faces.component.UIComponent

Perform the component tree processing required by the **Restore View** phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

**processRestoreState(FacesContext, Object)** - Method in class javax.faces.component.UIComponentBase

**processSaveState(FacesContext)** - Method in class javax.faces.component.UIComponent

Perform the component tree processing required by the state saving portion of the **Render Response** phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as
follows.

**processSaveState(FacesContext)** - Method in class
javax.faces.component.UIComponentBase

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIComponent

Perform the component tree processing required by the Update Model Values phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIComponentBase

Override the default UIComponentBase.processUpdates(javax.faces.context.FacesContext) processing to perform the following steps.

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIData

Override the default UIComponentBase.processUpdates(javax.faces.context.FacesContext) processing to perform the following steps.

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIForm

Override UIComponent.processUpdates(javax.faces.context.FacesContext) to ensure that the children of this UIForm instance are only processed if UIForm.isSubmitted() returns true.

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIInput

In addition to the standard processUpdates behavior inherited from UIComponentBase, calls updateModel().

**processUpdates(FacesContext)** - Method in class
javax.faces.component.UIViewRoot

Override the default UIComponentBase behavior to broadcast any queued events after the default processing has been completed and to clear out any events for later phases if the event processing for this phase caused FacesContext.renderResponse() or FacesContext.responseComplete() to be called.

**processValidators(FacesContext)** - Method in class
javax.faces.component.UIComponent

Perform the component tree processing required by the Process Validations phase of the request processing lifecycle for all facets of this component, all children of this component, and this component itself, as follows.

**processValidators(FacesContext)** - Method in class
javax.faces.component.UIComponentBase

**processValidators(FacesContext)** - Method in class
javax.faces.component.UIData

Override the default UIComponentBase.processValidators(javax.faces.context.FacesContext) processing to perform the following steps.

**processValidators(FacesContext)** - Method in class
javax.faces.component.UIForm
Override
UIComponent.processValidators(javax.faces.context.FacesContext)
to ensure that the children of this UIForm instance are only_processed if UIForm.isSubmitted() returns true.

processValidators(FacesContext) - Method in class
domains.component.UIInput
In addition to the standard processValidators behavior
inherited from UIComponentBase, calls validate() if the
immediate property is false (which is the default); if the
component is invalid afterwards, calls
FacesContext.renderResponse().

processValidators(FacesContext) - Method in class
domains.component.UIViewRoot
Override the default
UIComponentBase.processValidators(javax.faces.context.FacesCon
text) behavior to broadcast any queued events after the default
processing has been completed and to clear out any events for
later phases if the event processing for this phase caused
FacesContext.renderResponse() or
FacesContext.responseComplete() to be called.

processValueChange(ValueChangeEvent) - Method in class
domains.event.MethodExpressionValueChangeListener
Invoked when the value change described by the specified
ValueChangeEvent occurs.

ProgressEvent - Class in javax.enterprise.deploy.spi.status
An event which indicates that a deployment status change has
occurred.

ProgressEvent(Object, TargetModuleID, DeploymentStatus) - Constructor for class
domains.enterprise.deploy.spi.status.ProgressEvent
Creates a new object representing a deployment progress event.

ProgressListener - Interface in javax.enterprise.deploy.spi.status
The listener interface for receiving deployment progress
events.

ProgressObject - Interface in javax.enterprise.deploy.spi.status
The ProgressObject interface tracks and reports the progress of
the deployment activities, distribute, start, stop, undeploy.

PropertyException - Exception in javax.xml.bind
This exception indicates that an error was encountered while
getting or setting a property.

PropertyException(String) - Constructor for exception
domains.xml.bind.PropertyException
Construct a PropertyException with the specified detail
message.

PropertyException(String, String) - Constructor for exception
domains.xml.bind.PropertyException
Construct a PropertyException with the specified detail message and vendor specific errorCode.

**PropertyException( Throwable )** - Constructor for exception
javax.xml.bind.PropertyException

Construct a PropertyException with a linkedException.

**PropertyException( String, Throwable )** - Constructor for exception
javax.xml.bind.PropertyException

Construct a PropertyException with the specified detail message, vendor specific errorCode, and linkedException.

**PropertyException( String, String, Throwable )** - Constructor for exception
javax.xml.bind.PropertyException

Construct a PropertyException whose message field is set based on the name of the property and value.toString().

**propertyExists( String )** - Method in interface javax.jms.Message
Indicates whether a property value exists.

**PropertyNotFoundException** - Exception in javax.el
Thrown when a property could not be found while evaluating a ValueExpression or MethodExpression.

**PropertyNotFoundException()** - Constructor for exception
javax.el.PropertyNotFoundException

Creates a PropertyNotFoundException with no detail message.

**PropertyNotFoundException( String )** - Constructor for exception
javax.el.PropertyNotFoundException

Creates a PropertyNotFoundException with the provided detail message.

**PropertyNotFoundException( Throwable )** - Constructor for exception
javax.el.PropertyNotFoundException

Creates a PropertyNotFoundException with the given root cause.

**PropertyNotFoundException( String, Throwable )** - Constructor for exception
javax.el.PropertyNotFoundException

Creates a PropertyNotFoundException with the given detail message and root cause.

**PropertyNotFoundException** - Exception in javax.faces.el

*Deprecated.* This has been replaced by PropertyNotFoundException.

**PropertyNotFoundException()** - Constructor for exception
javax.faces.el.PropertyNotFoundException

*Deprecated.* Construct a new exception with no detail message or root cause.

**PropertyNotFoundException( String )** - Constructor for exception
javax.faces.el.PropertyNotFoundException

*Deprecated.* Construct a new exception with the specified detail message and no root cause.

**PropertyNotFoundException( Throwable )** - Constructor for exception
javax.faces.el.PropertyNotFoundException

*Deprecated.* Construct a new exception with the specified root cause.
cause.

**PropertyNotFoundException(String, Throwable)** - Constructor for exception `javax.faces.el.PropertyNotFoundException`  
*Deprecated.* Construct a new exception with the specified detail message and root cause.

**PropertyNotFoundException** - Exception in `javax.el`  
Thrown when a property could not be written to while setting the value on a `ValueExpression`.

**PropertyNotFoundException()** - Constructor for exception `javax.el.PropertyNotFoundException`  
Creates a `PropertyNotFoundException` with no detail message.

**PropertyNotFoundException(String)** - Constructor for exception `javax.el.PropertyNotFoundException`  
Creates a `PropertyNotFoundException` with the provided detail message.

**PropertyNotFoundException(Throwables)** - Constructor for exception `javax.el.PropertyNotFoundException`  
Creates a `PropertyNotFoundException` with the given root cause.

**PropertyNotFoundException(String, Throwable)** - Constructor for exception `javax.el.PropertyNotFoundException`  
Creates a `PropertyNotFoundException` with the given detail message and root cause.

**PropertyResolver** - Class in `javax.faces.el`  
*Deprecated.* This has been replaced by `ELResolver`.

**PropertyResolver()** - Constructor for class `javax.faces.el.PropertyResolver`  
*Deprecated.*

**protocolConnect(String, int, String, String)** - Method in class `javax.mail.Service`  
The service implementation should override this method to perform the actual protocol-specific connection attempt.

**ProtocolException** - Exception in `javax.xml.ws`  
The `ProtocolException` class is a base class for exceptions related to a specific protocol binding.

**ProtocolException()** - Constructor for exception `javax.xml.ws.ProtocolException`  
Constructs a new protocol exception with null as its detail message.

**ProtocolException(String)** - Constructor for exception `javax.xml.ws.ProtocolException`  
Constructs a new protocol exception with the specified detail message.

**ProtocolException(String, Throwable)** - Constructor for exception `javax.xml.ws.ProtocolException`  
Constructs a new runtime exception with the specified detail message and cause.

**ProtocolException(Throwables)** - Constructor for exception `javax.xml.ws.ProtocolException`  
Constructs a new runtime exception with the specified cause and
The Provider is a class that describes a protocol implementation.

Provider(Provider.Type, String, String, String, String) - Constructor for class javax.mail.Provider
Create a new provider of the specified type for the specified protocol.

Provider< T > - Interface in javax.xml.ws
Service endpoints may implement the Provider interface as a dynamic alternative to an SEI.

Provider - Class in javax.xml.ws.spi
Service provider for ServiceDelegate and Endpoint objects.

Provider() - Constructor for class javax.xml.ws.spi.Provider
Creates a new instance of Provider

provider() - Static method in class javax.xml.ws.spi.Provider
Creates a new provider object.

Provider.Type - Class in javax.mail
This inner class defines the Provider type.

providers - Static variable in class javax.persistence.Persistence

publish(Message) - Method in interface javax.jms.TopicPublisher
Publishes a message to the topic.
publish(Message, int, int, long) - Method in interface javax.jms.TopicPublisher
Publishes a message to the topic, specifying delivery mode, priority, and time to live.
publish(Topic, Message) - Method in interface javax.jms.TopicPublisher
Publishes a message to a topic for an unidentified message producer.
publish(Topic, Message, int, int, long) - Method in interface javax.jms.TopicPublisher
Publishes a message to a topic for an unidentified message producer, specifying delivery mode, priority and time to live.
publish(String) - Method in class javax.xml.ws.Endpoint
Publishes this endpoint at the given address.
publish(String, Object) - Static method in class javax.xml.ws.Endpoint
Creates and publishes an endpoint for the specified implementor object at the given address.
publish(Object) - Method in class javax.xml.ws.Endpoint
Publishes this endpoint at the provided server context.
pushBody(Writer) - Method in class javax.servlet.jsp.JspContext
Return a new JspWriter object that sends output to the provided Writer.
pushBody() - Method in class javax.servlet.jsp.PageContext
Return a new BodyContent object, save the current "out" JspWriter, and update the value of the "out" attribute in the page scope attribute namespace of the PageContext.
**putContext(Class, Object)** - Method in class javax.el.ELContext
Associates a context object with this ELContext.

**putResource(Object, Object)** - Method in interface javax.transaction.TransactionSynchronizationRegistry
Add or replace an object in the Map of resources being managed for the transaction bound to the current thread at the time this method is called.

**putValue(String, Object)** - Method in interface javax.servlet.http.HttpSession
**Deprecated. As of Version 2.2, this method is replaced by HttpSession.setAttribute(java.lang.String, java.lang.Object)**
QNAMeHolder - Class in javax.xml.rpc.holders

QNAMeHolder() - Constructor for class javax.xml.rpc.holders.QNameHolder

QNAMeHolder(QName) - Constructor for class javax.xml.rpc.holders.QNameHolder

Query - Interface in javax.persistence
  Interface used to control query execution.

Query - Interface in javax.xml.registry
  The Query interface encapsulates a query in a declarative query language.

QUERY_STRING - Static variable in interface javax.xml.ws.handler.MessageContext
  Standard property: Query string for request.

QUERY_TYPE_EBXML_FILTER_QUERY - Static variable in interface javax.xml.registry.Query
  An OASIS ebXML Registry XML Filter Query type.

QUERY_TYPE_SQL - Static variable in interface javax.xml.registry.Query
  An SQL query type.

QUERY_TYPE_XQUERY - Static variable in interface javax.xml.registry.Query
  A W3C XQuery type.

QueryHint - Annotation Type in javax.persistence
  An implementation-specific Query hint.

QueryManager - Interface in javax.xml.registry
  This is the common base interface for all QueryManagers in the API.

queryNames(ObjectName, QueryExp) - Method in interface javax.management.j2ee.Management
  Gets the names of managed objects controlled by the MEJB.

queue() - Method in class javax.faces.event.FacesEvent
  Convenience method to queue this event for broadcast at the end of the current request processing lifecycle phase.

Queue - Interface in javax.jms
  A Queue object encapsulates a provider-specific queue name.

QueueBrowser - Interface in javax.jms
  A client uses a QueueBrowser object to look at messages on a queue without removing them.

QueueConnection - Interface in javax.jms
  A QueueConnection object is an active connection to a point-to-point JMS provider.

QueueConnectionFactory - Interface in javax.jms
  A client uses a QueueConnectionFactory object to create
QueueConnection objects with a point-to-point JMS provider.

queueEvent(FacesEvent) - Method in class javax.faces.component.UICommand
Intercept queueEvent and, for ActionEvents, mark the phaseId for the event to be PhaseId.APPLY_REQUEST_VALUES if the immediate flag is true, PhaseId.INVOKE_APPLICATION otherwise.

queueEvent(FacesEvent) - Method in class javax.faces.component.UIComponent
Queue an event for broadcast at the end of the current request processing lifecycle phase.

queueEvent(FacesEvent) - Method in class javax.faces.component.UIComponentBase

queueEvent(FacesEvent) - Method in class javax.faces.component.UIData
Override the default UICOMPONENTBase.queueEvent(javax.faces.event.FacesEvent) processing to wrap any queued events in a wrapper so that we can reset the current row index in broadcast().

queueEvent(FacesEvent) - Method in class javax.faces.componentUIViewRoot
Override the default UICOMPONENTBase.queueEvent(javax.faces.event.FacesEvent) behavior to accumulate the queued events for later broadcasting.

queueEvent(MailEvent, Vector) - Method in class javax.mail.Service
Add the event and vector of listeners to the queue to be delivered.

QueueReceiver - Interface in javax.jms
A client uses a QueueReceiver object to receive messages that have been delivered to a queue.

QueueRequestor - Class in javax.jms
The QueueRequestor helper class simplifies making service requests.

QueueRequestor(QueueSession, Queue) - Constructor for class javax.jms.QueueRequestor
Constructor for the QueueRequestor class.

QueueSender - Interface in javax.jms
A client uses a QueueSender object to send messages to a queue.

QueueSession - Interface in javax.jms
A QueueSession object provides methods for creating QueueReceiver, QueueSender, QueueBrowser, and TemporaryQueue objects.

Quota - Class in javax.mail
This class represents a set of quotas for a given quota root.

Quota(String) - Constructor for class javax.mail.Quota
Create a Quota object for the named quotaroot with no associated resources.

Quota_Resource - Class in javax.mail
An individual resource in a quota root.
**Quota.Resource(String, long, long)** - Constructor for class `javax.mail.Quota.Resource`
   Construct a Resource object with the given name, usage, and limit.

**QuotaAwareStore** - Interface in `javax.mail`
   An interface implemented by Stores that support quotas.

**quotaRoot** - Variable in class `javax.mail.Quota`
   The name of the quota root.

**quote(String, String)** - Static method in class `javax.mail.internet.MimeUtility`
   A utility method to quote a word, if the word contains any characters from the specified 'specials' list.

**QUOTEDSTRING** - Static variable in class `javax.mail.internet.HeaderTokenizer.Token`
   Token type indicating a quoted string.
**RangeStatistic** - Interface in javax.management.j2ee.statistics
   Specifies standard measurements of the lowest and highest values an attribute has held as well as its current value.

**RAR** - Static variable in class javax.enterprise.deploy.shared.ModuleType
   The module is an Connector archive.

**read()** - Method in class javax.mail.util.SharedFileInputStream
   See the general contract of the read method of InputStream.

**read(byte[], int, int)** - Method in class javax.mail.util.SharedFileInputStream
   Reads bytes from this stream into the specified byte array, starting at the given offset.

**read(InputStream)** - Method in interface javax.resource.cci.Streamable
   Read data from an InputStream and initialize fields of a Streamable object.

**READ ONLY** - Static variable in class javax.mail.Folder
   The Folder is read only.

**READ_WRITE** - Static variable in class javax.mail.Folder
   The state and contents of this folder can be modified.

**readBoolean()** - Method in interface javax.jms.BytesMessage
   Reads a boolean from the bytes message stream.

**readBoolean()** - Method in interface javax.jms.StreamMessage
   Reads a boolean from the stream message.

**readByte()** - Method in interface javax.jms.BytesMessage
   Reads a signed 8-bit value from the bytes message stream.

**readByte()** - Method in interface javax.jms.StreamMessage
   Reads a byte value from the stream message.

**readBytes(byte[])** - Method in interface javax.jms.BytesMessage
   Reads a byte array from the bytes message stream.

**readBytes(byte[], int)** - Method in interface javax.jms.BytesMessage
   Reads a portion of the bytes message stream.

**readBytes(byte[])** - Method in interface javax.jms.StreamMessage
   Reads a byte array field from the stream message into the specified byte[] object (the read buffer).

**readChar()** - Method in interface javax.jms.BytesMessage
   Reads a Unicode character value from the bytes message stream.

**readChar()** - Method in interface javax.jms.StreamMessage
   Reads a Unicode character value from the stream message.

**readDouble()** - Method in interface javax.jms.BytesMessage
   Reads a double from the bytes message stream.

**readDouble()** - Method in interface javax.jms.StreamMessage
   Reads a double from the stream message.

**readEJBHome(ObjectInputStream)** - Method in interface javax.ejb.spi.HandleDelegate
   Deserializes the EJBHome reference corresponding to a
**readEJBObject(ObjectInputStream)** - Method in interface `
javax.ejb.spi.HandleDelegate`

Deserialize the EJBObject reference corresponding to a Handle.

**readExternal(ObjectInput)** - Method in class `
javax.activation.MimeType`

The object implements the readExternal method to restore its contents by calling the methods of DataInput for primitive types and readObject for objects, strings and arrays.

**readFloat()** - Method in interface `
javax.jms.BytesMessage`

Reads a float from the bytes message stream.

**readFloat()** - Method in interface `
javax.jms.StreamMessage`

Reads a float from the stream message.

**readInt()** - Method in interface `
javax.jms.BytesMessage`

Reads a signed 32-bit integer from the bytes message stream.

**readInt()** - Method in interface `
javax.jms.StreamMessage`

Reads a 32-bit integer from the stream message.

**readLine(byte[], int, int)** - Method in class `
javax.servlet.ServletInputStream`

Reads the input stream, one line at a time.

**readLong()** - Method in interface `
javax.jms.BytesMessage`

Reads a signed 64-bit integer from the bytes message stream.

**readLong()** - Method in interface `
javax.jms.StreamMessage`

Reads a 64-bit integer from the stream message.

**readObject()** - Method in interface `
javax.jms.StreamMessage`

Reads an object from the stream message.

**ReadOnlyFolderException** - Exception in `
javax.mail`

This exception is thrown when an attempt is made to open a folder read-write access when the folder is marked read-only.

**ReadOnlyFolderException(Folder)** - Constructor for exception `
javax.mail.ReadOnlyFolderException`

Constructs a MessagingException with the specified folder.

**ReadOnlyFolderException(Folder, String)** - Constructor for exception `
javax.mail.ReadOnlyFolderException`

Constructs a MessagingException with the specified folder and the specified detail message.

**readResolve()** - Method in class `
javax.mail.internet.MimeMessage.RecipientType`

When deserializing a RecipientType, we need to make sure to return only one of the known static final instances defined in this class.

**readShort()** - Method in interface `
javax.jms.BytesMessage`

Reads a signed 16-bit number from the bytes message stream.

**readShort()** - Method in interface `
javax.jms.StreamMessage`

Reads a 16-bit integer from the stream message.

**readString()** - Method in interface `
javax.jms.StreamMessage`

Reads a String from the stream message.

**readUnsignedByte()** - Method in interface `
javax.jms.BytesMessage`


Reads an unsigned 8-bit number from the bytes message stream.

**readUnsignedShort()** - Method in interface javax.jms.BytesMessage
Reads an unsigned 16-bit number from the bytes message stream.

**readUTF()** - Method in interface javax.jms.BytesMessage
Reads a string that has been encoded using a modified UTF-8 format from the bytes message stream.

**receive()** - Method in interface javax.jms.MessageConsumer
Receives the next message produced for this message consumer.

**receive(long)** - Method in interface javax.jms.MessageConsumer
Receives the next message that arrives within the specified timeout interval.

**ReceivedDateTerm** - Class in javax.mail.search
This class implements comparisons for the Message Received date.

**ReceivedDateTerm(int, Date)** - Constructor for class javax.mail.search.ReceivedDateTerm
Constructor.

**receiveNoWait()** - Method in interface javax.jms.MessageConsumer
Receives the next message if one is immediately available.

**RECENT** - Static variable in class javax.mail.Flags.Flag
This message is recent.

**RecipientStringTerm** - Class in javax.mail.search
This class implements string comparisons for the Recipient Address headers.

**RecipientStringTerm(Message.RecipientType, String)** - Constructor for class javax.mail.search.RecipientStringTerm
Constructor.

**RecipientTerm** - Class in javax.mail.search
This class implements comparisons for the Recipient Address headers.

**RecipientTerm(Message.RecipientType, Address)** - Constructor for class javax.mail.search.RecipientTerm
Constructor.

**Record** - Interface in javax.resource.cci
The javax.resource.cci.Record interface is the base interface for the representation of an input or output to the execute methods defined on an Interaction.

**RecordFactory** - Interface in javax.resource.cci
The RecordFactory interface is used for creating MappedRecord and IndexedRecord instances.

**recover()** - Method in interface javax.jms.Session
Stops message delivery in this session, and restarts message delivery with the oldest unacknowledged message.

**recover(int)** - Method in interface javax.resource.spi.XATerminator
Obtains a list of prepared transaction branches from a resource manager.

**recover(int)** - Method in interface javax.transaction.xa.XAResource
Obtains a list of prepared transaction branches from a resource manager.

**recycleNode()** - Method in interface javax.xml.soap.Node
Notifies the implementation that this Node object is no longer
being used by the application and that the implementation is free to reuse this object for nodes that may be created later.

**REDEPLOY** - Static variable in class
javax.enterprise.deploy.shared.CommandType

The DeploymentManger action operation being processed is redeploy.

**redeploy(TargetModuleID[], File, File)** - Method in interface
javax.enterprise.deploy.spi.DeploymentManager

(optimal) The redeploy method provides a means for updating currently deployed J2EE applications.

**redeploy(TargetModuleID[], InputStream, InputStream)** - Method in interface
javax.enterprise.deploy.spi.DeploymentManager

(optimal) The redeploy method provides a means for updating currently deployed J2EE applications.

**redirect(String)** - Method in class
javax.faces.context.ExternalContext

Redirect a request to the specified URL, and cause the responseComplete() method to be called on the FacesContext instance for the current request.

**ref** - Variable in exception
javax.mail.internet.AddressException

The string being parsed.

**Referenceable** - Interface in
javax.resource

The Referenceable interface extends the javax.naming.Referenceable interface.

**ReferenceSyntaxException** - Exception in
javax.faces.el

Deprecated. This has been replaced by ELException.

**ReferenceSyntaxException()** - Constructor for exception
javax.faces.el.ReferenceSyntaxException

Deprecated. Construct a new exception with no detail message or root cause.

**ReferenceSyntaxException(String)** - Constructor for exception
javax.faces.el.ReferenceSyntaxException

Deprecated. Construct a new exception with the specified detail message and no root cause.

**ReferenceSyntaxException(Throw abl e)** - Constructor for exception
javax.faces.el.ReferenceSyntaxException

Deprecated. Construct a new exception with the specified root cause.

**ReferenceSyntaxException(String, Throwable)** - Constructor for exception
javax.faces.el.ReferenceSyntaxException

Deprecated. Construct a new exception with the specified detail message and root cause.

**refresh(Object)** - Method in interface
javax.persistence.EntityManager

Refresh the state of the instance from the database, overwriting changes made to the entity, if any.

**register(Class, QName, SerializerFactory, DeserializerFactory)** - Method in interface
javax.xml.rpc.encoding.TypeMapping

Registers SerializerFactory and DeserializerFactory for a specific type mapping between an XML type and Java type.
**register(String, TypeMapping)** - Method in interface `javax.xml.rpc.encoding.TypeMappingRegistry`
Registers a TypeMapping instance with the TypeMappingRegistry.

**registerDefault(TypeMapping)** - Method in interface `javax.xml.rpc.encoding.TypeMappingRegistry`
Registers the TypeMapping instance that is default for all encoding styles supported by the TypeMappingRegistry.

**registerDeploymentFactory(DeploymentFactory)** - Method in class `javax.enterprise.deploy.shared.factories.DeploymentFactoryManager`
Registers a DeploymentFactory so it will be able to handle requests.

**registerHandler(String, PolicyContextHandler, boolean)** - Static method in class `javax.security.jacc.PolicyContext`
Authorization protected method used to register a container specific PolicyContext handler.

**registerInterposedSynchronization(Synchronization)** - Method in interface `javax.transaction.TransactionSynchronizationRegistry`
Register a Synchronization instance with special ordering semantics.

**registerSynchronization(Synchronization)** - Method in interface `javax.transaction.Transaction`
Register a synchronization object for the transaction currently associated with the target object.

**REGISTRY_ENTRY** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**REGISTRY_PACKAGE** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**RegistryEntry** - Interface in `javax.xml.registry.infomodel`
The RegistryEntry interface is a base interface for interfaces in the model that require additional metadata beyond what is provided by the RegistryObject interface.

**RegistryException** - Exception in `javax.xml.registry`
This is the common base class for all Exceptions that are detected on the registry provider side rather than the JAXR client side.

**RegistryException()** - Constructor for exception `javax.xml.registry.RegistryException`
Constructs a JAXRException object with no reason or embedded Throwable.

**RegistryException(String)** - Constructor for exception `javax.xml.registry.RegistryException`
Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**RegistryException(String, Throwable)** - Constructor for exception `javax.xml.registry.RegistryException`
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.
RegistryException(Throwables) - Constructor for exception

java.xml.registry.RegistryException

Constructs a JAXRException object initialized with the given Throwable object.

RegistryObject - Interface in java.xml.registry.infomodel

The RegistryObject class is an abstract base class used by most classes in the model.

RegistryPackage - Interface in java.xml.registry.infomodel

RegistryPackage instances are RegistryEntries that group logically related RegistryEntries together.

RegistryService - Interface in java.xml.registry

This is the principal interface implemented by a JAXR provider.

release() - Method in interface

javax.enterprise.deploy.spi.DeploymentManager

The release method is the mechanism by which the tool signals to the DeploymentManager that the tool does not need it to continue running connected to the platform.

release() - Method in class javax.faces.context.FacesContext

Release any resources associated with this FacesContext instance.

release() - Method in class javax.faces.webapp.AttributeTag

Deprecated. Release references to any acquired resources.

release() - Method in class javax.faces.webapp.ConverterTag

Deprecated. Release references to any acquired resources.

release() - Method in class javax.faces.webapp.FacetTag

Release any resources allocated by this tag instance.

release() - Method in class

javax.faces.webapp.UIComponentClassicTagBase

Release any resources allocated during the execution of this tag handler.

release() - Method in class javax.faces.webapp.UIComponentELTag

Release any resources allocated during the execution of this tag handler.

release() - Method in class javax.faces.webapp.UIComponentTag

Deprecated. Release any resources allocated during the execution of this tag handler.

release() - Method in class javax.faces.webapp.ValidatorTag

Deprecated. Release references to any acquired resources.

release() - Method in interface

javax.resource.spi.endpoint.MessageEndpoint

This method may be called by the resource adapter to indicate that it no longer needs a proxy endpoint instance.

release() - Method in interface javax.resource.spi.work.Work

The WorkManager might call this method to hint the active Work instance to complete execution as soon as possible.

release() - Method in class javax.servlet.jsp.PageContext

This method shall "reset" the internal state of a PageContext, releasing all internal references, and preparing the PageContext for potential reuse by a later invocation of initialize().
**release()** - Method in class javax.servlet.jsp.tagext.BodyTagSupport
Release state.

**release()** - Method in interface javax.servlet.jsp.tagext.Tag
Called on a Tag handler to release state.

**release()** - Method in class javax.servlet.jsp.tagext.TagAdapter
Must not be called.

**release()** - Method in class javax.servlet.jsp.tagext.TagLibraryValidator
Release any data kept by this instance for validation purposes.

**release()** - Method in class javax.servlet.jsp.tagext.TagSupport
Release state.

**RELEASED** - Static variable in class
javax.enterprise.deploy.shared.StateType
The DeploymentManager is running in disconnected mode.

**releaseFactories()** - Static method in class
javax.faces.FactoryFinder
Release any references to factory instances associated with the class loader for the calling web application.

**releasePageContext(PageContext)** - Method in class
javax.servlet.jsp.JspFactory
called to release a previously allocated PageContext object.

**Remote** - Annotation Type in javax.ejb
Declares the remote business interface(s) for a session bean.

**RemoteHome** - Annotation Type in javax.ejb
Declares the Remote Home or adapted Remote Home interface for a session bean.

**remove(String)** - Method in class
javax.activation.MimeTypeParameterList
Remove any value associated with the given name.

**remove(Handle)** - Method in interface javax.ejb.EJBHome
Remove an EJB object identified by its handle.

**remove(Object)** - Method in interface javax.ejb.EJBHome
Remove an EJB object identified by its primary key.

**remove(Object)** - Method in interface javax.ejb.EJBLocalHome
Remove an EJB object identified by its primary key.

**remove()** - Method in interface javax.ejb.EJBLocalObject
Remove the EJB local object.

**remove()** - Method in interface javax.ejb.EJBObject
Remove the EJB object.

**Remove** - Annotation Type in javax.ejb
Applied to a business method of a stateful session bean class.

**remove(Flags.Flag)** - Method in class javax.mail.Flags
Remove the specified system flag from this Flags object.

**remove(String)** - Method in class javax.mail.Flags
Remove the specified user flag from this Flags object.

**remove(Flags)** - Method in class javax.mail.Flags
Remove all flags in the given Flags object from this Flags object.

**remove(String)** - Method in class javax.mail.internet.ParameterList
Removes the specified parameter from this ParameterList.
**remove(Object)** - Method in interface `javax.persistence.EntityManager`
Remove the entity instance.

**remove()** - Method in class `javax.xml.stream.util.EventReaderDelegate`

Remove an existing `ActionListener` (if any) from the set of listeners interested in being notified when `ActionEvents` occur.

**removeActionListener(ActionListener)** - Method in class `javax.faces.component.UICommand`

**removeAllAttachments()** - Method in class `javax.xml.soap.SOAPMessage`
Removes all AttachmentPart objects that have been added to this SOAPMessage object.

**removeAllFaultSubcodes()** - Method in interface `javax.xml.soap.SOAPFault`
Removes any Subcodes that may be contained by this SOAPFault.

**removeAllHeaders()** - Method in class `javax.xml.soap.MimeHeaders`
Removes all the header entries from this MimeHeaders object.

**removeAllMimeHeaders()** - Method in class `javax.xml.soap.SAPPart`
Removes all the MimeHeader objects for this SOAPEnvelope object.

**removeAllParameters()** - Method in interface `javax.xml.rpc.Call`
Removes all specified parameters from this Call instance.

**removeAssociation(Association)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`
Removes specified Association from this object.

**removeAssociations(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`
Removes specified Associations from this object.

**removeAttachments(MimeHeaders)** - Method in class `javax.xml.soap.SOAPMessage`
Removes all the AttachmentPart objects that have header entries that match the specified headers.

**removeAttribute(String)** - Method in interface `javax.servlet.http.HttpSession`
Removes the object bound with the specified name from this session.

**removeAttribute(String)** - Method in class `javax.servlet.jsp.JspContext`
Remove the object reference associated with the given name from all scopes.

**removeAttribute(String, int)** - Method in class `javax.servlet.jsp.JspContext`
Remove the object reference associated with the specified name
in the given scope.

**removeAttribute(String)** - Method in interface javax.servlet.ServletContext

Removes the attribute with the given name from the servlet context.

**removeAttribute(String)** - Method in interface javax.servlet.ServletRequest

Removes an attribute from this request.

**removeAttribute(String)** - Method in class javax.servlet.ServletRequestWrapper

The default behavior of this method is to call removeAttribute(String name) on the wrapped request object.

**removeAttribute(Name)** - Method in interface javax.xml.soap.SOAPElement

Removes the attribute with the specified name.

**removeAttribute(QName)** - Method in interface javax.xml.soap.SOAPElement

Removes the attribute with the specified qname.

**removeBodyPart(BodyPart)** - Method in class javax.mail.internet.MimeMultipart

Remove the specified part from the multipart message.

**removeBodyPart(int)** - Method in class javax.mail.internet.MimeMultipart

Remove the part at specified location (starting from 0).

**removeBodyPart(BodyPart)** - Method in class javax.mail.Multipart

Remove the specified part from the multipart message.

**removeBodyPart(int)** - Method in class javax.mail.Multipart

Remove the part at specified location (starting from 0).

**removeClassification(Classification)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Removes specified Classification from this object.

**removeClassifications(Collection)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Removes a Collection of children Concepts.

**removeChildConcept(Concept)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Removes a child Concept.

**removeChildConcept(Concept)** - Method in interface javax.xml.registry.infomodel.Concept

Removes a child Concept.

**removeChildConcepts(Collection)** - Method in interface javax.xml.registry.infomodel.ClassificationScheme

Removes a Collection of children Concepts.

**removeChildConcepts(Collection)** - Method in interface javax.xml.registry.infomodel.Concept

Removes a Collection of children Concepts.

**removeChildOrganization(Organization)** - Method in interface javax.xml.registry.infomodel.Organization

Removes a child Organization.

**removeChildOrganizations(Collection)** - Method in interface javax.xml.registry.infomodel.Organization

Removes a Collection of children Organizations.

**removeClassification(Classification)** - Method in interface javax.xml.registry.infomodel.RegistryObject

Removes specified Classification from this object.

**removeClassifications(Collection)** - Method in interface
javax.xml.registry.infomodel.\texttt{RegistryObject}
Removes specified Classifications from this object.

\texttt{removeConnectionEventListener(ConnectionEventListener)} - Method in interface \texttt{javax.resource.spi.ManagedConnection}
Removes an already registered connection event listener from the ManagedConnection instance.

\texttt{removeConnectionListener(ConnectionListener)} - Method in class \texttt{javax.mail.Folder}
Remove a Connection event listener.

\texttt{removeConnectionListener(ConnectionListener)} - Method in class \texttt{javax.mail.Service}
Remove a Connection event listener.

\texttt{removeContents()} - Method in interface \texttt{javax.xml.soap.SOAPElement}
Detaches all children of this SOAPElement.

\texttt{REMOVED} - Static variable in class \texttt{javax.mail.event.MessageCountEvent}
The messages were removed from their folder

\texttt{removed} - Variable in class \texttt{javax.mail.event.MessageCountEvent}
If true, this event is the result of an explicit expunge by this client, and the messages in this folder have been renumbered to account for this.

\texttt{removeDataModelListener(DataModelListener)} - Method in class \texttt{javax.faces.model.DataModel}
Remove an existing \texttt{DataModelListener} from the set interested in notifications from this \texttt{DataModel}.

\texttt{removeDConfigBean(DConfigBean)} - Method in interface \texttt{javax.enterprise.deploy.spi.DConfigBean}
Remove a child DConfigBean from this bean.

\texttt{removeDConfigBean(DConfigBeanRoot)} - Method in interface \texttt{javax.enterprise.deploy.spi.DeploymentConfiguration}
Remove the root DConfigBean and all its children.

\texttt{removeDeserializer(Class, QName)} - Method in interface \texttt{javax.xml.rpc.encoding.TypeMapping}
Removes the DeserializerFactory registered for the specified pair of Java type and XML data type.

\texttt{removeELContextListener(ELContextListener)} - Method in class \texttt{javax.faces.application.Application}
Remove the argument listener from the list of \texttt{ELContextListeners}.

\texttt{RemoveException} - Exception in \texttt{javax.ejb}
The RemoveException exception is thrown at an attempt to remove an EJB object when the enterprise Bean or the container does not allow the EJB object to be removed.

\texttt{RemoveException()} - Constructor for exception \texttt{javax.ejb.RemoveException}
Constructs an RemoveException with no detail message.

\texttt{RemoveException(String)} - Constructor for exception \texttt{javax.ejb.RemoveException}
Constructs an RemoveException with the specified detail message.
removeExcludedPolicy() - Method in interface javax.security.jacc.PolicyConfiguration
    Used to remove any excluded policy statements from this PolicyConfiguration.

removeExternalIdentifier(ExternalIdentifier) - Method in interface javax.xml.registry.infomodel.RegistryObject
    Removes specified ExternalIdentifier as an external identifier from this object.

removeExternalIdentifiers(Collection) - Method in interface javax.xml.registry.infomodel.RegistryObject
    Removes specified ExternalIdentifiers as an external identifiers from this object.

removeExternallLink(ExternallLink) - Method in interface javax.xml.registry.infomodel.RegistryObject
    Removes specified ExternallLink from this object.

removeExtermallLinks(Collection) - Method in interface javax.xml.registry.infomodel.RegistryObject
    Removes specified ExternallLinks from this object.

removeFacesListener(FacesListener) - Method in class javax.faces.component.UIComponent
    Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

removeFacesListener(FacesListener) - Method in class javax.faces.component.UIComponentBase
    Remove the specified FacesListener from the set of listeners registered to receive event notifications from this UIComponent.

removeFolderListener(FolderListener) - Method in class javax.mail.Store
    Remove a listener for Folder events.

removeHeader(String) - Method in class javax.mail.internet.InternetHeaders
    Remove all header entries that match the given name

removeHeader(String) - Method in class javax.mail.internet.MimeBodyPart
    Remove all headers with this name.

removeHeader(String) - Method in class javax.mail.internet.MimeMessage
    Remove all headers with this name.

removeHeader(String) - Method in interface javax.mail.Part
    Remove all MimeHeader objects whose name matches the given name.

removeLocalizedString(LocalizedString) - Method in interface javax.xml.registry.infomodel.InternationalString
    Remove the specified LocalizedString from the set of localizations registered to receive event notifications from this InternationalString.
Removes a LocalizedString from this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedStrings(Collection)` - Method in class `javax.mail.Folder`
Remove a MessageChanged listener. `removeMessageChangedListener(MessageChangedListener)` - Method in class `javax.mail.Folder`
Remove a MessageCount listener. `removeMessageCountListener(MessageCountListener)` - Method in class `javax.mail.Folder`
Remove a MessageChanged listener. `removeMessageChangedListener(MessageChangedListener)` - Method in class `javax.mail.Folder`
Remove a MessageCount listener. `removeMessageCountListener(MessageCountListener)` - Method in class `javax.mail.Folder`
Remove a MessageChanged listener. `removeMessageChangedListener(MessageChangedListener)` - Method in class `javax.mail.Folder`
Remove a MessageCount listener. `removeMessageCountListener(MessageCountListener)` - Method in class `javax.mail.Folder`
Remove a MessageChanged listener. `removeMessageChangedListener(MessageChangedListener)` - Method in class `javax.mail.Folder`
Remove a MessageCount listener. `removeMessageCountListener(MessageCountListener)` - Method in class `javax.mail.Folder`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.InternationalString`
Removes specified LocalizedStrings to this object. `removeLocalizedString` - Method in interface `javax.xml.registry.infomodel.Internat
removeRole(String) - Method in interface javax.security.jacc.PolicyConfiguration
    Used to remove a role and all its permissions from this PolicyConfiguration.
removeSerializer(Class, QName) - Method in interface javax.xml.rpc.encoding.TypeMapping
    Removes the SerializerFactory registered for the specified pair of Java type and XML data type.
removeService(Service) - Method in interface javax.xml.registry.infomodel.Organization
    Removes a Service from this object.
removeServiceBinding(ServiceBinding) - Method in interface javax.xml.registry.infomodel.Service
    Removes a child ServiceBinding.
removeServiceBindings(Collection) - Method in interface javax.xml.registry.infomodel.Service
    Removes a Collection of children ServiceBindings.
removeServices(Collection) - Method in interface javax.xml.registry.infomodel.Organization
    Removes a Collection of children Services from this object.
removeSlot(String) - Method in interface javax.xml.registry.infomodel.ExtensibleObject
    Removes a Slot from this object.
removeSlots(Collection) - Method in interface javax.xml.registry.infomodel.ExtensibleObject
    Removes specified Slots from this object.
removeSpecificationLink(SpecificationLink) - Method in interface javax.xml.registry.infomodel.ServiceBinding
    Removes a child SpecificationLink.
removeSpecificationLinks(Collection) - Method in interface javax.xml.registry.infomodel.ServiceBinding
    Removes a Collection of children SpecificationLinks.
removeStoreListener(StoreListener) - Method in class javax.mail.Store
    Remove a listener for Store events.
removeTransportListener(TransportListener) - Method in class javax.mail.Transport
    Remove a listener for Transport events.
removeTypeMapping(TypeMapping) - Method in interface javax.xml.rpc.encoding.TypeMappingRegistry
    Removes a TypeMapping from the TypeMappingRegistry.
removeUncheckedPolicy() - Method in interface javax.security.jacc.PolicyConfiguration
    Used to remove any unchecked policy statements from this PolicyConfiguration.
removeUser(User) - Method in interface javax.xml.registry.infomodel.Organization
    Removes a User.
removeUsers(Collection) - Method in interface javax.xml.registry.infomodel.Organization
Removes a Collection of Users.

**removeValidator(Validator)** - Method in interface `javax.faces.component.EditableValueHolder`
Remove a `Validator` instance from the set associated with this component, if it was previously associated.

**removeValidator(Validator)** - Method in class `javax.faces.component.UIInput`
Remove a `Validator` instance from the set associated with this `UIInput`, if it was previously associated.

**removeValue(String)** - Method in interface `javax.servlet.http.HttpSession`
*Deprecated.* As of Version 2.2, this method is replaced by `HttpSession.removeAttribute(java.lang.String)`
Remove a value associated with a key.

**removeValueChangeListener(ValueChangeListener)** - Method in interface `javax.faces.component.EditableValueHolder`
Remove an existing `ValueChangeListener` (if any) from the set of listeners interested in being notified when `ValueChangeEvent`s occur.

**removeValueChangeListener(ValueChangeListener)** - Method in class `javax.faces.component.UIInput`
Remove an existing `ValueChangeListener` (if any) from the set of listeners interested in being notified when `ValueChangeEvent`s occur.

**removeXpathListener(String, XpathListener)** - Method in interface `javax.enterprise.deploy.model/DDBean`
Unregister a listener for a specific XPath.

**removeXpathListener(ModuleType, String, XpathListener)** - Method in interface `javax.enterprise.deploy.model.J2eeApplicationObject`
Unregister the listener for an XPath.

**RENAMED** - Static variable in class `javax.mail.event.FolderEvent`
The folder was renamed.

**renameTo(Folder)** - Method in class `javax.mail.Folder`
Rename this Folder.

**render(FacesContext)** - Method in class `javax.faces.lifecycle.Lifecycle`
Execute the Render Response phase of the request processing lifecycle, unless the responseComplete() method has been called on the `FacesContext` instance associated with the current request.

**RENDER_KIT_FACTORY** - Static variable in class `javax.faces.FactoryFinder`
The property name for the `RenderKitFactory` class name.

**RENDER_KIT_ID_PARAM** - Static variable in class `javax.faces.render.ResponseStateManager`
The name of the request parameter used by the default implementation of `ViewHandler.calculateRenderKitId(javax.faces.context.FacesContext)`
to derive a RenderKit ID.

**RENDER_RESPONSE** - Static variable in class javax.faces.event.PhaseId

Identifier for the *Render Response* phase of the request processing lifecycle.

**Renderer** - Class in javax.faces.render

A *Renderer* converts the internal representation of *UIComponent* instances into the output stream (or writer) associated with the response we are creating for a particular request.

**Renderer()** - Constructor for class javax.faces.render.Renderer

**RenderKit** - Class in javax.faces.render

*RenderKit* represents a collection of *Renderer* instances that, together, know how to render JavaServer Faces *UIComponent* instances for a specific client.

**RenderKit()** - Constructor for class javax.faces.render.RenderKit

**RenderKitFactory** - Class in javax.faces.render

*RenderKitFactory* is a factory object that registers and returns *RenderKit* instances.

**RenderKitFactory()** - Constructor for class javax.faces.render.RenderKitFactory

**renderResponse()** - Method in class javax.faces.context.FacesContext

Signal the JavaServer faces implementation that, as soon as the current phase of the request processing lifecycle has been completed, control should be passed to the *Render Response* phase, bypassing any phases that have not been executed yet.

**renderView(FacesContext, UIViewRoot)** - Method in class javax.faces.application.ViewHandler

Perform whatever actions are required to render the response view to the response object associated with the current *FacesContext*.

**renderView(FacesContext, UIViewRoot)** - Method in class javax.faces.application.ViewHandlerWrapper

The default behavior of this method is to call ViewHandler.renderView(javax.faces.context.FacesContext, javax.faces.component.UIViewRoot) on the wrapped ViewHandler object.

**reply(boolean)** - Method in class javax.mail.internet.MimeMessage

Get a new Message suitable for a reply to this message.

**reply(boolean)** - Method in class javax.mail.Message

Get a new Message suitable for a reply to this message.

**report(String, String, Object, Location)** - Method in interface javax.xml.stream.XMLReporter

Report the desired message in an application specific format.

**REPORTER** - Static variable in class javax.xml.stream.XMLInputFactory

The property used to set/get the implementation of the XMLReporter interface
**request**/**Message** - Method in class `javax.jms.QueueRequestor`
Sends a request and waits for a reply.

**request**/**Message** - Method in class `javax.jms.TopicRequestor`
Sends a request and waits for a reply.

**REQUEST** - Static variable in class `javax.servlet.jsp.PageContext`
Name used to store ServletRequest in PageContext name table.

**REQUEST SCOPE** - Static variable in class `javax.servlet.jsp.PageContext`
Request scope: the named reference remains available from the ServletRequest associated with the Servlet until the current request is completed.

**REQUEST_TIME_VALUE** - Static variable in class `javax.servlet.jsp.tagext.TagData`
Distinguished value for an attribute to indicate its value is a request-time expression (which is not yet available because TagData instances are used at translation-time).

**requestDestroyed**/**ServletRequestEvent** - Method in interface `javax.servlet.ServletRequestListener`
The request is about to go out of scope of the web application.

**RequestDispatcher** - Interface in `javax.servlet`
Defines an object that receives requests from the client and sends them to any resource (such as a servlet, HTML file, or JSP file) on the server.

**requestInitialized**/**ServletRequestEvent** - Method in interface `javax.servlet.ServletRequestListener`
The request is about to come into scope of the web application.

**requestPasswordAuthentication**/**InetAddress**, **int**, **String**, **String**, **String** - Method in class `javax.mail.Session`
Call back to the application to get the needed user name and password.

**RequestWrapper** - Annotation Type in `javax.xml.ws`
Used to annotate methods in the Service Endpoint Interface with the request wrapper bean to be used at runtime.

**require**/**int**, **String**, **String** - Method in class `javax.xml.stream.utilStreamReaderDelegate`
Test if the current event is of the given type and if the namespace and name match the current namespace and name of the current event.

**REQUIRED_MESSAGE_ID** - Static variable in class `javax.faces.component.UIInput`
The message identifier of the FacesMessage to be created if a required check fails.

**reset()** - Method in interface `javax.jms.BytesMessage`
Puts the message body in read-only mode and repositions the stream of bytes to the beginning.

**reset()** - Method in interface `javax.jms.StreamMessage`
Puts the message body in read-only mode and repositions the
stream to the beginning.

**reset()** - Method in class `javax.mail.util.SharedFileInputStream`
See the general contract of the reset method of InputStream.

**reset()** - Method in interface `javax.servlet.ServletResponse`
Clears any data that exists in the buffer as well as the status code and headers.

**reset()** - Method in class `javax.servlet.ServletResponseWrapper`
The default behavior of this method is to call reset() on the wrapped response object.

**reset()** - Method in class `javax.xml.bind.util.ValidationEventCollector`
Clear all collected errors and warnings.

**resetBuffer()** - Method in interface `javax.servlet.ServletResponse`
Clears the content of the underlying buffer in the response without clearing headers or status code.

**resetBuffer()** - Method in class `javax.servlet.ServletResponseWrapper`
The default behavior of this method is to call resetBuffer() on the wrapped response object.

**resetValue()** - Method in class `javax.faces.component.UIInput`
Convenience method to reset this component’s value to the un-initialized state.

**RESOLVABLE_AT_DESIGN_TIME** - Static variable in class `javax.el.ELResolver`
The attribute name of the named attribute in the FeatureDescriptor that specifies whether the variable or property can be resolved at runtime.

**resolveEntity(String, String, String, String)** - Method in interface `javax.xml.stream.XMLResolver`
Retrieves a resource.

**resolveFunction(String, String)** - Method in class `javax.el.FunctionMapper`
Resolves the specified prefix and local name into a java.lang.Method.

**resolveFunction(String, String)** - Method in interface `javax.servlet.jsp.el.FunctionMapper`
Deprecated. Resolves the specified local name and prefix into a Java.lang.Method.

**RESOLVER** - Static variable in class `javax.xml.stream.XMLInputFactory`
The property used to set/get the implementation of the XMLResolver.

**resolveVariable(String)** - Method in class `javax.el.VariableResolver`
Deprecated. Resolve the specified variable name, and return the corresponding object, if any; otherwise, return null.

**resolveVariable(FacesContext, String)** - Method in class `javax.faces.el.VariableResolver`

**resolveVariable(String)** - Method in interface `javax.servlet.jsp.el.VariableResolver`
**Deprecated.** Resolves the specified variable.

**Resource** - Annotation Type in javax.annotation

The Resource annotation marks a resource that is needed by the application.

**Resource.AuthenticationType** - Enum in javax.annotation

The two possible authentication types for a resource.

**ResourceAdapter** - Interface in javax.resource.spi

This represents a resource adapter instance and contains operations for lifecycle management and message endpoint setup.

**ResourceAdapterAssociation** - Interface in javax.resource.spi

This interface specifies the methods to associate a ResourceAdapter object with other objects that implement this interface like ManagedConnectionFactory and ActivationSpec.

**ResourceAdapterInternalException** - Exception in javax.resource.spi

A ResourceAdapterInternalException indicates any system-level error conditions related to a resource adapter.

**ResourceAdapterInternalException()** - Constructor for exception javax.resource.spi.ResourceAdapterInternalException

Constructs a new instance with null as its detail message.

**ResourceAdapterInternalException(String)** - Constructor for exception javax.resource.spi.ResourceAdapterInternalException

Constructs a new instance with the specified detail message.

**ResourceAdapterInternalException(Throwables)** - Constructor for exception javax.resource.spi.ResourceAdapterInternalException

Constructs a new throwable with the specified cause.

**ResourceAdapterInternalException(String, Throwable)** - Constructor for exception javax.resource.spi.ResourceAdapterInternalException

Constructs a new throwable with the specified detail message and cause.

**ResourceAdapterInternalException(String, String)** - Constructor for exception javax.resource.spi.ResourceAdapterInternalException

Constructs a new throwable with the specified detail message and an error code.

**ResourceAdapterMetaData** - Interface in javax.resource.cci

The interface javax.resource.cci.ResourceAdapterMetaData provides information about capabilities of a resource adapter implementation.

**ResourceAllocationException** - Exception in javax.jms

This exception is thrown when a provider is unable to allocate the resources required by a method.

**ResourceAllocationException(String, String)** - Constructor for exception javax.jms.ResourceAllocationException

Constructs a ResourceAllocationException with the specified reason and error code.

**ResourceAllocationException(String)** - Constructor for exception javax.jms.ResourceAllocationException

Constructs a ResourceAllocationException with the specified reason.

**ResourceAllocationException** - Exception in javax.resource.spi

A ResourceAllocationException can be thrown by an application
server or resource adapter to indicate any failure to allocate system resources (example: threads, physical connections).

**ResourceAllocationException()** - Constructor for exception

*javax.resource.spi.ResourceAllocationException*

Constructs a new instance with null as its detail message.

**ResourceAllocationException(String)** - Constructor for exception

*javax.resource.spi.ResourceAllocationException*

Constructs a new instance with the specified detail message.

**ResourceAllocationException(Throwables)** - Constructor for exception

*javax.resource.spi.ResourceAllocationException*

Constructs a new throwable with the specified cause.

**ResourceAllocationException(String, Throwable)** - Constructor for exception

*javax.resource.spi.ResourceAllocationException*

Constructs a new throwable with the specified detail message and cause.

**ResourceAllocationException(String, String)** - Constructor for exception

*javax.resource.spi.ResourceAllocationException*

Constructs a new throwable with the specified detail message and cause.

**ResourceBundleELResolver** - Class in *javax.el*

Defines property resolution behavior on instances of *ResourceBundle.*

**ResourceBundleELResolver()** - Constructor for class

*javax.el ResourceBundleELResolver*

**ResourceException** - Exception in *javax.resource*

This is the root interface of the exception hierarchy defined for the Connector architecture.

**ResourceException()** - Constructor for exception

*javax.resource.ResourceException*

Constructs a new instance with null as its detail message.

**ResourceException(String)** - Constructor for exception

*javax.resource.ResourceException*

Constructs a new instance with the specified detail message.

**ResourceException(Throwables)** - Constructor for exception

*javax.resource.ResourceException*

Constructs a new throwable with the specified cause.

**ResourceException(String, Throwable)** - Constructor for exception

*javax.resource.ResourceException*

Constructs a new throwable with the specified detail message and cause.

**ResourceException(String, String)** - Constructor for exception

*javax.resource.ResourceException*

Create a new throwable with the specified message and error code.

**Resources** - Annotation Type in *javax.annotation*

This class is used to allow multiple resources declarations.

**resources** - Variable in class *javax.mail.Quota*

The set of resources associated with this quota root.

**ResourceWarning** - Exception in *javax.resource.cci*
A ResourceWarning provides information on warnings related to execution of an interaction with an EIS.

**ResourceWarning()** - Constructor for exception

`javax.resource.cci.ResourceWarning`

Constructs a new instance with null as its detail message.

**ResourceWarning(String)** - Constructor for exception

`javax.resource.cci.ResourceWarning`

Constructs a new instance with the specified detail message.

**ResourceWarning( Throwable)** - Constructor for exception

`javax.resource.cci.ResourceWarning`

Constructs a new throwable with the specified cause.

**ResourceWarning(String, Throwable)** - Constructor for exception

`javax.resource.cci.ResourceWarning`

Constructs a new throwable with the specified detail message and cause.

**ResourceWarning(String, String)** - Constructor for exception

`javax.resource.cci.ResourceWarning`

Constructs a new throwable with the specified detail message and error code.

**RESPONSE** - Static variable in class `javax.servlet.jsp.PageContext`

Name used to store ServletResponse in PageContext name table.

**Response<T>** - Interface in `javax.xml.ws`

The Response interface provides methods used to obtain the payload and context of a message sent in response to an operation invocation.

**responseComplete()** - Method in class

`javax.faces.context.FacesContext`

Signal the JavaServer Faces implementation that the HTTP response for this request has already been generated (such as an HTTP redirect), and that the request processing lifecycle should be terminated as soon as the current phase is completed.

**ResponseStateManager** - Class in `javax.faces.render`

`ResponseStateManager` is the helper class to `StateManager` that knows the specific rendering technology being used to generate the response.

**ResponseStateManager()** - Constructor for class

`javax.faces.render.ResponseStateManager`

**ResponseStream** - Class in `javax.faces.context`

`ResponseStream` is an interface describing an adapter to an underlying output mechanism for binary output.

**ResponseStream()** - Constructor for class

`javax.faces.context.ResponseStream`

**ResponseWrapper** - Annotation Type in `javax.xml.ws`

Used to annotate methods in the Service Endpoint Interface with the response wrapper bean to be used at runtime.

**ResponseWriter** - Class in `javax.faces.context`

`ResponseWriter` is an abstract class describing an adapter to an underlying output mechanism for character-based output.
**ResponseWriter()** - Constructor for class `javax.faces.context.ResponseWriter`

**ResponseWriterWrapper** - Class in `javax.faces.context` provides a simple implementation of `ResponseWriter` that can be subclassed by developers wishing to provide specialized behavior to an existing `ResponseWriter` instance.

**ResponseWriterWrapper()** - Constructor for class `javax.faces.context.ResponseWriterWrapper`

**restore(InputStream)** - Method in interface `javax.enterprise.deploy.spi.DeploymentConfiguration`
Restore from disk to a full set of configuration beans previously stored.

**RESTORE_VIEW** - Static variable in class `javax.faces.event.PhaseId` Identifier that indicates an interest in events queued for the Restore View phase of the request processing lifecycle.

**restoreAttachedState(FacesContext, Object)** - Static method in class `javax.faces.component.UIComponentBase`
This method is called by `UIComponent` subclasses that need to restore the objects they saved using `UIComponentBase.saveAttachedState(javax.faces.context.FacesContext, java.lang.Object)`.

**restoreComponentState(FacesContext, UIViewRoot, String)** - Method in class `javax.faces.application.StateManager`
Deprecated. The distinction between tree structure and component state is now an implementation detail. The default implementation does nothing.

**restoreComponentState(FacesContext, UIViewRoot, String)** - Method in class `javax.faces.application.StateManagerWrapper`
The default behavior of this method is to call `StateManager.restoreComponentState(javax.faces.context.FacesContext, javax.faces.component.UIViewRoot, String)` on the wrapped `StateManager` object.

**restoreDConfigBean(InputStream, DDBeanRoot)** - Method in interface `javax.enterprise.deploy.spi.DeploymentConfiguration`
Restore from disk to instantated objects all the DConfigBeans associated with a specific deployment descriptor.

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlColumn`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlCommandButton`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlCommandLink`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlDataTable`
**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlForm`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlGraphicImage`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlInputHidden`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlInputSecret`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlInputText`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlInputTextarea`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlMessage`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlMessages`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlOutputFormat`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlOutputLabel`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlOutputLink`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlOutputText`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlPanelGrid`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlPanelGroup`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.html.HtmlSelectOneRadio`

`restoreState(FacesContext, Object)` - Method in interface `javax.faces.component.StateHolder`
   Perform any processing required to restore the state from the entries in the state Object.

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UICommand`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIComponentBase`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIData`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIGraphic`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIInput`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIMessage`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIMessages`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIOutput`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UIParameter`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UISelectItem`

`restoreState(FacesContext, Object)` - Method in class `javax.faces.component.UISelectItems`
**restoreState(FacesContext, Object)** - Method in class `javax.faces.component.UIViewRoot`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.convert.DateTimeConverter`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.convert.EnumConverter`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.convert.NumberConverter`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.event.MethodExpressionActionListener`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.event.MethodExpressionValueChangeListener`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.validator.DoubleRangeValidator`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.validator.LengthValidator`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.validator.LongRangeValidator`

**restoreState(FacesContext, Object)** - Method in class `javax.faces.validator.MethodExpressionValidator`

**restoreTreeStructure(FacesContext, String, String)** - Method in class `javax.faces.application.StateManager`

`Deprecated`. *the distinction between tree structure and component state is now an implementation detail. The default implementation returns null.*

**restoreTreeStructure(FacesContext, String, String)** - Method in class `javax.faces.application.StateManagerWrapper`

The default behavior of this method is to call `StateManager.restoreTreeStructure(javax.faces.context.FacesContext, String, String)` on the wrapped `StateManager` object.

**restoreView(FacesContext, String, String)** - Method in class `javax.faces.application.StateManager`

Restore the tree structure and the component state of the view for the specified viewId, in an implementation dependent manner, and return the restored `UIViewRoot`.

**restoreView(FacesContext, String, String)** - Method in class `javax.faces.application.StateManagerWrapper`

The default behavior of this method is to call `StateManager.restoreView(javax.faces.context.FacesContext, String, String)` on the wrapped `StateManager` object.
**restoreView(FacesContext, String)** - Method in class javax.faces.application.ViewHandler
Perform whatever actions are required to restore the view associated with the specified FacesContext and viewId.

**restoreView(FacesContext, String)** - Method in class javax.faces.application.ViewHandlerWrapper
The default behavior of this method is to call ViewHandler.restoreView(javax.faces.context.FacesContext, String) on the wrapped ViewHandler object.

**ResultDataModel** - Class in javax.faces.model
ResultDataModel is a convenience implementation of DataModel that wraps a JSTL Result object, typically representing the results of executing an SQL query via JSTL tags.

**ResultDataModel()** - Constructor for class javax.faces.model.ResultDataModel
Construct a new ResultDataModel with no specified wrapped data.

**ResultDataModel(Result)** - Constructor for class javax.faces.model.ResultDataModel
Construct a new ResultDataModel wrapping the specified Result.

**ResultSet** - Interface in javax.resource.cci
A ResultSet represents tabular data that is retrieved from an EIS instance by the execution of an Interaction..

**ResultSetDataModel** - Class in javax.faces.model
ResultSetDataModel is a convenience implementation of DataModel that wraps a ResultSet of Java objects.

**ResultSetDataModel()** - Constructor for class javax.faces.model.ResultSetDataModel
Construct a new ResultSetDataModel with no specified wrapped data.

**ResultSetDataModel(ResultSet)** - Constructor for class javax.faces.model.ResultSetDataModel
Construct a new ResultSetDataModel wrapping the specified ResultSet.

**ResultSetInfo** - Interface in javax.resource.cci
The interface javax.resource.cci.ResultSetInfo provides information on the support provided for ResultSet by a connected EIS instance.

**resume(Transaction)** - Method in interface javax.transaction.TransactionManager
Resume the transaction context association of the calling thread with the transaction represented by the supplied Transaction object.

**RFC822** - Static variable in class javax.mail.internet.HeaderTokenizer
RFC822 specials

**RolesAllowed** - Annotation Type in javax.annotation.security
Specifies the list of roles permitted to access method(s) in an application.

**rollback()** - Method in interface javax.jms.Session
Rolls back any messages done in this transaction and releases
any locks currently held.

**rollback()** - Method in interface `javax.jms.XASession`
Throws a `TransactionInProgressException`, since it should not be called for an XASession object.

**rollback()** - Method in interface `javax.persistence.EntityTransaction`
Roll back the current transaction

**rollback()** - Method in interface `javax.resource.cci.LocalTransaction`
Rollbacks the current resource manager local transaction.

**rollback()** - Method in interface `javax.resource.spi.LocalTransaction`
Rollback a local transaction

**rollback(Xid)** - Method in interface `javax.resource.spi.XATerminator`
Informs the resource manager to roll back work done on behalf of a transaction branch.

**rollback()** - Method in interface `javax.transaction.Transaction`
Rollback the transaction represented by this Transaction object.

**rollback()** - Method in interface `javax.transaction.TransactionManager`
Roll back the transaction associated with the current thread.

**rollback()** - Method in interface `javax.transaction.UserTransaction`
Roll back the transaction associated with the current thread.

**rollback(Xid)** - Method in interface `javax.transaction.xa.XAResource`
Informs the resource manager to roll back work done on behalf of a transaction branch.

**RollbackException** - Exception in `javax.persistence`
Thrown by the persistence provider when the `EntityTransaction.commit()` fails.

**RollbackException()** - Constructor for exception `javax.persistence.RollbackException`
Constructs a new RollbackException exception with null as its detail message.

**RollbackException(String)** - Constructor for exception `javax.persistence.RollbackException`
Constructs a new RollbackException exception with the specified detail message.

**RollbackException(String, Throwable)** - Constructor for exception `javax.persistence.RollbackException`
Constructs a new RollbackException exception with the specified detail message and cause.

**RollbackException(Throw)able** - Constructor for exception `javax.persistence.RollbackException`
Constructs a new RollbackException exception with the specified cause.

**RollbackException** - Exception in `javax.transaction`
RollbackException exception is thrown when the transaction has been marked for rollback only or the transaction has been rolled back instead of committed.
**RollbackException()** - Constructor for exception  
javax.transaction.RollbackException

**RollbackException(String)** - Constructor for exception  
javax.transaction.RollbackException

**rowSelected(DataModelEvent)** - Method in interface  
javax.faces.model.DataModelListener  
Notification that a particular row index, with the associated row data, has been selected for processing.

**run()** - Method in interface javax.jms.Session  
Optional operation, intended to be used only by Application Servers, not by ordinary JMS clients.

**RunAs** - Annotation Type in javax.annotation.security  
Defines the identity of the application during execution in a J2EE container.

**RUNNING** - Static variable in class  
javax.enterprise.deploy.shared.StateType  
The action operation is running normally.
**SAAJMetaFactory** - Class in `javax.xml.soap`
- The access point for the implementation classes of the factories defined in the SAAJ API.

**SAAJMetaFactory()** - Constructor for class `javax.xml.soap.SAAJMetaFactory`

**SAAJResult** - Class in `javax.xml.soap`
- Acts as a holder for the results of a JAXP transformation or a JAXB marshalling, in the form of a SAAJ tree.

**SAAJResult()** - Constructor for class `javax.xml.soap.SAAJResult`
- Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the default (SOAP 1.1) protocol.

**SAAJResult(String)** - Constructor for class `javax.xml.soap.SAAJResult`
- Creates a SAAJResult that will present results in the form of a SAAJ tree that supports the specified protocol.

**SAAJResult(SOAPMessage)** - Constructor for class `javax.xml.soap.SAAJResult`
- Creates a SAAJResult that will write the results into the SOAPPart of the supplied SOAPMessage.

**SAAJResult(SOAPElement)** - Constructor for class `javax.xml.soap.SAAJResult`
- Creates a SAAJResult that will write the results as a child node of the SOAPElement specified.

**save(OutputStream)** - Method in interface `javax.enterprise.deploy.spi.DeploymentConfiguration`
- Save to disk the current set configuration beans created for this deployable module.

**saveAssociations(Collection, boolean)** - Method in interface `javax.xml.registry.BusinessLifeCycleManager`
- Saves the specified Association instances.

**saveAttachedState(FacesContext, Object)** - Static method in class `javax.faces.component.UIComponentBase`
- This method is called by `UIComponent` subclasses that want to save one or more attached objects.

**saveChanges()** - Method in class `javax.mail.internet.MimeMessage`
- Updates the appropriate header fields of this message to be consistent with the message's contents.

**saveChanges()** - Method in class `javax.mail.Message`
- Save any changes made to this message into the message-store when the containing folder is closed, if the message is contained in a folder.

**saveChanges()** - Method in class `javax.xml.soap.SOAPMessage`
- Updates this SOAPMessage object with all the changes that have been made to it.

**saveClassificationSchemes(Collection)** - Method in interface
javax.xml.registry.**BusinessLifeCycleManager**

Saves the specified ClassificationScheme instances.

**saveConcepts(Collection)** - Method in interface
javax.xml.registry.**BusinessLifeCycleManager**

Saves the specified Concepts.

**saved** - Variable in class javax.mail.internet.**MimeMessage**

Does the saveChanges method need to be called on this message?

**saveDConfigBean(OutputStream, DConfigBeanRoot)** - Method in interface java.util.deployment.spi.**DeploymentConfiguration**

Save to disk all the configuration beans associated with a particular deployment descriptor file.

**SaveException** - Exception in java.xml.registry

A RegistryException that occurs during a save action.

**SaveException()** - Constructor for exception
javax.xml.registry.**SaveException**

Constructs a JAXRException object with no reason or embedded Throwable.

**SaveException(String)** - Constructor for exception
javax.xml.registry.**SaveException**

Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**SaveException(String, Throwable)** - Constructor for exception
javax.xml.registry.**SaveException**

Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

**SaveException(Throwable)** - Constructor for exception
javax.xml.registry.**SaveException**

Constructs a JAXRException object initialized with the given Throwable object.

**saveFile(File)** - Method in class javax.mail.internet.**MimeBodyPart**

Save the contents of this part in the specified file.

**saveFile(String)** - Method in class javax.mail.internet.**MimeBodyPart**

Save the contents of this part in the specified file.

**saveObjects(Collection)** - Method in interface
javax.xml.registry.**LifeCycleManager**

Saves one or more Objects to the registry.

**saveOrganizations(Collection)** - Method in interface
javax.xml.registry.**BusinessLifeCycleManager**

Saves the specified Organizations.

**saveRequired()** - Method in class java.xml.soap.**SOAPMessage**

Indicates whether this SOAPMessage object needs to have the method saveChanges called on it.

**saveSerializedView(FacesContext)** - Method in class
javax.faces.application.**StateManager**

**Deprecated. this has been replaced by**

StateManager.saveView(javax.faces.context.**FacesContext**). The default implementation returns null.

**saveSerializedView(FacesContext)** - Method in class
javax.faces.application.**StateManagerWrapper**
The default behavior of this method is to call `StateManager.saveSerializedView(javax.faces.context.FacesContext)` on the wrapped `StateManager` object.

Saves the specified ServiceBindings.

Saves the specified Services.

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlColumn`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlCommandButton`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlCommandLink`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlDataTable`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlForm`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlGraphicImage`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlInputHidden`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlInputSecret`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlInputText`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlInputTextarea`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlMessage`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlMessages`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlOutputFormat`

**saveState(FacesContext)** - Method in class `javax.faces.component.html.HtmlOutputLabel`
saveState(FacesContext) - Method in class javax.faces.component.html.HtmlOutputLink

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlOutputText

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlPanelGrid

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlPanelGroup

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectManyCheckbox

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectManyListbox

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectManyMenu

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectOneListbox

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectOneMenu

saveState(FacesContext) - Method in class javax.faces.component.html.HtmlSelectOneRadio

saveState(FacesContext) - Method in interface javax.faces.component.StateHolder

　Gets the state of the instance as a Serializable Object.

saveState(FacesContext) - Method in class javax.faces.component.UICommand

saveState(FacesContext) - Method in class javax.faces.component.UICOMPONENTBASE

saveState(FacesContext) - Method in class javax.faces.component.UIData

saveState(FacesContext) - Method in class javax.faces.component.UIGraphic

saveState(FacesContext) - Method in class...
javax.faces.component.UIInput

saveState(FacesContext) - Method in class
javax.faces.component.UIMessage

saveState(FacesContext) - Method in class
javax.faces.component.UIMessages

saveState(FacesContext) - Method in class
javax.faces.component.UIOutput

saveState(FacesContext) - Method in class
javax.faces.component.UIParameter

saveState(FacesContext) - Method in class
javax.faces.component.UISelectItem

saveState(FacesContext) - Method in class
javax.faces.component.UISelectItems

saveState(FacesContext) - Method in class
javax.faces.component.UIViewRoot

saveState(FacesContext) - Method in class
javax.faces.convert.DateTimeConverter

saveState(FacesContext) - Method in class
javax.faces.convert.EnumConverter

saveState(FacesContext) - Method in class
javax.faces.convert.NumberConverter

saveState(FacesContext) - Method in class
javax.faces.event.MethodExpressionActionListener

saveState(FacesContext) - Method in class
javax.faces.event.MethodExpressionValueChangeListener

saveState(FacesContext) - Method in class
javax.faces.validator.DoubleRangeValidator

saveState(FacesContext) - Method in class
javax.faces.validator.LengthValidator

saveState(FacesContext) - Method in class
javax.faces.validator.LongRangeValidator

saveState(FacesContext) - Method in class
javax.faces.validator.MethodExpressionValidator
**saveView(FacesContext)** - Method in class javax.faces.application.StateManager

Return an opaque Object containing sufficient information for this same instance to restore the state of the current UIViewRoot on a subsequent request.

**saveView(FacesContext)** - Method in class javax.faces.application.StateManagerWrapper

The default behavior of this method is to call StateManager.saveView(javax.faces.context.FacesContext) on the wrapped StateManager object.

**SC_ACCEPTED** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (202) indicating that a request was accepted for processing, but was not completed.

**SC_BAD_GATEWAY** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (502) indicating that the HTTP server received an invalid response from a server it consulted when acting as a proxy or gateway.

**SC_BAD_REQUEST** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (400) indicating the request sent by the client was syntactically incorrect.

**SC_CONFLICT** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (409) indicating that the request could not be completed due to a conflict with the current state of the resource.

**SC_CONTINUE** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (100) indicating the client can continue.

**SC_CREATED** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (201) indicating the request succeeded and created a new resource on the server.

**SC_EXPECTATION_FAILED** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (417) indicating that the server could not meet the expectation given in the Expect request header.

**SC_FORBIDDEN** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (403) indicating the server understood the request but refused to fulfill it.

**SC_FOUND** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (302) indicating that the resource reside temporarily under a different URI.

**SC_GATEWAY_TIMEOUT** - Static variable in interface javax.servlet.http.HttpServletResponse

Status code (504) indicating that the server did not receive a
timely response from the upstream server while acting as a gateway or proxy.

**SC GONE** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (410) indicating that the resource is no longer available at the server and no forwarding address is known.

**SC HTTP_VERSION_NOT_SUPPORTED** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (505) indicating that the server does not support or refuses to support the HTTP protocol version that was used in the request message.

**SC INTERNAL_SERVER_ERROR** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (500) indicating an error inside the HTTP server which prevented it from fulfilling the request.

**SC_LENGTH_REQUIRED** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (411) indicating that the request cannot be handled without a defined `Content-Length`.

**SC_METHOD_NOT_ALLOWED** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (405) indicating that the method specified in the `Request-Line` is not allowed for the resource identified by the `Request-URI`.

**SC_MOVED_PERMANENTLY** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (301) indicating that the resource has permanently moved to a new location, and that future references should use a new URI with their requests.

**SC_MOVED_TEMPORARILY** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (302) indicating that the resource has temporarily moved to another location, but that future references should still use the original URI to access the resource.

**SC_MULTIPLE_CHOICES** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (300) indicating that the requested resource corresponds to any one of a set of representations, each with its own specific location.

**SC_NO_CONTENT** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (204) indicating that the request succeeded but that there was no new information to return.

**SC_NON_AUTHORITATIVE_INFORMATION** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (203) indicating that the meta information presented by the client did not originate from the server.

**SC_NOT_ACCEPTABLE** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (406) indicating that the resource identified by
the request is only capable of generating response entities which have content characteristics not acceptable according to the accept headers sent in the request.

**SC_NOT_FOUND** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (404) indicating that the requested resource is not available.

**SC_NOT_IMPLEMENTED** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (501) indicating the HTTP server does not support the functionality needed to fulfill the request.

**SC_NOT_MODIFIED** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (304) indicating that a conditional GET operation found that the resource was available and not modified.

**SC_OK** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (200) indicating the request succeeded normally.

**SC_PARTIAL_CONTENT** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (206) indicating that the server has fulfilled the partial GET request for the resource.

**SC_PAYMENT_REQUIRED** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (402) reserved for future use.

**SC_PRECONDITION_FAILED** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (412) indicating that the precondition given in one or more of the request-header fields evaluated to false when it was tested on the server.

**SC_PROXY_AUTHENTICATION_REQUIRED** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (407) indicating that the client MUST first authenticate itself with the proxy.

**SC_REQUEST_ENTITY_TOO_LARGE** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (413) indicating that the server is refusing to process the request because the request entity is larger than the server is willing or able to process.

**SC_REQUEST_TIMEOUT** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (408) indicating that the client did not produce a request within the time that the server was prepared to wait.

**SC_REQUEST_URI_TOO_LONG** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (414) indicating that the server is refusing to service the request because the Request-URI is longer than the server is willing to interpret.

**SC_REQUESTED_RANGE_NOT_SATISFIABLE** - Static variable in interface javax.servlet.http.HttpServletResponse
Status code (416) indicating that the server cannot serve the requested byte range.

**SC_RESET_CONTENT** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (205) indicating that the agent SHOULD reset the document view which caused the request to be sent.

**SCSEE_OTHER** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (303) indicating that the response to the request can be found under a different URI.

**SC_SERVICE_UNAVAILABLE** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (503) indicating that the HTTP server is temporarily overloaded, and unable to handle the request.

**SC_SWITCHING_PROTOCOLS** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (101) indicating the server is switching protocols according to Upgrade header.

**SC_TEMPORARY_REDIRECT** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (307) indicating that the requested resource resides temporarily under a different URI.

**SC_UNAUTHORIZED** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (401) indicating that the request requires HTTP authentication.

**SC_UNSUPPORTED_MEDIA_TYPE** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (415) indicating that the server is refusing to service the request because the entity of the request is in a format not supported by the requested resource for the requested method.

**SC_USE_PROXY** - Static variable in interface `javax.servlet.http.HttpServletResponse`
Status code (305) indicating that the requested resource MUST be accessed through the proxy given by the Location field.

**ScalarDataModel** - Class in `javax.faces.model`
*ScalarDataModel* is a convenience implementation of `DataModel` that wraps an individual Java object.

**ScalarDataModel()** - Constructor for class `javax.faces.model.ScalarDataModel`
Construct a new `ScalarDataModel` with no specified wrapped data.

**ScalarDataModel(Object)** - Constructor for class `javax.faces.model.ScalarDataModel`
Construct a new `ScalarDataModel` wrapping the specified scalar object.

**scheduleWork(Work)** - Method in interface `javax.resource.spi.work.WorkManager`
Accepts a Work instance for processing.

**scheduleWork(Work, long, ExecutionContext, WorkListener)** - Method
in interface javax.resource.spi.work.WorkManager
  Accepts a Work instance for processing.

**SchemaOutputResolver** - Class in **javax.xml.bind**
  Controls where a JAXB implementation puts the generates schema files.

**SchemaOutputResolver()** - Constructor for class
javax.xml.bind.SchemaOutputResolver

**scope** - Variable in class javax.xml.bind.JAXBElement
  Scope of xml element declaration representing this xml element instance.

**ScopedAttributeELResolver** - Class in **javax.servlet.jsp.el**
  Defines variable resolution behavior for scoped attributes.

**ScopedAttributeELResolver()** - Constructor for class
javax.servlet.jsp.el.ScopedAttributeELResolver

**search(SearchTerm)** - Method in class javax.mail.Folder
  Search this Folder for messages matching the specified search criterion.

**search(SearchTerm, Message[])** - Method in class javax.mail.Folder
  Search the given array of messages for those that match the specified search criterion.

**SearchException** - Exception in **javax.mail.search**
  The exception thrown when a Search expression could not be handled.

**SearchException()** - Constructor for exception
javax.mail.search.SearchException
  Constructs a SearchException with no detail message.

**SearchException(String)** - Constructor for exception
javax.mail.search.SearchException
  Constructs a SearchException with the specified detail message.

**SearchTerm** - Class in **javax.mail.search**
  Search criteria are expressed as a tree of search-terms, forming a parse-tree for the search expression.

**SearchTerm()** - Constructor for class javax.mail.search.SearchTerm

**SecondaryTable** - Annotation Type in **javax.persistence**
  This annotation is used to specify a secondary table for the annotated entity class.

**SecondaryTables** - Annotation Type in **javax.persistence**
  This annotation is used to specify multiple secondary tables for an entity.

**SecurityException** - Exception in **javax.resource.spi**
  A SecurityException indicates error conditions related to the security contract between an application server and resource adapter.

**SecurityException()** - Constructor for exception
javax.resource.spi.SecurityException
  Constructs a new instance with null as its detail message.

**SecurityException(String)** - Constructor for exception
javax.resource.spi.SecurityException
Constructs a new instance with the specified detail message.

SecurityException(Throwable) - Constructor for exception
Constructs a new throwable with the specified cause.

SecurityException(String, Throwable) - Constructor for exception
Constructs a new throwable with the specified detail message and cause.

SecurityException(String, String) - Constructor for exception
Constructs a new throwable with the specified detail message and an error code.

SEEN - Static variable in class javax.mail.Flags.Flag
This message is seen.

SelectItem - Class in javax.faces.model
SelectItem represents a single item in the list of supported items associated with a UISelectMany or UISelectOne component.

SelectItem() - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem with no initialized property values.

SelectItem(Object) - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem with the specified value.

SelectItem(Object, String) - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem with the specified value and label.

SelectItem(Object, String, String) - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem instance with the specified value, label and description.

SelectItem(Object, String, String, boolean) - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem instance with the specified property values.

SelectItem(Object, String, boolean, boolean) - Constructor for class javax.faces.model.SelectItem
Construct a SelectItem instance with the specified property values.

SelectItemGroup - Class in javax.faces.model
SelectItemGroup is a subclass of SelectItem that identifies a set of options that will be made available as a subordinate "submenu" or "options list", depending upon the requirements of the UISelectMany or UISelectOne renderer that is actually used.

SelectItemGroup() - Constructor for class javax.faces.model.SelectItemGroup
Construct a SelectItemGroup with no initialized property values.

SelectItemGroup(String) - Constructor for class javax.faces.model.SelectItemGroup
Construct a SelectItemGroup with the specified label and no
associated selectItems.

`SelectItemGroup(String, String, boolean, SelectItem[])` - Constructor for class `javax.faces.model.SelectItemGroup`

Construct a SelectItemGroup with the specified properties.

`send(Message)` - Method in interface `javax.jms.MessageProducer`

Sends a message using the MessageProducer's default delivery mode, priority, and time to live.

`send(Message, int, int, long)` - Method in interface `javax.jms.MessageProducer`

Sends a message to the destination, specifying delivery mode, priority, and time to live.

`send(Destination, Message)` - Method in interface `javax.jms.MessageProducer`

Sends a message to a destination for an unidentified message producer.

`send(Destination, Message, int, int, long)` - Method in interface `javax.jms.MessageProducer`

Sends a message to a destination for an unidentified message producer, specifying delivery mode, priority and time to live.

`send(Message)` - Method in interface `javax.jms.QueueSender`

Sends a message to the queue.

`send(Message, int, int, long)` - Method in interface `javax.jms.QueueSender`

Sends a message to the queue, specifying delivery mode, priority, and time to live.

`send(Queue, Message)` - Method in interface `javax.jms.QueueSender`

Sends a message to a queue for an unidentified message producer.

`send(Queue, Message, int, int, long)` - Method in interface `javax.jms.QueueSender`

Sends a message to a queue for an unidentified message producer, specifying delivery mode, priority and time to live.

`send(Message)` - Static method in class `javax.mail.Transport`

Send a message.

`send(Message, Address[])` - Static method in class `javax.mail.Transport`

Send the message to the specified addresses, ignoring any recipients specified in the message itself.


Sends an error response to the client using the specified status.

`sendError(int)` - Method in interface `javax.servlet.http.HttpServletResponse`

Sends an error response to the client using the specified status code and clearing the buffer.

`sendError(int, String)` - Method in class `javax.servlet.http.HttpServletResponseWrapper`

The default behavior of this method is to call sendError(int sc, String msg) on the wrapped response object.
**sendError(int)** - Method in class javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to call sendError(int sc) on the wrapped response object.

**SendFailedException** - Exception in javax.mail
This exception is thrown when the message cannot be sent.

**SendFailedException()** - Constructor for exception
javax.mail.SendFailedException
Constructs a SendFailedException with no detail message.

**SendFailedException(String)** - Constructor for exception
javax.mail.SendFailedException
Constructs a SendFailedException with the specified detail message.

**SendFailedException(String, Exception)** - Constructor for exception
javax.mail.SendFailedException
Constructs a SendFailedException with the specified Exception and detail message.

**SendFailedException(String, Exception, Address[], Address[], Address[])** - Constructor for exception
javax.mail.SendFailedException
Constructs a SendFailedException with the specified string and the specified address objects.

**sendMessage(Message, Address[])** - Method in class
javax.mail.Transport
Send the Message to the specified list of addresses.

**sendRedirect(String)** - Method in interface
javax.servlet.http.HttpServletResponse
Sends a temporary redirect response to the client using the specified redirect location URL.

**sendRedirect(String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to return sendRedirect(String location) on the wrapped response object.

**SentDateTerm** - Class in javax.mail.search
This class implements comparisons for the Message SentDate.

**SentDateTerm(int, Date)** - Constructor for class
javax.mail.search.SentDateTerm
Constructor.

**SEPARATOR_CHAR** - Static variable in interface
javax.faces.component.NamingContainer
The separator character used in component identifiers to demarcate navigation to a child naming container.

**SequenceGenerator** - Annotation Type in javax.persistence
This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the GeneratedValue annotation.

**SerializationContext** - Interface in javax.xml.rpc.encoding
The javax.xml.rpc.encoding.SerializationContext interface is implemented by the JAX-RPC runtime system in an XML processing mechanism specific manner.
**Serializer** - Interface in `javax.xml.rpc.encoding`
The `javax.xml.rpc.encoding.Serializer` interface defines the base interface for serializers.

**SerializerFactory** - Interface in `javax.xml.rpc.encoding`
The `javax.xml.rpc.encoding.SerializerFactory` is a factory of the serializers.

**ServerSession** - Interface in `javax.jms`
A `ServerSession` object is an application server object that is used by a server to associate a thread with a JMS session (optional).

**ServerSessionPool** - Interface in `javax.jms`
A `ServerSessionPool` object is an object implemented by an application server to provide a pool of `ServerSession` objects for processing the messages of a `ConnectionConsumer` (optional).

**service(ServletRequest, ServletResponse)** - Method in class `javax.faces.webapp.FacesServlet`
Process an incoming request, and create the corresponding response, by executing the request processing lifecycle.

**Service** - Class in `javax.mail`
An abstract class that contains the functionality common to messaging services, such as stores and transports.

**Service(Session, URLName)** - Constructor for class `javax.mail.Service`
Constructor.

**service(ServletRequest, ServletResponse)** - Method in class `javax.servlet.GenericServlet`
Called by the servlet container to allow the servlet to respond to a request.

**service(HttpServletRequest, HttpServletResponse)** - Method in class `javax.servlet.http.HttpServlet`
Receives standard HTTP requests from the public service method and dispatches them to the doXXX methods defined in this class.

Dispatches client requests to the protected service method.

**Service(ServletRequest, ServletResponse)** - Method in interface `javax.servlet.Servlet`
Called by the servlet container to allow the servlet to respond to a request.

**Service** - Interface in `javax.xml.registry.infomodel`
Service instances are RegistryObjects that provide information on services (for example, web services) offered by an Organization.

**SERVICE** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**Service** - Interface in `javax.xml.rpc`
Service class acts as a factory of the following: Dynamic proxy for the target service endpoint.

**Service** - Class in `javax.xml.ws`
Service objects provide the client view of a Web service. 
**Service(URL, QName)** - Constructor for class javax.xml.ws.Service

**Service.Mode** - Enum in javax.xml.ws
The orientation of a dynamic client or service.
**SERVICE_BINDING** - Static variable in interface javax.xml.registry.LifeCycleManager
Maps to UDDI serviceSubset.

**SERVICE_SUBSET** - Static variable in interface javax.xml.registry.FindQualifier
Maps to UDDI serviceSubset.

**ServiceBinding** - Interface in javax.xml.registry.infomodel
ServiceBinding instances are RegistryObjects that represent technical information on a specific way to access a specific interface offered by a Service instance.

**ServiceDelegate** - Class in javax.xml.ws.spi
Service delegates are used internally by Service objects to allow pluggability of JAX-WS implementations.
**ServiceDelegate()** - Constructor for class javax.xml.ws.spi.ServiceDelegate

**ServiceException** - Exception in javax.xml.rpc
The javax.xml.rpc.ServiceException is thrown from the methods in the javax.xml.rpc.Service interface and ServiceFactory class.

**ServiceException()** - Constructor for exception javax.xml.rpc.ServiceException
Constructs a new exception with null as its detail message.

**ServiceException(String)** - Constructor for exception javax.xml.rpc.ServiceException
Constructs a new exception with the specified detail message.

**ServiceException(String, Throwable)** - Constructor for exception javax.xml.rpc.ServiceException
Constructs a new exception with the specified detail message and cause.

**ServiceException(Throwlable)** - Constructor for exception javax.xml.rpc.ServiceException
Constructs a new exception with the specified cause and a detail message of (cause==null ?

**ServiceFactory** - Class in javax.xml.rpc
The javax.xml.rpc.ServiceFactory is an abstract class that provides a factory for the creation of instances of the type javax.xml.rpc.Service.

**ServiceFactory()** - Constructor for class javax.xml.rpc.ServiceFactory

**SERVICEFACTORY_PROPERTY** - Static variable in class javax.xml.rpc.ServiceFactory
A constant representing the property used to lookup the name of a ServiceFactory implementation class.
**ServiceLifecycle** - Interface in [javax.xml.rpc.server](http://example.com/)
The javax.xml.rpc.server.ServiceLifecycle defines a lifecycle interface for a JAX-RPC service endpoint.

**ServiceMode** - Annotation Type in [javax.xml.ws](http://example.com/)
Used to indicate whether a Provider implementation wishes to work with entire protocol messages or just with protocol message payloads.

**Servlet** - Interface in [javax.servlet](http://example.com/)
Defines methods that all servlets must implement.

**SERVLET CONTEXT** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: servlet context object.

**SERVLET REQUEST** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: servlet request object.

**SERVLET RESPONSE** - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: servlet response object.

**ServletConfig** - Interface in [javax.servlet](http://example.com/)
A servlet configuration object used by a servlet container to pass information to a servlet during initialization.

**ServletContext** - Interface in [javax.servlet](http://example.com/)
Defines a set of methods that a servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.

**ServletContextAttributeEvent** - Class in [javax.servlet](http://example.com/)
This is the event class for notifications about changes to the attributes of the servlet context of a web application.

**ServletContextAttributeEvent(ServletContext, String, Object)** - Constructor for class javax.servlet.ServletContextAttributeEvent
Construct a ServletContextAttributeEvent from the given context for the given attribute name and attribute value.

**ServletContextAttributeListener** - Interface in [javax.servlet](http://example.com/)
Implementations of this interface receive notifications of changes to the attribute list on the servlet context of a web application.

**ServletContextEvent** - Class in [javax.servlet](http://example.com/)
This is the event class for notifications about changes to the servlet context of a web application.

**ServletContextEvent(ServletContext)** - Constructor for class javax.servlet.ServletContextEvent
Construct a ServletContextEvent from the given context.

**ServletContextListener** - Interface in [javax.servlet](http://example.com/)
Implementations of this interface receive notifications about changes to the servlet context of the web application they are part of.

**ServletEndpointContext** - Interface in [javax.xml.rpc.server](http://example.com/)
The ServletEndpointContext provides an endpoint context maintained by the underlying servlet container based JAX-RPC runtime system.
**ServletException** - Exception in `javax.servlet`
Defines a general exception a servlet can throw when it encounters difficulty.

**ServletException()** - Constructor for exception
java.servlet. ServletException
Constructs a new servlet exception.

**ServletException(String)** - Constructor for exception
java.servlet. ServletException
Constructs a new servlet exception with the specified message.

**ServletException(String, Throwable)** - Constructor for exception
java.servlet. ServletException
Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation, including a description message.

**ServletException( Throwable)** - Constructor for exception
java.servlet. ServletException
Constructs a new servlet exception when the servlet needs to throw an exception and include a message about the "root cause" exception that interfered with its normal operation.

**ServletInputStream** - Class in `javax.servlet`
Provides an input stream for reading binary data from a client request, including an efficient `readLine` method for reading data one line at a time.

**ServletInputStream()** - Constructor for class
java.servlet. ServletInputStream
Does nothing, because this is an abstract class.

**ServletOutputStream** - Class in `javax.servlet`
Provides an output stream for sending binary data to the client.

**ServletOutputStream()** - Constructor for class
java.servlet. ServletOutputStream
Does nothing, because this is an abstract class.

**ServletRequest** - Interface in `javax.servlet`
Defines an object to provide client request information to a servlet.

**ServletRequestAttributeEvent** - Class in `javax.servlet`
This is the event class for notifications of changes to the attributes of the servlet request in an application.

**ServletRequestAttributeEvent(ServletContext, ServletRequest, String, Object)** - Constructor for class
java.servlet. ServletRequestAttributeEvent
Construct a ServletRequestAttributeEvent giving the servlet context of this web application, the ServletRequest whose attributes are changing and the name and value of the attribute.

**ServletRequestAttributeListener** - Interface in `javax.servlet`
A ServletRequestAttributeListener can be implemented by the developer interested in being notified of request attribute changes.
ServletRequestEvent - Class in javax.servlet
Events of this kind indicate lifecycle events for a ServletRequest.

ServletRequestEvent(ServletContext, ServletRequest) - Constructor for class javax.servlet.ServletRequestEvent
Construct a ServletRequestEvent for the given ServletContext and ServletRequest.

ServletRequestListener - Interface in javax.servlet
A ServletRequestListener can be implemented by the developer interested in being notified of requests coming in and out of scope in a web component.

ServletRequestWrapper - Class in javax.servlet
Provides a convenient implementation of the ServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet.

ServletRequestWrapper(ServletRequest) - Constructor for class javax.servlet.ServletRequestWrapper
Creates a ServletRequest adaptor wrapping the given request object.

ServletResponse - Interface in javax.servlet
Defines an object to assist a servlet in sending a response to the client.

ServletResponseWrapper - Class in javax.servlet
Provides a convenient implementation of the ServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet.

ServletResponseWrapper(ServletResponse) - Constructor for class javax.servletServletResponseWrapper
Creates a ServletResponse adaptor wrapping the given response object.

ServletStats - Interface in javax.management.j2ee.statistics
Specifies the statistics provided by a Servlet.

Session - Interface in javax.jms
A Session object is a single-threaded context for producing and consuming messages.

session - Variable in class javax.mail.Message
The Session object for this Message

session - Variable in class javax.mail.Service
The session from which this service was created.

Session - Class in javax.mail
The Session class represents a mail session and is not subclassed.

SESSION - Static variable in class javax.servlet.jsp.PageContext
Name used to store HttpSession in PageContext name table.

SESSION_MAINTAIN_PROPERTY - Static variable in interface javax.xml.rpc.Call
Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint.

SESSION_MAINTAIN_PROPERTY - Static variable in interface
javax.xml.rpc.Stub
Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint.

SESSION_MAINTAIN_PROPERTY - Static variable in interface javax.xml.ws.BindingProvider
Standard property: This boolean property is used by a service client to indicate whether or not it wants to participate in a session with a service endpoint.

SESSION_SCOPE - Static variable in class javax.servlet.jsp.PageContext
Session scope (only valid if this page participates in a session): the named reference remains available from the HttpSession (if any) associated with the Servlet until the HttpSession is invalidated.

SESSION_TRANSACTED - Static variable in interface javax.jms.Session
This value is returned from the method getAcknowledgeMode if the session is transacted.

SessionBean - Interface in javax.ejb
The SessionBean interface is implemented by every session enterprise Bean class.

SessionBeanStats - Interface in javax.management.j2ee.statistics
Specifies the statistics provided by session beans of both stateful and stateless types.

SessionContext - Interface in javax.ejb
The SessionContext interface provides access to the runtime session context that the container provides for a session enterprise Bean instance.

sessionCreated(HttpSessionEvent) - Method in interface javax.servlet.http.HttpSessionListener
Notification that a session was created.

sessionDestroyed(HttpSessionEvent) - Method in interface javax.servlet.http.HttpSessionListener
Notification that a session is about to be invalidated.

sessionDidActivate(HttpSessionEvent) - Method in interface javax.servlet.http.HttpSessionActivationListener
Notification that the session has just been activated.

SessionSynchronization - Interface in javax.ejb
The SessionSynchronization interface allows a session Bean instance to be notified by its container of transaction boundaries.

sessionWillPassivate(HttpSessionEvent) - Method in interface javax.servlet.http.HttpSessionActivationListener
Notification that the session is about to be passivated.

set(String, String) - Method in class javax.activation.MimeTypeParameterList
Set the value to be associated with the given name, replacing any previous association.

set(String, String) - Method in class javax.mail.internet.ParameterList
Set a parameter.

**set(String, String, String)** - Method in class
javax.mail.internet.*ParameterList*

Set a parameter.

**set** - Variable in class javax.mail.search.*FlagTerm*
Indicates whether to test for the presence or absence of the specified Flag.

**setAccept(String)** - Method in class
javax.faces.component.html.*HtmlForm*
Set the value of the accept property.

**setAcceptcharset(String)** - Method in class
javax.faces.component.html.*HtmlForm*
Set the value of the acceptcharset property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlCommandButton*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlCommandLink*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlInputSecret*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlInputText*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlInputTextarea*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlOutputLabel*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlOutputLink*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlSelectBooleanCheckbox*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlSelectManyCheckbox*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlSelectManyListbox*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlSelectManyMenu*
Set the value of the accesskey property.

**setAccesskey(String)** - Method in class
javax.faces.component.html.*HtmlSelectOneListbox*
Set the value of the accesskey property.
javax.faces.component.html.HtmlSelectOneMenu

- Set the value of the accesskey property.

`setAccesskey(String)` - Method in class

javax.faces.component.html.HtmlSelectOneRadio

- Set the value of the accesskey property.

`setAccessURI(String)` - Method in interface

javax.xml.registry.infomodel.ServiceBinding

- Sets the URI that gives access to the service via this binding.

`setaction(MethodBinding)` - Method in interface

javax.faces.component.ActionSource

- Deprecated. This has been replaced by


`setaction(MethodBinding)` - Method in class

javax.faces.component.UICommand

- Deprecated. This has been replaced by


`setactionExpression(MethodExpression)` - Method in interface

javax.faces.component.ActionSource2

- Set the `MethodExpression` pointing at the application action to be invoked, if this `UIComponent` is activated by the user, during the Apply Request Values or Invoke Application phase of the request processing lifecycle, depending on the value of the immediate property.

`setactionExpression(MethodExpression)` - Method in class

javax.faces.component.UICommand

`setActionListener(ActionListener)` - Method in class

javax.faces.application.Application

- Set the default `ActionListener` to be registered for all `ActionSource` components.

`setactionListener(MethodBinding)` - Method in interface

javax.faces.component.ActionSource

- Deprecated. This has been replaced by


`setactionListener(MethodBinding)` - Method in class

javax.faces.component.UICommand

- Deprecated. This has been replaced by


`setActor(String)` - Method in interface

javax.xml.soap.SOAPHeaderElement

- Sets the actor associated with this SOAPHeaderElement object to the specified actor.

`setaapter(XmlAdapter)` - Method in class

javax.xml.bind.helpers.AbstractMarshallerImpl

`setAdapter(Class<A>, A)` - Method in class

javax.xml.bind.helpers.AbstractMarshallerImpl

`setaapter(XmlAdapter)` - Method in class

javax.xml.bind.helpers.AbstractUnmarshallerImpl
**setAdapter(Class<A>, A)** - Method in class
javax.xml.bind.helpers.*AbstractUnmarshallerImpl*

**setAdapter(XmlAdapter)** - Method in interface
javax.xml.bind.*Marshaller*
Associates a configured instance of *XmlAdapter* with this marshaller.

**setAdapter(Class<A>, A)** - Method in interface
javax.xml.bind.*Marshaller*
Associates a configured instance of *XmlAdapter* with this marshaller.

**setAdapter(XmlAdapter)** - Method in interface
javax.xml.bind.*Unmarshaller*
Associates a configured instance of *XmlAdapter* with this unmarshaller.

**setAdapter(Class<A>, A)** - Method in interface
javax.xml.bind.*Unmarshaller*
Associates a configured instance of *XmlAdapter* with this unmarshaller.

**setAddress(String)** - Method in class
javax.mail.internet.*InternetAddress*
Set the email address.

**setAddress(String)** - Method in interface
javax.xml.registry.infomodel.*EmailAddress*
Sets the email address for this object.

**setAfterPhaseListener(MethodExpression)** - Method in class
javax.faces.component.*UIViewRoot*
Allow an arbitrary method to be called for the "afterPhase" event as the UIViewRoot runs through its lifecycle.

**setAlt(String)** - Method in class
javax.faces.component.html.*HtmlCommandButton*
Set the value of the alt property.

**setAlt(String)** - Method in class
javax.faces.component.html.*HtmlGraphicImage*
Set the value of the alt property.

**setAlt(String)** - Method in class
javax.faces.component.html.*HtmlInputSecret*
Set the value of the alt property.

**setAlt(String)** - Method in class
javax.faces.component.html.*HtmlInputText*
Set the value of the alt property.

**setApplication(Application)** - Method in class
javax.faces.application.*ApplicationFactory*
Replace the *Application* instance that will be returned for this web application.

**setAreaCode(String)** - Method in interface
javax.xml.registry.infomodel.*TelephoneNumber*
Sets the area code.

**setAssociations(Collection)** - Method in interface
Replaces all previous Associations from this object with specified Associations.

**setAssociationType(Concept)** - Method in interface `javax.xml.registry.infomodel.Association`
Sets the association type for this Association.

**setAttachmentMarshaller(AttachmentMarshaller)** - Method in class `javax.xml.bind.helpers.AbstractMarshallerImpl`
Associate a context that enables binary data within an XML document to be transmitted as XML-binary optimized attachment.

**setAttachmentUnmarshaller(AttachmentUnmarshaller)** - Method in class `javax.xml.bind.helpers.AbstractUnmarshallerImpl`
Associate a context that resolves cid's, content-id URIs, to binary data passed as attachments.

**setAttribute(ObjectName, Attribute)** - Method in interface `javax.management.j2ee.Management`
Sets the value of a specific attribute of a named managed object.

**setAttribute(String, Object)** - Method in interface `javax.servlet.http.HttpSession`
Binds an object to this session, using the name specified.

**setAttribute(String, Object)** - Method in class `javax.servlet.jsp.JspContext`
Register the name and value specified with page scope semantics.

**setAttribute(String, Object, int)** - Method in class `javax.servlet.jsp.JspContext`
Register the name and value specified with appropriate scope semantics.

**setAttribute(String, Object)** - Method in class `javax.servlet.jsp.tagext.TagData`
Set the value of an attribute.

**setAttribute(String, Object)** - Method in interface `javax.servlet.ServletContext`
Binds an object to a given attribute name in this servlet context.

**setAttribute(String, Object)** - Method in interface `javax.servlet.HttpServletRequest`
Stores an attribute in this request.

**setAttribute(String, Object)** - Method in class `javax.servlet.HttpServletRequestWrapper`
The default behavior of this method is to return `setAttribute(String name, Object o)` on the wrapped request object.
**setAttributes(ObjectName, AttributeList)** - Method in interface `javax.management.j2ee.Management`
Sets the values of several attributes of a named managed object.

**setAutocomplete(String)** - Method in class `javax.faces.component.html.HtmlInputSecret`
Set the value of the autocomplete property.

**setAutocomplete(String)** - Method in class `javax.faces.component.html.HtmlInputText`
Set the value of the autocomplete property.

**setBase64Content(InputStream, String)** - Method in class `javax.xml.soap.AttachmentPart`
Sets the content of this attachment part from the Base64 source InputStream and sets the value of the Content-Type header to the value contained in contentType, This method would first decode the base64 input and write the resulting raw bytes to the attachment.

**setBeforePhaseListener(MethodExpression)** - Method in class `javax.faces.component.UIViewRoot`
Allow an arbitrary method to be called for the "beforePhase" event as the UIViewRoot runs through its lifecycle.

**setBgcolor(String)** - Method in class `javax.faces.component.html.HtmlDataTable`
Set the value of the bgcolor property.

**setBgcolor(String)** - Method in class `javax.faces.component.html.HtmlPanelGrid`
Set the value of the bgcolor property.

**setBinding(String)** - Method in class `javax.faces.webapp.ConverterTag`
Deprecated. Set the expression that will be used to create a `ValueExpression` that references a backing bean property of the `Converter` instance to be created.

**setBinding(ValueExpression)** - Method in class `javax.faces.webapp.UIComponentELTag`
Set the value expression for our component.

**setBinding(String)** - Method in class `javax.faces.webapp.UIComponentTag`
Deprecated. Set the value binding expression for our component.

**setBinding(String)** - Method in class `javax.faces.webapp.ValidatorTag`
Deprecated. Set the expression that will be used to create a `ValueExpression` that references a backing bean property of the `Validator` instance to be created.

**setBodyContent(BodyContent)** - Method in class `javax.faces.webapp.UIComponentClassicTagBase`
Set the `bodyContent` for this tag handler.

**setBodyContent(BodyContent)** - Method in class `javax.servlet.jsp.tagext.BodyTag`
Set the `bodyContent` property.

**setBodyContent(BodyContent)** - Method in class
javax.servlet.jsp.tagext.**BodyTagSupport**

Prepare for evaluation of the body: stash the bodyContent away.

`setBoolean(String, boolean)` - Method in interface javax.jms.**MapMessage**

Sets a boolean value with the specified name into the Map.

`setBooleanProperty(String, boolean)` - Method in interface javax.jms.**Message**

Sets a boolean property value with the specified name into the message.

`setBorder(int)` - Method in class javax.faces.component.html.**HtmlDataTable**

Set the value of the border property.

`setBorder(int)` - Method in class javax.faces.component.html.**HtmlPanelGrid**

Set the value of the border property.

`setBorder(int)` - Method in class javax.faces.component.html.**HtmlSelectManyCheckbox**

Set the value of the border property.

`setBorder(int)` - Method in class javax.faces.component.html.**HtmlSelectOneRadio**

Set the value of the border property.

`setBufferSize(int)` - Method in interface javax.servlet.**ServletResponse**

Sets the preferred buffer size for the body of the response.

`setBufferSize(int)` - Method in class javax.servlet.**ServletResponseWrapper**

The default behavior of this method is to call `setBufferSize(int size)` on the wrapped response object.

`setBuilder(DocumentBuilder)` - Method in interface javax.xml.bind.annotation.**W3CDomHandler**

`setByte(String, byte)` - Method in interface javax.jms.**MapMessage**

Sets a byte value with the specified name into the Map.

`setByteProperty(String, byte)` - Method in interface javax.jms.**Message**

Sets a byte property value with the specified name into the message.

`setBytes(String, byte[])` - Method in interface javax.jms.**MapMessage**

Sets a byte array value with the specified name into the Map.

`setBytes(String, byte[], int, int)` - Method in interface javax.jms.**MapMessage**

Sets a portion of the byte array value with the specified name into the Map.

`setCalendar(Calendar)` - Method in class javax.mail.internet.**MailDateFormat**

Don't allow setting the calendar.

`setCaptionClass(String)` - Method in class javax.faces.component.html.**HtmlDataTable**

Set the value of the captionClass property.

`setCaptionClass(String)` - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the captionClass property.

setCaptionStyle(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the captionStyle property.

setCaptionStyle(String) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the captionStyle property.

setCellpadding(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the cellpadding property.

setCellpadding(String) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the cellpadding property.

setCellspacing(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the cellspacing property.

setCellspacing(String) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the cellspacing property.

setChangeEvent(PropertyChangeEvent) - Method in class
javax.enterprise.deploy.model.XpathEvent

setChar(String, char) - Method in interface javax.jms.MapMessage
Sets a Unicode character value with the specified name into the Map.

setCharacterEncoding(String) - Method in interface
javax.servlet.ServletRequest
Overrides the name of the character encoding used in the body of this request.

setCharacterEncoding(String) - Method in class
javax.servlet.ServletRequestWrapper
The default behavior of this method is to set the character encoding on the wrapped request object.

setCharacterEncoding(String) - Method in interface
javax.servlet.ServletResponse
Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8.

setCharacterEncoding(String) - Method in class
javax.servlet.ServletResponseWrapper
The default behavior of this method is to call setCharacterEncoding(String charset) on the wrapped response object.

setCharset(String) - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the charset property.

setCharset(String) - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the charset property.

setCharsetName(String) - Method in interface
javax.xml.registry.infomodel.LocalizedString
Set the canonical name for the charset for this object.

setCity(String) - Method in interface
javax.xml.registry.infomodel.PostalAddress
Sets the city.

setClassifications(Collection) - Method in interface
javax.xml.registry.infomodel.RegistryObject
Replaces all previous Classifications with specified
Classifications.

setClassificationScheme(ClassificationScheme) - Method in interface
javax.xml.registry.infomodel.Classification
Sets the ClassificationScheme for this external classification.

setClassifiedObject(RegistryObject) - Method in interface
javax.xml.registry.infomodel.Classification
Sets the object that is being classified.

setClientID(String) - Method in interface javax.jms.Connection
Sets the client identifier for this connection.

setCols(int) - Method in class
javax.faces.component.html.HtmlInputTextarea
Set the value of the cols property.

setColumnClasses(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the columnClasses property.

setColumnClasses(String) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the columnClasses property.

setColumnNumber(int) - Method in class
javax.xml.bind.helpers.ValidationEventLocatorImpl
Set the columnNumber field on this event locator.

setColumns(int) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the columns property.

setCommandContext(String, DataHandler) - Method in interface
javax.activation.CommandObject
Initialize the Command with the verb it is requested to handle
and the DataHandler that describes the data it will operate on.

setCommandMap(CommandMap) - Method in class
javax.activation.DataHandler
Set the CommandMap for use by this DataHandler.

setComment(String) - Method in class javax.servlet.http.Cookie
Specifies a comment that describes a cookie's purpose.

setConcept(Concept) - Method in interface
javax.xml.registry.infomodel.Classification
Sets the concept for this internal classification.

setConnected(boolean) - Method in class javax.mail.Service
Set the connection state of this service.

setConnectionHandle(Object) - Method in class
javax.resource.spi.ConnectionEvent
Set the connection handle.

setContent(Object, String) - Method in class
A convenience method for setting this body part's content.

**setContent(Multipart)** - Method in class

This method sets the body part's content to a Multipart object.

**setContent(Object, String)** - Method in class

A convenience method for setting this Message's content.

**setContent(Multipart)** - Method in class

This method sets the Message's content to a Multipart object.

**setContent(Object, String)** - Method in interface javax.mail.Part

A convenience method for setting this part's content.

**setContent(Multipart)** - Method in interface javax.mail.Part

This method sets the given Multipart object as this message's content.

**setContent(Object, String)** - Method in class

Sets the content of this attachment part to that of the given Object and sets the value of the Content-Type header to the given type.

**setContent(Source)** - Method in class javax.xml.soap.SOAPPart

Sets the content of the SOAPEnvelop object with the data from the given Source object.

**setContentDescription(String)** - Method in class

Sets the description of this SOAPMessage object's content with the given description.

**setContentID(String)** - Method in class

Set the "Content-ID" header field of this body part.

**setContentID(String)** - Method in class

Set the "Content-ID" header field of this Message.

**setContentID(String)** - Method in class

Sets the MIME header whose name is "Content-ID" with the given value.

**setContentLanguage(String[])** - Method in class

Set the Content-Language header of this MimePart.

**setContentLanguage(String[])** - Method in class

Set the "Content-Language" header of this MimePart.

**setContentLanguage(String[])** - Method in interface

Set the Content-Language header of this MimePart.
**setContentType(String)** - Method in interface
javax.servlet.
Sets the content type of the response being sent to the client, if the response has not been committed yet.

**setContentType(String)** - Method in class
javax.servlet.
The default behavior of this method is to call
setContentType(String type) on the wrapped response object.

**setContentType(String)** - Method in class
javax.xml.soap.
Sets the MIME header whose name is "Content-Type" with the given value.

**setContextID(String)** - Static method in class
javax.security.jacc.
Authorization protected method used to modify the value of the policy context identifier associated with the thread on which this method is called.

**setConverter(Converter)** - Method in class
javax.faces.component.
Set the Converter (if any) that is registered for this UIComponent.

**setConverterId(String)** - Method in class
javax.faces.component.

javax.faces.webapp.ConverterTag
   Deprecated. Set the identifier of the Converter instance to be created.

setConverterMessage(String) - Method in class
   javax.faces.component.UIInput
   Override any ValueExpression set for the "converterMessage"
   with the literal argument provided to this method.

setCoords(String) - Method in class
   javax.faces.component.html.HtmlCommandLink
   Set the value of the coords property.

setCoords(String) - Method in class
   javax.faces.component.html.HtmlOutputLink
   Set the value of the coords property.

setCountry(String) - Method in interface
   javax.xml.registry.infomodel.PostalAddress
   Sets the country.

setCountryCode(String) - Method in interface
   javax.xml.registry.infomodel.TelephoneNumber
   Sets country code.

setCredentials(Set) - Method in interface
   javax.xml.registry.Connection
   Sets the Credentials associated with this client.

setCurrencyCode(String) - Method in class
   javax.faces.convert.NumberConverter
   Set the ISO 4217 currency code used by getAsString() with a
   type of currency.

setCurrencySymbol(String) - Method in class
   javax.faces.convert.NumberConverter
   Set the currency symbol used by getAsString() with a type of
   currency.

setCurrentInstance(FacesContext) - Static method in class
   javax.faces.context.FacesContext
   Set the FacesContext instance for the request that is being
   processed by the current thread.

setDataContentHandlerFactory(DataContentHandlerFactory) - Static
   method in class javax.activation.DataHandler
   Sets the DataContentHandlerFactory.

setDataHandler(DataHandler) - Method in class
   javax.mail.internet.MimeBodyPart
   This method provides the mechanism to set this body part's
   content.

setDataHandler(DataHandler) - Method in class
   javax.mail.internet.MimeMessage
   This method provides the mechanism to set this part's content.

setDataHandler(DataHandler) - Method in interface javax.mail.Part
   This method provides the mechanism to set this part's content.

setDataHandler(DataHandler) - Method in class
   javax.xml.soap.AttachmentPart
   Sets the given DataHandler object as the data handler for this
   AttachmentPart object.
**setDataModel(DataModel)** - Method in class javax.faces.component.UIData
   Set the internal DataModel.

**setDatatypeConverter(DatatypeConverterInterface)** - Static method in class javax.xml.bind.DatatypeConverter
   This method is for JAXB provider use only.

**setDateHeader(String, long)** - Method in interface javax.servlet.http.HttpServletRequest
   Sets a response header with the given name and date-value.

**setDateHeader(String, long)** - Method in class javax.servlet.http.HttpServletResponseWrapper
   The default behavior of this method is to call 
   setDateHeader(String name, long date) on the wrapped response object.

**setDateStyle(String)** - Method in class javax.faces.convert.DateTimeConverter
   Set the style to be used to format or parse dates.

**setDConfigBeanVersion(DConfigBeanVersionType)** - Method in interface javax.enterprise.deploy.spi.DeploymentManager
   Set the configuration beans to be used to the J2EE platform version specified.

**setDebug(boolean)** - Method in class javax.mail.Session
   Set the debug setting for this Session.

**setDebugOut(PrintStream)** - Method in class javax.mail.Session
   Set the stream to be used for debugging output for this session.

**setDefaultCommandMap(CommandMap)** - Static method in class javax.activation.CommandMap
   Set the default CommandMap.

**setDefaultFactory(JspFactory)** - Static method in class javax.servlet.jsp.JspFactory
   Set the default factory for this implementation.

**setDefaultFileTypeMap(FileTypeMap)** - Static method in class javax.activation.FileTypeMap
   Sets the default FileTypeMap for the system.

**setDefaultLocale(Locale)** - Method in class javax.faces.application.Application
   Set the default Locale for this application.

**setDefaultNamespace(String)** - Method in interface javax.xml.stream.XMLEventWriter
   Binds a URI to the default namespace This URI is bound in the 
   scope of the current START_ELEMENT / END_ELEMENT pair.

**setDefaultNamespace(String)** - Method in interface javax.xml.stream.XMLStreamWriter
   Binds a URI to the default namespace This URI is bound in the 
   scope of the current START_ELEMENT / END_ELEMENT pair.

**setDefaultRenderKitId(String)** - Method in class javax.faces.application.Application
   Set the renderKitId to be used to render this application.

**setDeliveryMode(int)** - Method in interface

---

**setDataModel(DataModel)** - Method in class javax.faces.component.UIData
   Set the internal DataModel.

**setDatatypeConverter(DatatypeConverterInterface)** - Static method in class javax.xml.bind.DatatypeConverter
   This method is for JAXB provider use only.

**setDateHeader(String, long)** - Method in interface javax.servlet.http.HttpServletRequest
   Sets a response header with the given name and date-value.

**setDateHeader(String, long)** - Method in class javax.servlet.http.HttpServletResponseWrapper
   The default behavior of this method is to call 
   setDateHeader(String name, long date) on the wrapped response object.

**setDateStyle(String)** - Method in class javax.faces.convert.DateTimeConverter
   Set the style to be used to format or parse dates.

**setDConfigBeanVersion(DConfigBeanVersionType)** - Method in interface javax.enterprise.deploy.spi.DeploymentManager
   Set the configuration beans to be used to the J2EE platform version specified.

**setDebug(boolean)** - Method in class javax.mail.Session
   Set the debug setting for this Session.

**setDebugOut(PrintStream)** - Method in class javax.mail.Session
   Set the stream to be used for debugging output for this session.

**setDefaultCommandMap(CommandMap)** - Static method in class javax.activation.CommandMap
   Set the default CommandMap.

**setDefaultFactory(JspFactory)** - Static method in class javax.servlet.jsp.JspFactory
   Set the default factory for this implementation.

**setDefaultFileTypeMap(FileTypeMap)** - Static method in class javax.activation.FileTypeMap
   Sets the default FileTypeMap for the system.

**setDefaultLocale(Locale)** - Method in class javax.faces.application.Application
   Set the default Locale for this application.

**setDefaultNamespace(String)** - Method in interface javax.xml.stream.XMLEventWriter
   Binds a URI to the default namespace This URI is bound in the 
   scope of the current START_ELEMENT / END_ELEMENT pair.

**setDefaultNamespace(String)** - Method in interface javax.xml.stream.XMLStreamWriter
   Binds a URI to the default namespace This URI is bound in the 
   scope of the current START_ELEMENT / END_ELEMENT pair.

**setDefaultRenderKitId(String)** - Method in class javax.faces.application.Application
   Set the renderKitId to be used to render this application.

**setDeliveryMode(int)** - Method in interface
javax.jms.**MessageProducer**
Sets the producer's default delivery mode.

**setDescription(String)** - Method in class

javax.faces.model.**SelectItem**
Set the description of this item, for use in development tools.

**setDescription(String)** - Method in class

javax.mail.internet.**MimeBodyPart**
Set the "Content-Description" header field for this body part.

**setDescription(String, String)** - Method in class

javax.mail.internet.**MimeBodyPart**
Set the "Content-Description" header field for this body part.

**setDescription(String)** - Method in class

javax.mail.internet.**MimeMessage**
Set the "Content-Description" header field for this Message.

**setDescription(String, String)** - Method in class

javax.mail.internet.**MimeMessage**
Set the "Content-Description" header field for this Message.

**setDescription(String)** - Method in class

javax.mail.internet.**Part**
Set a description String for this part.

**setDescription(InternationalString)** - Method in class

javax.xml.registry.infomodel.**RegistryObject**
Sets the context independent textual description for this object.

**setDetail(String)** - Method in class

javax.faces.application.**FacesMessage**
Set the localized detail text.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlCommandButton**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlCommandLink**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlDataTable**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlForm**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlGraphicImage**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlInputSecret**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlInputText**
Set the value of the dir property.

**setDir(String)** - Method in class

javax.faces.component.html.**HtmlInputTextarea**
Set the value of the dir property.
**setDir(String)** - Method in class
javax.faces.component.html.HtmlMessage
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlMessages
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlOutputFormat
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlOutputLabel
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlOutputText
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Set the value of the dir property.

**setDir(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the dir property.

**setDisabled(boolean)** - Method in class
javax.faces.component.html.HtmlCommandButton
Set the value of the disabled property.

**setDisabled(boolean)** - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the disabled property.

**setDisabled(boolean)** - Method in class
javax.faces.component.html.HtmlInputSecret
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlInputText  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlInputTextarea  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlOutputLink  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectBooleanCheckbox  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectManyCheckbox  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectManyListbox  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectManyMenu  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectOneListbox  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectOneMenu  
Set the value of the disabled property.  
**setDisabled(boolean)** - Method in class  
javax.faces.component.html.HtmlSelectOneRadio  
Set the value of the disabled property.  
Set the disabled flag for this item, which should modify the rendered output to make this item unavailable for selection by the user if set to true.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectManyCheckbox  
Set the value of the disabledClass property.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectManyListbox  
Set the value of the disabledClass property.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectManyMenu  
Set the value of the disabledClass property.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectOneListbox  
Set the value of the disabledClass property.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectOneMenu  
Set the value of the disabledClass property.  
**setDisabledClass(String)** - Method in class  
javax.faces.component.html.HtmlSelectOneMenu
**setDisabledClass(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the disabledClass property.

**setDisableMessageID(boolean)** - Method in interface
javax.jms.MessageProducer
Sets whether message IDs are disabled.

**setDisableMessageTimestamp(boolean)** - Method in interface
javax.jms.MessageProducer
Sets whether message timestamps are disabled.

**setDisposition(String)** - Method in class
javax.mail.internet.ContentDisposition
Set the disposition.

**setDisposition(String)** - Method in class
javax.mail.internet.MimeBodyPart
Set the "Content-Disposition" header field of this body part.

**setDisposition(String)** - Method in class
javax.mail.internet.MimeMessage
Set the "Content-Disposition" header field of this Message.

**setDisposition(String)** - Method in interface javax.mail.Part
Set the disposition of this part.

**setDomain(String)** - Method in class javax.servlet.http.Cookie
Specifies the domain within which this cookie should be presented.

**setDouble(String, double)** - Method in interface
javax.jms.MapMessage
Sets a double value with the specified name into the Map.

**setDoubleProperty(String, double)** - Method in interface
javax.jms.Message
Sets a double property value with the specified name into the message.

**setDynamicAttribute(String, String, Object)** - Method in interface
javax.servlet.jsp.tagext.DynamicAttributes
Called when a tag declared to accept dynamic attributes is passed an attribute that is not declared in the Tag Library Descriptor.

**setElementQName(QName)** - Method in interface
javax.xml.soap.SOAPElement
Changes the name of this Element to newName if possible.

**setEmailAddresses(Collection)** - Method in interface
javax.xml.registry.infomodel.User
Sets the Collection of EmailAddress instances for this User.

**setEnabledClass(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Set the value of the enabledClass property.

**setEnabledClass(String)** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Set the value of the enabledClass property.

**setEnabledClass(String)** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Set the value of the enabledClass property.
setEnabledClass(String) - Method in class
javax.faces.component.html.HtmSelectOneListbox
  Set the value of the enabledClass property.
setEnabledClass(String) - Method in class
javax.faces.component.html.HtmSelectOneMenu
  Set the value of the enabledClass property.
setEnabledClass(String) - Method in class
javax.faces.component.html.HtmSelectOneRadio
  Set the value of the enabledClass property.
setEncoding(String) - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl
  Convenience method for setting the output encoding.
setEncodingStyle(String) - Method in interface
javax.xml.soap.SOAPElement
  Sets the encoding style for this SOAPElement object to one specified.
setEncType(String) - Method in class
javax.faces.component.html.HtmForm
  Set the value of the encType property.
setEntityContext(EntityContext) - Method in interface
javax.ejb.EntityBean
  Set the associated entity context.
setErrorClass(String) - Method in class
javax.faces.component.html.HtmMessage
  Set the value of the errorClass property.
setErrorClass(String) - Method in class
javax.faces.component.html.HtmMessages
  Set the value of the errorClass property.
setErrorCode(String) - Method in exception
javax.resource.ResourceException
  Set the error code.
setErrorObjectKey(Key) - Method in exception
javax.xml.registry.RegistryException
  Sets the Key to the first object that encountered an error in the registry.
setErrorStyle(String) - Method in class
javax.faces.component.html.HtmMessage
  Set the value of the errorStyle property.
setErrorStyle(String) - Method in class
javax.faces.component.html.HtmMessages
  Set the value of the errorStyle property.
setEscape(boolean) - Method in class
javax.faces.component.html.HtmOutputFormat
  Set the value of the escape property.
setEscape(boolean) - Method in class
javax.faces.component.html.HtmOutputLabel
  Set the value of the escape property.
setEscape(boolean) - Method in class
javax.faces.component.html.HtmOutputText
  Set the value of the escape property.
**setEscape(boolean)** - Method in class `javax.faces.model.SelectItem`
  Setter for property escape.

**setEventAllocator(XMLEventAllocator)** - Method in class `javax.xml.stream.XMLInputFactory`
  Set a user defined event allocator for events.

**setEventHandler(ValidationEventHandler)** - Method in class `javax.xml.bind.Binder`
  Allow an application to register a ValidationEventHandler.

**setEventHandler(ValidationEventHandler)** - Method in class `javax.xml.bind.helpers.AbstractMarshallerImpl`

**setEventHandler(ValidationEventHandler)** - Method in class `javax.xml.bind.helpers.AbstractUnmarshallerImpl`

**setEventHandler(ValidationEventHandler)** - Method in interface `javax.xml.bind.Marerisher`
  Allow an application to register a validation event handler.

**setEventHandler(ValidationEventHandler)** - Method in interface `javax.xml.bind.Unmarshaller`
  Allow an application to register a validation event handler.

**setEventHandler(ValidationEventHandler)** - Method in interface `javax.xml.bind.Validator`
  Deprecated. since JAXB2.0

**setExceptionListener(ExceptionListener)** - Method in interface `javax.jms.Connection`
  Sets an exception listener for this connection.

**setExecutor(Executor)** - Method in class `javax.xml.ws.Endpoint`
  Sets the executor for this Endpoint instance.

**setExecutor(Executor)** - Method in class `javax.xml.ws.Service`
  Sets the executor for this Service instance.

**setExecutor(Executor)** - Method in class `javax.xml.ws.spi.ServiceDelegate`
  Sets the executor for this Service instance.

**setExpiration(Date)** - Method in interface `javax.xml.registry.infomodel.RegistryEntry`
  Sets the expirationDate.

**setExpunged(boolean)** - Method in class `javax.mail.Message`
  Sets the expunged flag for this Message.

**setExtension(String)** - Method in interface `javax.xml.registry.infomodel.TelephoneNumber`
  Sets the internal extension.

**setExternalIdentifiers(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`
  Replaces all previous external identifiers with specified Collection of ExternalIdentifiers as an external identifier.

**setExternalLinks(Collection)** - Method in interface `javax.xml.registry.infomodel.RegistryObject`
  Replaces all previous ExternalLinks with specified ExternalLinks.

**setExternalURI(String)** - Method in interface
javax.xml.registry.infomodel.Externallink
Sets URI for an external resource.

**setFactory(String, String)** - Static method in class javax.faces.FactoryFinder
This method will store the argument factoryName/implName mapping in such a way that FactoryFinder.getFactory(java.lang.String) will find this mapping when searching for a match.

**setFatalClass(String)** - Method in class javax.faces.component.html.HtmlMessage
Set the value of the fatalClass property.

**setFatalClass(String)** - Method in class javax.faces.component.html.HtmlMessages
Set the value of the fatalClass property.

**setFatalStyle(String)** - Method in class javax.faces.component.html.HtmlMessage
Set the value of the fatalStyle property.

**setFatalStyle(String)** - Method in class javax.faces.component.html.HtmlMessages
Set the value of the fatalStyle property.

**setFaultActor(String)** - Method in interface javax.xml.soap.SOAPFault
Sets this SOAPFault object with the given fault actor.

**setFaultCode(Name)** - Method in interface javax.xml.soap.SOAPFault
Sets this SOAPFault object with the given fault code.

**setFaultCode(QName)** - Method in interface javax.xml.soap.SOAPFault
Sets this SOAPFault object with the given fault code.

**setFaultCode(String)** - Method in interface javax.xml.soap.SOAPFault
Sets this SOAPFault object with the given fault code.

**setFaultNode(String)** - Method in interface javax.xml.soap.SOAPFault
Creates or replaces any existing Node element value for this SOAPFault object.

**setFaultRole(String)** - Method in interface javax.xml.soap.SOAPFault
Creates or replaces any existing Role element value for this SOAPFault object.

**setFaultString(String)** - Method in interface javax.xml.soap.SOAPFault
Sets the fault string for this SOAPFault object to the given string.

**setFaultString(String, Locale)** - Method in interface javax.xml.soap.SOAPFault
Sets the fault string for this SOAPFault object to the given string and localized to the given locale.

**setFileName(String)** - Method in class javax.mail.internet.MimeBodyPart
Set the filename associated with this body part, if possible.

**setFileName(String)** - Method in class javax.mail.internet.MimeMessage
Set the filename associated with this part, if possible.

**setFileName(String)** - Method in interface javax.mail.Part
Set the filename associated with this part, if possible.

**setFileTypeMap(FileTypeMap)** - Method in class javax.activation.FileDataSource
Set the FileTypeMap to use with this FileDataSource

**setFirst(int)** - Method in class javax.faces.component.UIData
Set the zero-relative row number of the first row to be displayed.

**setFirstName(String)** - Method in interface javax.xml.registry.infomodel.PersonName
Sets the first name for this Person.

**setFirstResult(int)** - Method in interface javax.persistence.Query
Set the position of the first result to retrieve.

**setFlag(Flags.Flag, boolean)** - Method in class javax.mail.Message
Set the specified flag on this message to the specified value.

**setFlags(Message[], Flags, boolean)** - Method in class javax.mail.Folder
Set the specified flags on the messages specified in the array.

**setFlags(int, int, Flags, boolean)** - Method in class javax.mail.Folder
Set the specified flags on the messages numbered from start through end, both start and end inclusive.

**setFlags(int[], Flags, boolean)** - Method in class javax.mail.Folder
Set the specified flags on the messages whose message numbers are in the array.

**setFlags(Flags, boolean)** - Method in class javax.mail.internet.MimeMessage
Set the flags for this message.

**setFlags(Flags, boolean)** - Method in class javax.mail.Message
Set the specified flags on this message to the specified value.

**setFloat(String, float)** - Method in interface javax.jms.MapMessage
Sets a float value with the specified name into the Map.

**setFloatProperty(String, float)** - Method in interface javax.jms.Message
Sets a float property value with the specified name into the message.

**setFlushMode(FlushModeType)** - Method in interface javax.persistence.EntityManager
Set the flush mode that applies to all objects contained in the persistence context.

**setFlushMode(FlushModeType)** - Method in interface javax.persistence.Query
Set the flush mode type to be used for the query execution.

**setFooter(UIComponent)** - Method in class javax.faces.component.UIColumn
Set the footer facet of the column.

**setFooter(UIComponent)** - Method in class javax.faces.component.UIData
Set the footer facet of this component.

**setFooterClass(String)** - Method in class javax.faces.component.html.HtmlColumn
Set the value of the footerClass property.

**setFooterClass(String)** - Method in class
javax.faces.component.html.HtmlDataTable
  Set the value of the footerClass property.

**setFooterClass(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
  Set the value of the footerClass property.

**setFor(String)** - Method in class
javax.faces.component.html.HtmlOutputLabel
  Set the value of the for property.

**setFor(String)** - Method in class javax.faces.component.UIMessage
  Set the client identifier of the component for which this component represents associated message(s) (if any).

**setFormattedOutput(boolean)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl
  Convenience method for setting the formatted output flag.

**setFragment(boolean)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl
  Convenience method for setting the fragment flag.

**setFrame(String)** - Method in class
javax.faces.component.html.HtmlDataTable
  Set the value of the frame property.

**setFrame(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
  Set the value of the frame property.

**setFrom(Address)** - Method in class
javax.mail.internet.MimeMessage
  Set the RFC 822 "From" header field.

**setFrom()** - Method in class javax.mail.internet.MimeMessage
  Set the RFC 822 "From" header field using the value of the InternetAddress.getLocalAddress method.

**setFrom()** - Method in class javax.mail.Message
  Set the "From" attribute in this Message.

**setFrom(Address)** - Method in class javax.mail.Message
  Set the "From" attribute in this Message.

**setFullName(String)** - Method in interface
javax.xml.registry.infomodel.PersonName
  Sets the fully formatted name for this person.

**setGlobalOnly(boolean)** - Method in class
javax.faces.component.UIMessages
  Set the flag indicating whether only global messages (that is, messages with no associated client identifier) should be rendered.

**setGroupingUsed(boolean)** - Method in class
javax.faces.convert.NumberConverter
  Set the flag indicating whether getAsString() should include grouping separators if necessary.

**setHandlerChain(QName, List)** - Method in interface
javax.xml.rpc.handler.HandlerRegistry
  Sets the handler chain for the specified service endpoint as a java.util.List.
**setHandlerChain(List<Handler>)** - Method in interface `javax.xml.ws.Binding`  
Sets the handler chain for the protocol binding instance.

**setHandlerClass(Class)** - Method in class `javax.xml.rpc.handler.HandlerInfo`  
Sets the Handler class

**setHandlerConfig(Map)** - Method in class `javax.xml.rpc.handler.HandlerInfo`  
Sets the Handler configuration as `java.util.Map`

**setHandlerData(Object)** - Static method in class `javax.security.jacc.PolicyContext`  
Authorization protected method that may be used to associate a thread-scoped handler data object with the PolicyContext.

**setHandlerResolver(HandlerResolver)** - Method in class `javax.xml.ws.Service`  
Sets the `HandlerResolver` for this `Service` instance.

**setHandlerResolver(HandlerResolver)** - Method in class `javax.xml.ws.spi.ServiceDelegate`  
Sets the `HandlerResolver` for this `Service` instance.

**setHeader(UIComponent)** - Method in class `javax.faces.component.UIColumn`  
Set the header facet of the column.

**setHeader(UIComponent)** - Method in class `javax.faces.component.UIData`  
Set the header facet of this component.

**setHeader(String, String)** - Method in class `javax.mail.internet.InternetHeaders`  
Change the first header line that matches name to have value, adding a new header if no existing header matches.

**setHeader(String, String)** - Method in class `javax.mail.internet.MimeBodyPart`  
Set the value for this header name.

**setHeader(String, String)** - Method in class `javax.mail.internet.MimeMessage`  
Set the value for this header name.

**setHeader(String, String)** - Method in interface `javax.mail.Part`  
Set the value for this header name.

**setHeader(String, String)** - Method in interface `javax.servlet.http.HttpServletResponse`  
Sets a response header with the given name and value.

**setHeader(String, String)** - Method in class `javax.servlet.http.HttpServletResponseWrapper`  
The default behavior of this method is to return `setHeader(String name, String value)` on the wrapped response object.

**setHeader(String, String)** - Method in class `javax.xml.soap.MimeHeaders`  
Replaces the current value of the first header entry whose name matches the given name with the given value, adding a new header if no existing header name matches.
**setHeaderClass(String)** - Method in class javax.faces.component.html.HtmlColumn
Set the value of the headerClass property.

**setHeaderClass(String)** - Method in class javax.faces.component.html.HtmlDataTable
Set the value of the headerClass property.

**setHeaderClass(String)** - Method in class javax.faces.component.html.HtmlPanelGrid
Set the value of the headerClass property.

**setHeaders(QName[])** - Method in class javax.xml.rpc.handler.HandlerInfo
Sets the header blocks processed by this Handler.

**setHeight(String)** - Method in class javax.faces.component.html.HtmlGraphicImage
Set the value of the height property.

**setHint(String, Object)** - Method in interface javax.persistence.Query
Set an implementation-specific hint.

**setHost(String)** - Method in class javax.mail.internet.NewsAddress
Set the host.

**setHreflang(String)** - Method in class javax.faces.component.html.HtmlCommandLink
Set the value of the hreflang property.

**setHreflang(String)** - Method in class javax.faces.component.html.HtmlOutputLink
Set the value of the hreflang property.

**setHumanPresentableName(String)** - Method in class javax.activation.ActivationDataFlavor
Set the human presentable name.

**setId(String)** - Method in class javax.faces.component.UIComponent
Set the component identifier of this UIComponent (if any).

**setId(String)** - Method in class javax.faces.component.UIComponentBase

**setId(String)** - Method in class javax.faces.webapp.UIComponentClassicTagBase
Set the component identifier for our component.

**setId(String)** - Method in class javax.faces.webapp.UIComponentTagBase
Set the component identifier for the component corresponding to this tag instance.

**setId(String)** - Method in class javax.servlet.jsp.tagext.TagSupport
Set the id attribute for this tag.

**setId(String)** - Method in interface javax.xml.registry.infomodel.Key
Sets the unique id associated with this key.

**setIdentificationScheme(ClassificationScheme)** - Method in interface javax.xml.registry.infomodel.ExternalIdentifier
Sets the ClassificationScheme that is used as the identification scheme for identifying this object.
**setImage(String)** - Method in class
javax.faces.component.html.HtmCommandButton
Set the value of the image property.

**setImmediate(boolean)** - Method in interface
javax.faces.component.ActionSource
Set the "immediate execution" flag for this UIComponent.

**setImmediate(boolean)** - Method in interface
javax.faces.componentEditableValueHolder
Set the "immediate" state for this component.

**setImmediate(boolean)** - Method in class
javax.faces.component.UICommand

**setImmediate(boolean)** - Method in class
javax.faces.component.UIInput

**setInfoClass(String)** - Method in class
javax.faces.component.html.HtmMessage
Set the value of the infoClass property.

**setInfoClass(String)** - Method in class
javax.faces.component.html.HtmMessages
Set the value of the infoClass property.

**setInfoStyle(String)** - Method in class
javax.faces.component.html.HtmMessage
Set the value of the infoStyle property.

**setInfoStyle(String)** - Method in class
javax.faces.component.html.HtmMessages
Set the value of the infoStyle property.

**setInitParameters(Map<String, Object>)** - Method in class
javax.servlet.jsp.tagext.TagLibraryValidator
Set the init data in the TLD for this validator.

**setInt(String, int)** - Method in interface javax.jms.MapMessage
Sets an int value with the specified name into the Map.

**setIntegerOnly(boolean)** - Method in class
javax.faces.convert.NumberConverter
Set to true if only the integer portion of the given value should be returned from getAsObject().

**setIntHeader(String, int)** - Method in interface
javax.servlet.http.HttpServletRequest
Sets a response header with the given name and integer value.

**setIntHeader(String, int)** - Method in class
javax.servlet.http.HttpServletRequestWrapper
The default behavior of this method is to call
setIntHeader(String name, int value) on the wrapped response object.

**setIntProperty(String, int)** - Method in interface javax.jms.Message
Sets an int property value with the specified name into the message.

**setInvalidPropertyDescriptors(PropertyDescriptor[])** - Method in
exception javax.resource.spi.InvalidPropertyException
Set a list of invalid properties.
**setIsmap(boolean)** - Method in class javax.faces.component.html.HtmlGraphicImage
Set the value of the ismap property.

**setItemDescription(String)** - Method in class javax.faces.component.UISelectItem
Set the description for this selection item.

**setItemDisabled(boolean)** - Method in class javax.faces.component.UISelectItem
Set the disabled value for this selection item.

**setItemEscaped(boolean)** - Method in class javax.faces.component.UISelectItem
Set the escape value for the label of this selection item.

**setItemLabel(String)** - Method in class javax.faces.component.UISelectItem
Set the localized label for this selection item.

**setItemValue(Object)** - Method in class javax.faces.component.UISelectItem
Set the server value for this selection item.

**setJMSCorrelationID(String)** - Method in interface javax.jms.Message
Sets the correlation ID for the message.

**setJMSCorrelationIDAsBytes(byte[])** - Method in interface javax.jms.Message
Sets the correlation ID as an array of bytes for the message.

**setJMSDeliveryMode(int)** - Method in interface javax.jms.Message
Sets the DeliveryMode value for this message.

**setJMSDestination(Destination)** - Method in interface javax.jms.Message
Sets the Destination object for this message.

**setJMSExpiration(long)** - Method in interface javax.jms.Message
Sets the message's expiration value.

**setJMSMessageID(String)** - Method in interface javax.jms.Message
Sets the message ID.

**setJMSPriority(int)** - Method in interface javax.jms.Message
Sets the priority level for this message.

**setJMSRedelivered(boolean)** - Method in interface javax.jms.Message
Specifies whether this message is being redelivered.

**setJMSReplyTo(Destination)** - Method in interface javax.jms.Message
Sets the Destination object to which a reply to this message should be sent.

**setJMSTimestamp(long)** - Method in interface javax.jms.Message
Sets the message timestamp.

**setJMSType(String)** - Method in interface javax.jms.Message
Sets the message type.

**setJspBody(JspFragment)** - Method in interface javax.servlet.jsp.tagext.SimpleTag
Provides the body of this tag as a JspFragment object, able to be invoked zero or more times by the tag handler.

**setJspBody(JspFragment)** - Method in class javax.servlet.jsp.tagext.SimpleTagSupport
Stores the provided JspFragment.
setJspContext(JspContext) - Method in interface javax.servlet.jsp.tagext.SimpleTag
   Called by the container to provide this tag handler with the JspContext for this invocation.

setJspContext(JspContext) - Method in class javax.servlet.jsp.tagext.SimpleTagSupport
   Stores the provided JSP context in the private jspContext field.

setJspId(String) - Method in class javax.faces.webapp.UIComponentClassicTagBase
   Defined on JspIdConsumer.

setJspId(String) - Method in interface javax.servlet.jsp.tagext.JspIdConsumer
   Called by the container generated code to set a value for the jspId attribute.

setKey(Key) - Method in interface javax.xml.registry.infomodel.RegistryObject
   Sets the key representing the universally unique ID (UUID) for this object.

setLabel(String) - Method in class javax.faces.component.html.HtmlCommandButton
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlInputSecret
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlInputText
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlInputTextarea
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlSelectManyListbox
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlSelectOneListbox
   Set the value of the label property.

setLabel(String) - Method in class javax.faces.component.html.HtmlSelectOneMenu
   Set the value of the label property.

setLabel(String) - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the label property.

setLabel(String) - Method in class javax.faces.model.SelectItem
Set the label of this item, to be rendered visibly for the user.

setLang(String) - Method in class
javax.faces.component.html.HtmlCommandButton
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlForm
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlGraphicImage
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlInputSecret
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlInputText
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlInputTextarea
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlMessage
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlMessages
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlOutputFormat
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlOutputLabel
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlOutputText
Set the value of the lang property.

setLang(String) - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the lang property.
**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
  Set the value of the lang property.

**setLang(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
  Set the value of the lang property.

**setLastName(String)** - Method in interface
javax.xml.registry.infomodel.PersonName
  Sets the last name (surname) for this Person.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlMessages
  Set the value of the layout property.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlPanelGroup
  Set the value of the layout property.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
  Set the value of the layout property.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
  Set the value of the layout property.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
  Set the value of the layout property.

**setLayout(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
  Set the value of the layout property.

**setLineNumber(int)** - Method in class
javax.xml.bind.helpers.ValidationEventLocatorImpl
  Set the lineNumber field on this event locator.

**setLinkedException(Exception)** - Method in exception
javax.jms.JMSException
  Adds a linked Exception.

**setLinkedException(Exception)** - Method in exception
javax.resource.ResourceException
  Deprecated. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to get Cause and initCause methods of the java.lang.Throwable class.

**setLinkedException(Throwable)** - Method in class
javax.xml.bind.helpers.**ValidationEventImpl**
Set the linked exception field of this event.
`setLinkedException(Throwable)` - Method in exception
javax.xml.bind.**JAXBException**
Add a linked Exception.
`setLinkedException(Throwable)` - Method in exception
javax.xml.bind.**TypeConstraintException**
Add a linked Exception.
`setLinkedException(Throwable)` - Method in exception
javax.resource.cci.**ResourceWarning**
`Deprecated`. J2SE release 1.4 supports a chained exception facility that allows any throwable to know about another throwable, if any, that caused it to get thrown. Refer to `getCause` and `initCause` methods of the `java.lang.Throwable` class.
`setListener(Marshaller.Listener)` - Method in class
javax.xml.bind.helpers.**AbstractMarshallerImpl**
`setListener(Unmarshaller.Listener)` - Method in class
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**
`setListener(Marshaller.Listener)` - Method in interface
javax.xml.bind.**Marshaller**
Register marshal event callback `Marshaller.Listener` with this `Marshaller`.
`setListener(Unmarshaller.Listener)` - Method in interface
javax.xml.bind.**Unmarshaller**
Register unmarshal event callback `Unmarshaller.Listener` with this `Unmarshaller`.
`setLocale(Locale)` - Method in class
javax.el.**ELContext**
Set the Locale for this instance.
`setLocale(Locale)` - Method in interface
javax.enterprise.deploy.spi.**DeploymentManager**
Set the active locale for this implementation of `javax.enterprise.deploy.spi` subpackages to run.
`setLocale(Locale)` - Method in class
javax.faces.component.**UIViewRoot**
Set the Locale to be used in localizing the response being created for this view.
`setLocale(Locale)` - Method in class
javax.faces.convert.**DateTimeConverter**
Set the Locale to be used when parsing or formatting dates and times.
`setLocale(Locale)` - Method in class
javax.faces.convert.**NumberConverter**
Set the Locale to be used when parsing numbers.
`setLocale(Locale)` - Method in interface
javax.servlet.**ServletResponse**
Sets the locale of the response, if the response has not been committed yet.
**setLocale(Locale)** - Method in class javax.servlet.ServletResponseWrapper
The default behavior of this method is to call setLocale(Locale loc) on the wrapped response object.

**setLocale(Locale)** - Method in interface javax.xml.registry.infomodelLocalizedString
Set the Locale for this object.

**setLocalValueSet(boolean)** - Method in interface javax.faces.component.EditableValueHolder
Sets the "local value set" state for this component.

**setLocalValueSet(boolean)** - Method in class javax.faces.component.UIInput
Sets the "local value set" state for this component.

**setLocation(Location)** - Method in class javax.xml.stream.XMLEventFactory
This method allows setting of the Location on each event that is created by this factory.

**setLocator(ValidationEventLocator)** - Method in class javax.resource.spi.security.PasswordCredential
Sets the target ManagedConnectionFactory instance for which the user name and password has been set by the application server.

**setMaxAge(int)** - Method in class javax.servlet.http.Cookie
Sets the maximum age of the cookie in seconds.

**setMaxFractionDigits(int)** - Method in class javax.faces.convert.NumberConverter
Set the maximum number of digits getAsString() should render in the fraction portion of the result.

**setMaximum(double)** - Method in class
javax.faces.validator.**DoubleRangeValidator**
Set the maximum value to be enforced by this **Validator**.
`setMaximum(int)` - Method in class

javax.faces.validator.**LengthValidator**
Set the maximum length to be enforced by this **Validator**.
`setMaximum(long)` - Method in class

javax.faces.validator.**LongRangeValidator**
Set the maximum value to be enforced by this **Validator**.
`setMaxInactiveInterval(int)` - Method in interface
javax.servlet.http.**HttpSession**
Specifies the time, in seconds, between client requests before the servlet container will invalidate this session.

javax.faces.convert.**NumberConverter**
Set the maximum number of digits `getAsString()` should render in the integer portion of the result.
`setMaxLength(int)` - Method in class

javax.faces.component.html.**HtmlInputSecret**
Set the value of the `maxlength` property.
`setMaxLength(int)` - Method in class

javax.faces.component.html.**HtmlInputText**
Set the value of the `maxlength` property.
`setMaxResults(int)` - Method in interface javax.persistence.**Query**
Set the maximum number of results to retrieve.
`setMessage(String)` - Method in class
javax.xml.bind.helpers.**ValidationEventImpl**
Set the message field of this event.
`setMessage(SOAPMessage)` - Method in interface
javax.xml.rpc.handler.soap.**SOAPMessageContext**
Sets the SOAPMessage in this message context
`setMessage(SOAPMessage)` - Method in interface
javax.xml.ws.handler.soap.**SOAPMessageContext**
Sets the SOAPMessage in this message context
`setMessageBundle(String)` - Method in class
javax.faces.application.**Application**
Set the fully qualified class name of the ResourceBundle to be used for JavaServer Faces messages for this application.
`setMessageDrivenContext(MessageDrivenContext)` - Method in interface
javax.ejb.**MessageDrivenBean**
Set the associated message-driven context.
`setMessageListener(MessageListener)` - Method in interface
javax.jms.**MessageConsumer**
Sets the message consumer's MessageListener.
`setMessageListener(MessageListener)` - Method in interface
javax.jms.**Session**
Sets the session's distinguished message listener (optional).
`setMessageNumber(int)` - Method in class javax.mail.**Message**
Set the Message number for this Message.
`setMetadata(List<Source>)` - Method in class javax.xml.ws.**Endpoint**
Sets the metadata for this endpoint.
setMiddleName(String) - Method in interface
javax.xml.registry.infomodel.PersonName
Sets the middle name for this Person.

setMimeHeader(String, String) - Method in class
javax.xml.soap.AttachmentPart
Changes the first header entry that matches the given name to the given value, adding a new header if no existing header matches.

setMimeHeader(String, String) - Method in class
javax.xml.soap.SOAPPart
Changes the first header entry that matches the given header name so that its value is the given value, adding a new header with the given name and value if no existing header is a match.

setMimeType(String) - Method in interface
javax.xml.registry.infomodel.ExtrinsicObject
Sets the mime type associated with this object.

setMinFractionDigits(int) - Method in class
javax.faces.convert.NumberConverter
Set the minimum number of digits getAsString() should render in the fraction portion of the result.

setMinimum(double) - Method in class
javax.faces.validator.DoubleRangeValidator
Set the minimum value to be enforced by this Validator.

setMinimum(int) - Method in class
javax.faces.validator.LengthValidator
Set the minimum length to be enforced by this Validator.

setMinimum(Long) - Method in class
javax.faces.validator.LongRangeValidator
Set the minimum value to be enforced by this Validator.

setMinIntegerDigits(int) - Method in class
javax.faces.convert.NumberConverter
Set the minimum number of digits getAsString() should render in the integer portion of the result.

setMinorVersion(int) - Method in interface
javax.xml.registry.infomodel.Versionable
Sets the minor revision number for this version of the Versionable object.

setMTOMEnabled(boolean) - Method in interface
javax.xml.ws.soap.SOAPBinding
Enables or disables use of MTOM.

setMultipartDataSource(MultipartDataSource) - Method in class
javax.mail.Multipart
Setup this Multipart object from the given MultipartDataSource.

setMustUnderstand(boolean) - Method in interface
javax.xml.soap.SOAPHeaderElement
Sets the mustUnderstand attribute for this SOAPHeaderElement object to be either true or false.

setName(String) - Method in class javax.faces.component.UIParameter
Set the optional parameter name for this parameter.

setName(String) - Method in class javax.faces.webapp.AttributeTag
**Deprecated.** Set the attribute name.

**setName(String)** - Method in class javax.faces.webapp.FacetTag  
Set the name to be assigned to this facet.

**setName(String)** - Method in class javax.mail.util.ByteArrayDataSource  
Set the name of the data.

**setName(InternationalString)** - Method in interface javax.xml.registry.infomodel.RegistryObject  
Sets user-friendly name of object in repository.

**setName(String)** - Method in interface javax.xml.registry.infomodel.Slot  
Sets the name for this Slot.

**setNamespaceContext(NamespaceContext)** - Method in interface javax.xml.stream.XMLEventWriter  
Sets the current namespace context for prefix and uri bindings.

Sets the current namespace context for prefix and uri bindings.

Set the NavigationHandler instance that will be passed the outcome returned by any invoked application action for this web application.

**setNewsgroup(String)** - Method in class javax.mail.internet.NewsAddress  
Set the newsgroup.

**setNextException(Exception)** - Method in exception javax.mail.MessagingException  
Add an exception to the end of the chain.

**setNil(boolean)** - Method in class javax.xml.bind.JAXBElement  
Set whether this element has nil content.

**setNode(Node)** - Method in class javax.xml.bind.helpers.ValidationEventLocatorImpl  
Set the Node field on this event locator.

**setNoNSSchemaLocation(String)** - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl  
Convenience method for setting the noNamespaceSchemaLocation.

**setNumber(String)** - Method in interface javax.xml.registry.infomodel.TelephoneNumber  
Sets the telephone number suffix, not including the country or area code.

**setNumberFormat(NumberFormat)** - Method in class javax.mail.internet.MailDateFormat  
Don't allow setting the NumberFormat

**setObject(String, Object)** - Method in interface javax.jms.MapMessage  
Sets an object value with the specified name into the Map.

**setObject(Serializable)** - Method in interface javax.jms.ObjectMessage  
Sets the serializable object containing this message's data.
**setObject(Object)** - Method in class
javax.xml.bind.helpers.**ValidationEventLocatorImpl**
Set the Object field on this event locator.

**setObjectProperty(String, Object)** - Method in interface
javax.jms.**Message**
Sets a Java object property value with the specified name into the message.

**setOffset(int)** - Method in class
javax.xml.bind.helpers.**ValidationEventLocatorImpl**
Set the offset field on this event locator.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlCommandButton**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlCommandLink**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlInputSecret**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlInputText**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlInputTextarea**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlOutputLabel**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlOutputLink**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlSelectBooleanCheckbox**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlSelectManyCheckbox**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlSelectManyListbox**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlSelectOneListbox**
Set the value of the onblur property.

**setOnblur(String)** - Method in class
javax.faces.component.html.**HtmlSelectOneMenu**
Set the value of the onblur property.
javax.faces.component.html.HtmlSelectOneRadio
   Set the value of the onblur property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlCommandButton
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlInputElement
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlInputPassword
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlInputText
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlInputTextarea
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectManyListbox
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectManyMenu
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectOneListbox
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectOneMenu
   Set the value of the onchange property.
setOnchange(String) - Method in class
javax.faces.component.html.HtmlSelectOneRadio
   Set the value of the onchange property.
setOnclick(String) - Method in class
javax.faces.component.html.HtmlCommandButton
   Set the value of the onclick property.
setOnclick(String) - Method in class
javax.faces.component.html.HtmlCommandLink
   Set the value of the onclick property.
setOnclick(String) - Method in class
javax.faces.component.html.HtmlDataTable
   Set the value of the onclick property.
setOnclick(String) - Method in class
javax.faces.component.html.HtmlForm
   Set the value of the onclick property.
setOnclick(String) - Method in class
javax.faces.component.html.HtmlGraphicImage
   Set the value of the onclick property.
**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlInputSecret
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlInputText
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlInputTextarea
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlOutputLabel
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlOutputLink
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
  Set the value of the onclick property.

**setOnclick(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
  Set the value of the onclick property.

**setOndblclick(String)** - Method in class
javax.faces.component.html.HtmlCommandButton
  Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class
javax.faces.component.html.HtmlCommandLink
  Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class
javax.faces.component.html.HtmlDataTable
  Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class
javax.faces.component.html.HtmlForm
  Set the value of the ondblclick property.
Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class javax.faces.component.html.HtmlGraphicImage

Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class javax.faces.component.html.HtmlInputSecret

Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class javax.faces.component.html.HtmlInputText

Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class javax.faces.component.html.HtmlInputTextarea

Set the value of the ondblclick property.

**setOndblclick(String)** - Method in class javax.faces.component.html.HtmlInputLabel

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlOutputLink

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlPanelGrid

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectManyListbox

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectManyMenu

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectOneListbox

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectOneMenu

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlSelectOneRadio

Set the value of the ondblclick property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlCommandButton

Set the value of the onfocus property.

**setOnfocus(String)** - Method in class javax.faces.component.html.HtmlCommandLink

Set the value of the onfocus property.
javax.faces.component.html.HtmlInputSecret
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlInputText
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlInputTextarea
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlOutputLabel
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Set the value of the onfocus property.

setOnfocus(String) - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the onfocus property.

setOnkeydown(String) - Method in class
javax.faces.component.html.HtmlCommandButton
Set the value of the onkeydown property.

setOnkeydown(String) - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the onkeydown property.

setOnkeydown(String) - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the onkeydown property.

setOnkeydown(String) - Method in class
javax.faces.component.html.HtmlForm
Set the value of the onkeydown property.

setOnkeydown(String) - Method in class
javax.faces.component.html.HtmlGraphicImage
Set the value of the onkeydown property.
**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlInputSecret
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlInputText
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlInputTextarea
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlOutputLabel
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlOutputLink
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlPanelGrid
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectManyListbox
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectManyMenu
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectOneListbox
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectOneMenu
  Set the value of the onkeydown property.

**setOnkeydown(String)** - Method in class javax.faces.component.html.HtmlSelectOneRadio
  Set the value of the onkeydown property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlCommandButton
  Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlCommandLink
  Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlDataTable
  Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlForm
Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlGraphicImage

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlInputSecret

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlInputText

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlInputTextarea

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlOutputLabel

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlOutputLink

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlPanelGrid

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelect MANYListbox

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelect MANYMenu

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelectOneListbox

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelectOneMenu

Set the value of the onkeypress property.

**setOnkeypress(String)** - Method in class javax.faces.component.html.HtmlSelectOneRadio

Set the value of the onkeypress property.

**setOnkeyup(String)** - Method in class javax.faces.component.html.HtmlCommandButton

Set the value of the onkeyup property.

**setOnkeyup(String)** - Method in class javax.faces.component.html.HtmlCommandLink

Set the value of the onkeyup property.
javax.faces.component.html.HtdmlDataTable
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlForm
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlGraphicImage
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlInputSecret
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlInputText
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlInputTextarea
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlOutputLabel
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlOutputLink
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlPanelGrid
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectBooleanCheckbox
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectManyCheckbox
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectManyListbox
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectManyMenu
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectOneListbox
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectOneMenu
  Set the value of the onkeyup property.
**setOnkeyup(String)** - Method in class
javax.faces.component.html.HtdmlSelectOneRadio
  Set the value of the onkeyup property.
**setOnmousedown(String)** - Method in class
javax.faces.component.html.HtdmlCommandButton
  Set the value of the onmousedown property.
`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlCommandLink`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlDataTable`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlForm`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlGraphicImage`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlInputSecret`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlInputText`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlInputTextarea`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlOutputLabel`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlOutputLink`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlPanelGrid`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
Set the value of the onmousedown property.

`setOnmousedown(String)` - Method in class `javax.faces.component.html.HtmlSelectOneRadio`
Set the value of the onmousedown property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlCommandButton`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlCommandLink`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlDataTable`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlForm`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlGraphicImage`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlInputSecret`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlInputText`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlInputTextarea`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlOutputLabel`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlOutputLink`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlPanelGrid`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Set the value of the onmousemove property.

`setOnmousemove(String)` - Method in class
Set the value of the onmousemove property.

**setOnmousemove(String)** - Method in class

- javax.faces.component.html.HtmlSelectOneMenu

Set the value of the onmousemove property.

**setOnmousemove(String)** - Method in class

- javax.faces.component.html.HtmlSelectOneRadio

Set the value of the onmousemove property.

**setOnmousemove(String)** - Method in class

- javax.faces.component.html.HtmlCommandButton

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlCommandLink

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlDataTable

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlForm

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlGraphicImage

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlInputSecret

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlInputText

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlInputTextarea

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlOutputLabel

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlOutputLink

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlPanelGrid

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlSelectBooleanCheckbox

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlSelectManyCheckbox

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlSelectManyListbox

Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class

- javax.faces.component.html.HtmlSelectManyMenu

Set the value of the onmouseout property.
**setOnmouseout(String)** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
  Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class `javax.faces.component.html.HtmlSelectOneMenu`
  Set the value of the onmouseout property.

**setOnmouseout(String)** - Method in class `javax.faces.component.html.HtmlSelectOneRadio`
  Set the value of the onmouseout property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlCommandButton`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlCommandLink`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlDataTable`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlForm`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlGraphicImage`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlInputSecret`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlInputText`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlInputTextarea`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlOutputLabel`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlOutputLink`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlPanelGrid`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`
  Set the value of the onmouseover property.

**setOnmouseover(String)** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
  Set the value of the onmouseover property.
Set the value of the onmouseover property.
\texttt{setOnmouseover(String)} - Method in class
\texttt{javax.faces.component.html.HtmlSelectManyMenu}

Set the value of the onmouseover property.
\texttt{setOnmouseover(String)} - Method in class
\texttt{javax.faces.component.html.HtmlSelectOneListbox}

Set the value of the onmouseover property.
\texttt{setOnmouseover(String)} - Method in class
\texttt{javax.faces.component.html.HtmlSelectOneMenu}

Set the value of the onmouseover property.
\texttt{setOnmouseover(String)} - Method in class
\texttt{javax.faces.component.html.HtmlSelectOneRadio}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlCommandButton}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlCommandLink}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlDataTable}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlForm}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlGraphicImage}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlInputSecret}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlInputText}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlInputTextarea}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlOutputLabel}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlOutputLink}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlPanelGrid}

Set the value of the onmouseup property.
\texttt{setOnmouseup(String)} - Method in class
\texttt{javax.faces.component.html.HtmlSelectBooleanCheckbox}
Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`

Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`

Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`

Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`

Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`

Set the value of the onmouseup property.

`setOnmouseup(String)` - Method in class `javax.faces.component.html.HtmlSelectOneRadio`

Set the value of the onmouseup property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlInputSecret`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlInputText`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlInputTextarea`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectManyCheckbox`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectManyListbox`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectManyMenu`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectOneListbox`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`

Set the value of the onreset property.

`setOnreset(String)` - Method in class `javax.faces.component.html.HtmlSelectOneMenu`

Set the value of the onreset property.
**setOnselect(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the onselect property.

**setOnsubmit(String)** - Method in class
javax.faces.component.html.HtmlForm
Set the value of the onsubmit property.

**setOpaque(boolean)** - Method in interface
javax.xml.registry.infomodel.ExtrinsicObject
Sets whether the ExtrinsicObject is opaque (not readable) by the registry.

**setOperationName(QName)** - Method in interface javax.xml.rpc.Call
Sets the name of the operation to be invoked using this Call instance.

**setPageContext(PageContext)** - Method in class
javax.faces.webapp.UICOMPONENTCLASSICTAGBASE
Set the PageContext of the page containing this tag instance.

**setPageContext(PageContext)** - Method in interface
javax.servlet.jsp.tagext.Tag
Set the current page context.

**setPageContext(PageContext)** - Method in class
javax.servlet.jsp.tagext.TagAdapter
Must not be called.

**setPageContext(PageContext)** - Method in class
javax.servlet.jsp.tagext.TagSupport
Set the page context.

**setParameter(String, String)** - Method in class
javax.activation.MimeType
Set the value to be associated with the given name, replacing any previous association.

**setParameter(String, String)** - Method in class
javax.mail.internet.ContentDisposition
Set the specified parameter.

**setParameter(String, String)** - Method in class
javax.mail.internet.ContentType
Set the specified parameter.

**setParameter(String, Object)** - Method in interface
javax.persistence.Query
Bind an argument to a named parameter.

**setParameter(String, Date, TemporalType)** - Method in interface
javax.persistence.Query
Bind an instance of java.util.Date to a named parameter.

**setParameter(String, Calendar, TemporalType)** - Method in interface
javax.persistence.Query
Bind an instance of java.util.Calendar to a named parameter.

**setParameter(int, Object)** - Method in interface
javax.persistence.Query
Bind an argument to a positional parameter.

**setParameter(int, Date, TemporalType)** - Method in interface
javax.persistence.Query
Bind an instance of java.util.Date to a positional parameter.
setParameter(int, Calendar, TemporalType) - Method in interface
javax.persistence.Query
    Bind an instance of java.util.Calendar to a positional parameter.
setParameterList(ParameterList) - Method in class
javax.mail.internet.ContentDisposition
    Set a new ParameterList.
setParameterList(ParameterList) - Method in class
javax.mail.internet.ContentType
    Set a new ParameterList.
setParameter(Object[]) - Method in interface
javax.interceptor.InvocationContext
    Sets the parameters that will be used to invoke the business method.
setParameter(UIComponent) - Method in class
javax.faces.component.UIComponent
    Set the parent UIComponent of this UIComponent.
setParameter(UIComponent) - Method in class
javax.faces.component.UIComponentBase
setParameter(Tag) - Method in class
javax.faces.webapp.UIComponentClassicTagBase
    Set the Tag that is the parent of this instance.
setParameter(Part) - Method in class javax.mail.Multipart
    Set the parent of this Multipart to be the specified Part.
setParameter(JspTag) - Method in interface
javax.servlet.jsp.tagext.SimpleTag
    Sets the parent of this tag, for collaboration purposes.
setParameter(JspTag) - Method in class
javax.servlet.jsp.tagext.SimpleTagSupport
    Sets the parent of this tag, for collaboration purposes.
setParameter(Tag) - Method in interface javax.servlet.jsp.tagext.Tag
    Set the parent (closest enclosing tag handler) of this tag handler.
setParameter(Tag) - Method in class
javax.servlet.jsp.tagext.TagAdapter
    Must not be called.
setParameter(Tag) - Method in class
javax.servlet.jsp.tagext.TagSupport
    Set the nesting tag of this tag.
setParameter(XMLEventReader) - Method in class
javax.xml.stream.util.EventReaderDelegate
    Set the parent of this instance.
setParameter(XMLStreamReader) - Method in class
javax.xml.stream.util.StreamReaderDelegate
    Set the parent of this instance.
setParameterElement(SOAPElement) - Method in interface
javax.xml.soap.Node
    Sets the parent of this Node object to the given SOAPElement object.
**setPasswordAuthentication(URLName, PasswordAuthentication)** - Method in class `javax.mail.Session`  
Save a PasswordAuthentication for this (store or transport) URLName.

**setPath(String)** - Method in class `javax.servlet.http.Cookie`  
Specifies a path for the cookie to which the client should return the cookie.

**setPattern(String)** - Method in class `javax.faces.convert.DateTimeConverter`  
Set the format pattern to be used when formatting and parsing dates and times.

**setPattern(String)** - Method in class `javax.faces.convert.NumberConverter`  
Set the format pattern to be used when formatting and parsing numbers.

**setPayload(Source)** - Method in interface `javax.xml.ws.LogicalMessage`  
Sets the message payload.

**setPayload(Object, JAXBContext)** - Method in interface `javax.xml.ws.LogicalMessage`  
Sets the message payload.

**setPersonal(String, String)** - Method in class `javax.mail.internet.InternetAddress`  
Set the personal name.

**setPersonal(String)** - Method in class `javax.mail.internet.InternetAddress`  
Set the personal name.

**setPersonName(PersonName)** - Method in interface `javax.xml.registry.infomodel.User`  
Sets the name of this User.

**setPhaseId(PhaseId)** - Method in class `javax.faces.event.FacesEvent`  
Set the **PhaseId** during which this event will be delivered.

**setPortTypeName(QName)** - Method in interface `javax.xml.rpc.Call`  
Sets the qualified name of the port type.

**setPostalAddress(PostalAddress)** - Method in interface `javax.xml.registry.infomodel.Organization`  
Sets the address for this Organization.

**setPostalAddresses(Collection)** - Method in interface `javax.xml.registry.infomodel.User`  
Sets the addresses for this User.

**setPostalCode(String)** - Method in interface `javax.xml.registry.infomodel.PostalAddress`  
Sets the postal or zip code.

**setPostalScheme(ClassificationScheme)** - Method in interface `javax.xml.registry.infomodel.PostalAddress`  
Sets a user-defined postal scheme for codifying the attributes of PostalAddress.

**setPreamble(String)** - Method in class `javax.mail.internet.MimeMultipart`  
Set the preamble text to be included before the first body
part.

**setPrefix(String, String)** - Method in interface javax.xml.stream.XMLEventWriter
Sets the prefix the uri is bound to.

**setPrefix(String, String)** - Method in interface javax.xml.stream.XMLStreamWriter
Sets the prefix the uri is bound to.

**setPrependId(boolean)** - Method in class javax.faces.component.UIForm

**setPrimaryContact(User)** - Method in interface javax.xml.registry.infomodel.Organization
Sets the primary contact for this Organization.

**setPrimaryType(String)** - Method in class javax.activation.MimeType
Set the primary type for this object to the given String.

**setPrimaryType(String)** - Method in class javax.mail.internet.ContentType
Set the primary type.

**setPriority(int)** - Method in interface javax.jms.MessageProducer
Sets the producer's default priority.

**setProperties(UIComponent)** - Method in class javax.faces.webapp.UIComponentClassicTagBase
Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set.

**setProperties(UIComponent)** - Method in class javax.faces.webapp.UIComponentELTag
Override properties and attributes of the specified component, if the corresponding properties of this tag handler instance were explicitly set.

**setProperties(UIComponent)** - Method in class javax.faces.webapp.UIComponentTag
Deprecated.

**setProperties(Properties)** - Method in class javax.xml.registry.infomodel.ConnectionFactory
Sets the Properties used during createConnection and createFederatedConnection calls.

**setProperties(Map<String, Object>)** - Method in class javax.xml.ws.Endpoint
Sets the property bag for this Endpoint instance.

**setProperty(String, Object)** - Method in class javax.xml.bind.Binder
Set the particular property in the underlying implementation of Binder.

**setProperty(String, Object)** - Method in class javax.xml.bind.helpers.AbstractMarshallerImpl
Default implementation of the setProperty method handles the four defined properties in Marshaller.

**setProperty(String, Object)** - Method in class javax.xml.bind.helpers.AbstractUnmarshallerImpl
Default implementation of the setProperty method always throws
setProperty(String, Object) - Method in interface javax.xml.bind.Marshaller
Set the particular property in the underlying implementation of Marshaller.

setProperty(String, Object) - Method in interface javax.xml.bind.Unmarshaller
Set the particular property in the underlying implementation of Unmarshaller.

setProperty(String, Object) - Method in interface javax.xml.bind.databind.Secret

setProperty(String, Object) - Method in interface javax.xml.bind.Valideator
Deprecated. since JAXB2.0

setProperty(String, Object) - Method in interface javax.xml.rpc.Call
Sets the value for a named property.

setProperty(String, Object) - Method in interface javax.xml.rpc.handler.MessageContext
Sets the name and value of a property associated with the MessageContext.

setProperty(String, Object) - Method in class javax.xml.soap.SOAPMessage
Associates the specified value with the specified property.

setProperty(String, Object) - Method in class javax.xml.stream.XMLInputFactory
Allows the user to set specific feature/property on the underlying implementation.

setProperty(String, Object) - Method in class javax.xml.stream.XMLOutputFactory
Allows the user to set specific features/properties on the underlying implementation.

setPropertyResolved(boolean) - Method in class javax.el.ELContext
Called to indicate that a ELResolver has successfully resolved a given (base, property) pair.

setPropertyResolver(PropertyResolver) - Method in class javax.faces.application.Application
Deprecated. The recommended way to affect the execution of the EL is to provide an <el-resolver> element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. This method now will cause the argument resolver to be wrapped inside an implementation of ELResolver and exposed to the EL resolution system as if the user had called Application.addELResolver(javax.el.ELResolver).

setProtocolForAddress(String, String) - Method in class javax.mail.Session
Set the default transport protocol to use for addresses of the specified type.

setProvider(Provider) - Method in class javax.mail.Session
Set the passed Provider to be the default implementation for the protocol in Provider.protocol overriding any previous
values.

**setProvidingOrganization(Organization)** - Method in interface javax.xml.registry.infomodel.Service
   Sets the Organization that provides this service.

**setQuota(Quota)** - Method in interface javax.mail.QuotaAwareStore
   Set the quotas for the quota root specified in the quota argument.

**setRawContent(InputStream, String)** - Method in class javax.xml.soap.AttachmentPart
   Sets the content of this attachment part to that contained by the InputStream content and sets the value of the Content-Type header to the value contained in contentType.

**setRawContentBytes(byte[], int, int, String)** - Method in class javax.xml.soap.AttachmentPart
   Sets the content of this attachment part to that contained by the byte[] array content and sets the value of the Content-Type header to the value contained in contentType.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlCommandButton
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlInputSecret
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlInputText
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlInputTextarea
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectManyCheckbox
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectManyListbox
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectManyMenu
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectOneListbox
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectOneMenu
   Set the value of the readonly property.

**setReadonly(boolean)** - Method in class javax.faces.component.html.HtmlSelectOneRadio
   Set the value of the readonly property.
**setRecipient(Message.RecipientType, Address)** - Method in class `javax.mail.Message`
Set the recipient address.

**setRecipients(Message.RecipientType, Address[])** - Method in class `javax.mail.internet.MimeMessage`
Set the specified recipient type to the given addresses.

**setRecipients(Message.RecipientType, String)** - Method in class `javax.mail.internet.MimeMessage`
Set the specified recipient type to the given addresses.

**setRecipients(Message.RecipientType, Address[])** - Method in class `javax.mail.Message`
Set the recipient addresses.

**setRecordName(String)** - Method in interface `javax.resource.cci.Record`
Sets the name of the Record.

**setRecordShortDescription(String)** - Method in interface `javax.resource.cci.Record`
Sets a short description string for the Record.

**setRedisplay(boolean)** - Method in class `javax.faces.component.html.HtmlInputSecret`
Set the value of the redisplay property.

**setReference(Reference)** - Method in interface `javax.resource.Referenceable`
Sets the Reference instance.

**setRel(String)** - Method in class `javax.faces.component.html.HtmlCommandLink`
Set the value of the rel property.

**setRel(String)** - Method in class `javax.faces.component.html.HtmlOutputLink`
Set the value of the rel property.

**setRelay(boolean)** - Method in interface `javax.xml.soap.SOAPHeaderElement`
Sets the relay attribute for this SOAPHeaderElement to be either true or false.

**setRendered(boolean)** - Method in class `javax.faces.component.UIComponent`
Set the rendered property of this `UIComponent`.

**setRendered(boolean)** - Method in class `javax.faces.component.UIComponentBase`
Set an override for the rendered property.

**setRendererType(String)** - Method in class `javax.faces.component.UIComponent`
Set the `Renderer` type for this `UIComponent`, or null for components that render themselves.
**setRendererType(String)** - Method in class
javax.faces.component.UIComponentBase

**setRenderKitId(String)** - Method in class
javax.faces.componentUIViewRoot

Set the render kit identifier of the RenderKit associated with this view.

**setReplyTo(Address[])** - Method in class
javax.mail.internet.MimeMessage

Set the RFC 822 "Reply-To" header field.

**setReplyTo(Address[])** - Method in class javax.mail.Message

Set the addresses to which replies should be directed.

**setRepositoryItem(DataHandler)** - Method in interface
javax.xml.registry.infomodel.ExtrinsicObject

Sets the repository item for this object.

**setRequest(Object)** - Method in class
javax.faces.context.ExternalContext

Set the environment-specific request to be returned by subsequent calls to ExternalContext.getRequest().

**setRequest(ServletRequest)** - Method in class
javax.servlet.HttpServletRequestWrapper

Sets the request object being wrapped.

**setRequestCharacterEncoding(String)** - Method in class
javax.faces.context.ExternalContext

Overrides the name of the character encoding used in the body of this request.

**setRequired(boolean)** - Method in interface
javax.faces.component.EditableValueHolder

Set the "required field" state for this component.

**setRequired(boolean)** - Method in class
javax.faces.component.UIInput

Set the "required field" state for this component.

**setRequiredMessage(String)** - Method in class
javax.faces.component.UIInput

Override any ValueExpression set for the "requiredMessage" with the literal argument provided to this method.

**setResourceAdapter(ResourceAdapter)** - Method in interface
javax.resource.spi.ResourceAdapterAssociation

Associate this object with a ResourceAdapter object.

**setResourceLimit(String, long)** - Method in class javax.mail.Quota

Set a resource limit for this quota root.

**setResponse(Object)** - Method in class
javax.faces.context.ExternalContext

Set the environment-specific response to be returned by subsequent calls to ExternalContext.getResponse().

**setResponse(ServletResponse)** - Method in class
javax.servlet.HttpServletResponseWrapper

Sets the response being wrapped.

**setResponseCharacterEncoding(String)** - Method in class
javax.faces.context.ExternalContext
Sets the character encoding (MIME charset) of the response being sent to the client, for example, to UTF-8.


Set the `ResponseStream` to which components should direct their binary output.


Set the `ResponseWriter` to which components should direct their character-based output.

**setReturnType(QName)** - Method in interface `javax.xml.rpc.Call`

Sets the return type for a specific operation.

**setReturnType(QName, Class)** - Method in interface `javax.xml.rpc.Call`

Sets the return type for a specific operation.

**setRev(String)** - Method in class `javax.faces.component.html.HtmlCommandLink`

Set the value of the `rev` property.

**setRev(String)** - Method in class `javax.faces.component.html.HtmlOutputLink`

Set the value of the `rev` property.

**setRole(String)** - Method in interface `javax.xml.soap.SOAPHeaderElement`

Sets the Role associated with this SOAPHeaderElement object to the specified Role.

**setRoles(String[])** - Method in interface `javax.xml.rpc.handler.HandlerChain`

Sets SOAP Actor roles for this HandlerChain.

**setRoles(Set<String>)** - Method in interface `javax.xml.ws.soap.SOAPBinding`

Sets the roles played by the SOAP binding instance.

**setRollbackOnly()** - Method in interface `javax.ejb.EJBContext`

Mark the current transaction for rollback.

**setRollbackOnly()** - Method in interface `javax.persistence.EntityTransaction`

Mark the current transaction so that the only possible outcome of the transaction is for the transaction to be rolled back.

**setRollbackOnly()** - Method in interface `javax.transaction.Transaction`

Modify the transaction associated with the target object such that the only possible outcome of the transaction is to roll back the transaction.

**setRollbackOnly()** - Method in interface `javax.transaction.TransactionManager`

Modify the transaction associated with the current thread such that the only possible outcome of the transaction is to roll back the transaction.

**setRollbackOnly()** - Method in interface `javax.transaction.TransactionSynchronizationRegistry`

Set the rollbackOnly status of the transaction bound to the...
current thread at the time this method is called.

**setRollbackOnly()** - Method in interface
javax.transaction.UserTransaction
Modify the transaction associated with the current thread such that the only possible outcome of the transaction is to rollback the transaction.

**setRowClasses(String)** - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the rowClasses property.

**setRowClasses(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the rowClasses property.

**setRowIndex(int)** - Method in class
javax.faces.model.UIData
Set the zero relative index of the current row, or -1 to indicate that no row is currently selected, by implementing the following algorithm.

**setRowIndex(int)** - Method in class
javax.faces.model.ArrayDataModel
Set the zero-relative index of the currently selected row, or -1 to indicate that we are not positioned on a row.

**setRowIndex(int)** - Method in class
javax.faces.model.ListDataModel

**setRowIndex(int)** - Method in class
javax.faces.model.ResultSetDataModel

**setRowIndex(int)** - Method in class
javax.faces.model.ScalarDataModel

**setRows(int)** - Method in class
javax.faces.component.html.HtmlInputTextarea
Set the value of the rows property.

**setRows(int)** - Method in class
javax.faces.component.UIData
Set the number of rows to be displayed, or zero for all remaining rows in the table.

**setRules(String)** - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the rules property.

**setRules(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the rules property.

**setSchema(Schema)** - Method in class
javax.xml.bind.Binder
Specifies whether marshal, unmarshal and update methods performs validation on their XML content.

**setSchema(Schema)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl
**setSchema(Schema)** - Method in class
javax.xml.bind.helpers.AbstractUnmarshallerImpl

**setSchema(Schema)** - Method in interface javax.xml.bind.Marshaller
Specify the JAXP 1.3 Schema object that should be used to validate subsequent marshal operations against.

**setSchema(Schema)** - Method in interface javax.xml.bind.Unmarshaller
Specify the JAXP 1.3 Schema object that should be used to validate subsequent unmarshal operations against.

**setSchemaLocation(String)** - Method in class
javax.xml.bind.helpers.AbstractMarshallerImpl
Convenience method for setting the schemaLocation.

**setScope(String, MessageContext.Scope)** - Method in interface
javax.xml.ws.handler.MessageContext
Sets the scope of a property.

Indicates to the browser whether the cookie should only be sent using a secure protocol, such as HTTPS or SSL.

**setSelected(boolean)** - Method in class
javax.faces.component.UISelectBoolean
Set the local value of the selected state of this component.

**setSelectedValues(Object[])** - Method in class
javax.faces.component.UISelectMany
Set the currently selected values, or null to indicate that there are no currently selected values.

**setSelectItems(SelectItem[])** - Method in class
javax.faces.model.SelectItemGroup
Set the set of subordinate SelectItems for this group.

**setSender(Address)** - Method in class
javax.mail.internet.MimeMessage
Set the RFC 822 "Sender" header field.

**setSentDate(Date)** - Method in class javax.mail.internet.MimeMessage
Set the RFC 822 "Date" header field.

**setSentDate(Date)** - Method in class javax.mail.Message
Set the sent date of this message.

**setSessionContext(SessionContext)** - Method in interface
javax.ejb.SessionBean
Set the associated session context.

**setSeverity(FacesMessage.Severity)** - Method in class
javax.faces.application.FacesMessage
Set the severity level.

**setSeverity(int)** - Method in class
javax.xml.bind.helpers.ValidationEventImpl
Set the severity field of this event.

**setShape(String)** - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the shape property.

**setShape(String)** - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the shape property.
**setShort(String, short)** - Method in interface `javax.jms.MapMessage`
Sets a short value with the specified name into the Map.

**setShortProperty(String, short)** - Method in interface `javax.jms.Message`
Sets a short property value with the specified name into the message.

**setShowDetail(boolean)** - Method in class `javax.faces.component/UIMessage`
Set the flag indicating whether the detail property of the associated message(s) should be displayed.

**setShowSummary(boolean)** - Method in class `javax.faces.component/UIMessage`
Set the flag indicating whether the summary property of the associated message(s) should be displayed.

**setSize(int)** - Method in class `javax.faces.component.html.HtmlInputSecret`
Set the value of the size property.

**setSize(int)** - Method in class `javax.faces.component.html.HtmlInputText`
Set the value of the size property.

**setSize(int)** - Method in class `javax.faces.component.html.HtmlSelectManyListbox`
Set the value of the size property.

**setSize(int)** - Method in class `javax.faces.component.html.HtmlSelectOneListbox`
Set the value of the size property.

**setSlotType(String)** - Method in interface `javax.xml.registry.infomodel.Slot`
Sets the slotType for this Slot.

**setSourceObject(RegistryObject)** - Method in interface `javax.xml.registry.infomodel.Association`
Sets the Object that is the source of this Association.

Sets the specification object for this object.

**setStability(int)** - Method in interface `javax.xml.registry.infomodel.RegistryEntry`
Sets the stability indicator for the RegistryEntry.

**setStateManager(StateManager)** - Method in class `javax.faces.application.Application`
Set the StateManager instance that will be utilized during the Restore View and Render Response phases of the request.
processing lifecycle.

**setStateOrProvince(String)** - Method in interface
javax.xml.registry.infomodel.PostalAddress
Sets the state or province.

**setStatus(int)** - Method in interface
javax.servlet.http.HttpServletResponse
Sets the status code for this response.

**setStatus(int, String)** - Method in interface
javax.servlet.http.HttpServletResponse
**Deprecated.** As of version 2.1, due to ambiguous meaning of the message parameter. To set a status code use setStatus(int), to send an error with a description use sendError(int, String).
Sets the status code and message for this response.

**setStatus(int)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to call setStatus(int sc) on the wrapped response object.

**setStatus(int, String)** - Method in class
javax.servlet.http.HttpServletResponseWrapper
The default behavior of this method is to call setStatus(int sc, String sm) on the wrapped response object.

**setStreet(String)** - Method in interface
javax.xml.registry.infomodel.PostalAddress
Sets the street name.

**setStreetNumber(String)** - Method in interface
javax.xml.registry.infomodel.PostalAddress
Sets the street number.

**setString(String, String)** - Method in interface
javax.jms.MapMessage
Sets a String value with the specified name into the Map.

**setStringProperty(String, String)** - Method in interface
javax.jms.Message
Sets a String property value with the specified name into the message.

**setStyle(String)** - Method in class
javax.faces.component.html.HtmlCommandButton
Set the value of the style property.

**setStyle(String)** - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the style property.

**setStyle(String)** - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the style property.

**setStyle(String)** - Method in class
javax.faces.component.html.HtmlForm
Set the value of the style property.

**setStyle(String)** - Method in class
javax.faces.component.html.HtmlGraphicImage
Set the value of the style property.
javax.faces.component.html.HtmlInputSecret
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlInputSecret

javax.faces.component.html.HtmlInputText
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlInputText

javax.faces.component.html.HtmlInputTextarea
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlInputTextarea

javax.faces.component.html.HtmlMessage
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlMessage

javax.faces.component.html.HtmlMessages
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlMessages

javax.faces.component.html.HtmlOutputFormat
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlOutputFormat

javax.faces.component.html.HtmlOutputLabel
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlOutputLabel

javax.faces.component.html.HtmlOutputLink
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlOutputLink

javax.faces.component.html.HtmlOutputText
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlOutputText

javax.faces.component.html.HtmlPanelGrid
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlPanelGrid

javax.faces.component.html.HtmlPanelGroup
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlPanelGroup

javax.faces.component.html.HtmlSelectBooleanCheckbox
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectBooleanCheckbox

javax.faces.component.html.HtmlSelectManyCheckbox
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectManyCheckbox

javax.faces.component.html.HtmlSelectManyListbox
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectManyListbox

javax.faces.component.html.HtmlSelectManyMenu
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectManyMenu

javax.faces.component.html.HtmlSelectOneListbox
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectOneListbox

javax.faces.component.html.HtmlSelectOneMenu
   Set the value of the style property.
setStyle(String) - Method in class javax.faces.component.html.HtmlSelectOneMenu
**setStyle(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the style property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlCommandButton
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlCommandLink
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlDataTable
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlForm
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlGraphicImage
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlInputSecret
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlInputText
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlInputTextarea
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlMessage
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlMessages
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlOutputFormat
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlOutputLabel
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlOutputLink
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlOutputText
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlPanelGrid
Set the value of the styleClass property.

**setStyleClass(String)** - Method in class
javax.faces.component.html.HtmlPanelGroup
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectBooleanCheckbox
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectManyListbox
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectManyMenu
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectOneListbox
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectOneMenu
```
Set the value of the styleClass property.

```
setStyleClass(String) - Method in class
javax.faces.component.html.HtmlSelectOneRadio
```
Set the value of the styleClass property.

```
setSubject(String) - Method in class
javax.mail.internet.MimeMessage
```
Set the "Subject" header field.

```
setSubject(String, String) - Method in class
javax.mail.internet.MimeMessage
```
Set the "Subject" header field.

```
setSubject(String) - Method in class
javax.mail.Message
```
Set the subject of this message.

```
setSubmitted(boolean) - Method in class
javax.faces.component.UIForm
```
If this UIForm instance (as opposed to other forms in the page) is experiencing a submit during this request processing lifecycle, this method must be called, with true as the argument, during the

```
UIComponent.decode(javax.faces.context.FacesContext) for this UIForm instance.
```

```
setSubmittedValue(Object) - Method in interface
javax.faces.component.EditableValueHolder
```
Set the submittedValue value of this component.

```
setSubmittedValue(Object) - Method in class
javax.faces.component.UIInput
```
Set the submittedValue value of this UIInput component.

```
setSubscribed(boolean) - Method in class
javax.mail.Folder
```
Subscribe or unsubscribe this Folder.

```
setSubType(String) - Method in class
javax.activationMimeType
```
Set the subtype for this object to the given String.

```
setSubType(String) - Method in class
javax.mail.internet.ContentType
```
Set the subType.

**setSubType(String)** - Method in class
javax.mail.internet.MimeMultipart
Set the subtype.

**setSummary(String)** - Method in class
javax.faces.application.FacesMessage
Set the localized summary text.

**setSummary(String)** - Method in class
javax.faces.component.html.DataTable
Set the value of the summary property.

**setSummary(String)** - Method in class
javax.faces.component.html.DataTable
Set the value of the summary property.

**setSupportedEncodings(String[])** - Method in interface
javax.xml.rpc.encoding.TypeMapping
Sets the encodingStyle URIs supported by this TypeMapping instance.

**setSupportedLocales(Collection<Locale>)** - Method in class
javax.faces.application.Application
Set the Locale instances representing the supported Locales for this application.

**setSynchronous(boolean)** - Method in interface
javax.xml.registry.Connection
Sets whether the client uses synchronous communication or not.

**setTabindex(String)** - Method in class
javax.faces.component.html.CommandButton
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.CommandLink
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.InputSecret
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.InputText
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.InputTextarea
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.OutputLabel
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.OutputLink
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.SelectBooleanCheckbox
Set the value of the tabindex property.

**setTabindex(String)** - Method in class
javax.faces.component.html.SelectManyCheckbox
Set the value of the tabindex property.

```
setTabindex(String) - Method in class
javax.faces.component.html.HtmlSelectManyListbox
```
Set the value of the tabindex property.
```
setTabindex(String) - Method in class
javax.faces.component.html.HtmlSelectManyMenu
```
Set the value of the tabindex property.
```
setTabindex(String) - Method in class
javax.faces.component.html.HtmlSelectOneListbox
```
Set the value of the tabindex property.
```
setTabindex(String) - Method in class
javax.faces.component.html.HtmlSelectOneMenu
```
Set the value of the tabindex property.
```
setTabindex(String) - Method in class
javax.faces.component.html.HtmlSelectOneRadio
```
Set the value of the tabindex property.
```
setTagExtraInfo(TagExtraInfo) - Method in class
javax.servlet.jsp.tagext.TagInfo
```
Set the instance for extra tag information.
```
setTagInfo(TagInfo) - Method in class
javax.servlet.jsp.tagext.TagExtraInfo
```
Set the TagInfo for this class.
```
setTagLibrary(TagLibraryInfo) - Method in class
javax.servlet.jsp.tagext.TagInfo
```
Set the TagLibraryInfo property.
```
setTarget(String) - Method in class
javax.faces.component.html.HtmlCommandLink
```
Set the value of the target property.
```
setTarget(String) - Method in class
javax.faces.component.html.HtmlForm
```
Set the value of the target property.
```
setTarget(String) - Method in class
javax.faces.component.html.HtmlOutputLink
```
Set the value of the target property.
```
setTargetBinding(ServiceBinding) - Method in interface
javax.xml.registry.infomodel.ServiceBinding
```
Sets the next ServiceBinding in case there is a redirection.
```
setTargetEndpointAddress(String) - Method in interface
javax.xml.rpc.Call
```
Sets the address of the target service endpoint.
```
setTargetObject(RegistryObject) - Method in interface
javax.xml.registry.infomodel.Association
```
Sets the Object that is the target of this Association.
```
setTelephoneNumbers(Collection) - Method in interface
javax.xml.registry.infomodel.Organization
```
Sets the various telephone numbers for this user.
```
setTelephoneNumbers(Collection) - Method in interface
javax.xml.registry.infomodel.User
```
Sets the various telephone numbers for this user.
```
setText(String) - Method in interface java.util.TextMessage
```

Sets the string containing this message's data.

**setText(String)** - Method in class javax.mail.internet.MimeBodyPart
Convenience method that sets the given String as this part's content, with a MIME type of "text/plain".

**setText(String, String)** - Method in class javax.mail.internet.MimeBodyPart
Convenience method that sets the given String as this part's content, with a MIME type of "text/plain" and the specified charset.

**setText(String, String, String)** - Method in class javax.mail.internet.MimeBodyPart
Convenience method that sets the given String as this part's content, with a primary MIME type of "text" and the specified MIME subtype.

**setText(String)** - Method in class javax.mail.internet.MimeMessage
Convenience method that sets the given String as this part's content, with a MIME type of "text/plain".

**setText(String, String)** - Method in class javax.mail.internet.MimeMessage
Convenience method that sets the given String as this part's content, with a MIME type of "text/plain" and the specified charset.

**setText(String, String, String)** - Method in class javax.mail.internet.MimeMessage
Convenience method that sets the given String as this part's content, with a primary MIME type of "text" and the specified MIME subtype.

**setText(String)** - Method in class javax.mail.internet.MimeMessage
Convenience method that sets the given String as this part's content, with a MIME type of "text/plain".

**setText(String, String, String)** - Method in class javax.mail.internet.MimeMessage
Convenience method that sets the given String as this part's content, with a primary MIME type of "text" and the specified MIME subtype.

**setText(String)** - Method in class javax.mail.Part
A convenience method that sets the given String as this part's content with a MIME type of "text/plain".

**setTimeStyle(String)** - Method in class javax.faces.convert.DateTimeConverter
Set the style to be used to format or parse times.

**setTimeToLive(long)** - Method in interface javax.jms.MessageProducer
Sets the default length of time in milliseconds from its dispatch time that a produced message should be retained by the message system.
`setTimeZone(TimeZone)` - Method in class `javax.faces.convert.DateTimeConverter`  
Set the TimeZone used to interpret a time value.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlCommandButton`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlCommandLink`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlDataTable`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlForm`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlGraphicImage`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlInputSecret`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlInputText`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlInputTextarea`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlMessage`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlMessages`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlOutputFormat`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlOutputLabel`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlOutputLink`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlOutputText`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlPanelGrid`  
Set the value of the title property.

`setTitle(String)` - Method in class `javax.faces.component.html.HtmlSelectBooleanCheckbox`  
Set the value of the title property.
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectManyCheckbox
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectManyListbox
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectManyMenu
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectOneListbox
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectOneMenu
Set the value of the title property.

**setTitle(String)** - Method in class
javax.faces.component.html.HtmlSelectOneRadio
Set the value of the title property.

**setTooltip(boolean)** - Method in class
javax.faces.component.html.HtmlMessage
Set the value of the tooltip property.

**setTooltip(boolean)** - Method in class
javax.faces.component.html.HtmlMessages
Set the value of the tooltip property.

**setTransactionTimeout(long)** - Method in class
javax.resource.spi.work.ExecutionContext
Set the transaction timeout value for a imported transaction.

**setTransactionTimeout(int)** - Method in interface
javax.transaction.TransactionManager
Modify the timeout value that is associated with transactions started by the current thread with the begin method.

**setTransactionTimeout(int)** - Method in interface
javax.transaction.UserTransaction
Modify the timeout value that is associated with transactions started by the current thread with the begin method.

**setTransactionTimeout(int)** - Method in interface
javax.transaction.xa.XAResource
Sets the current transaction timeout value for this XAResource instance.

**setTransient(boolean)** - Method in interface
javax.faces.component.StateHolder
Denotes whether or not the Object implementing this interface must or must not participate in state saving or restoring.

**setTransient(boolean)** - Method in class
javax.faces.component.UIComponentBase

**setTransient(boolean)** - Method in class
javax.faces.convert.DateTimeConverter
**setTransient(boolean)** - Method in class
javax.faces.convert.**EnumConverter**

**setTransient(boolean)** - Method in class
javax.faces.convert.**NumberConverter**

**setTransient(boolean)** - Method in class
javax.faces.event.**MethodExpressionActionListener**

**setTransient(boolean)** - Method in class
javax.faces.event.**MethodExpressionValueChangeListener**

**setTransient(boolean)** - Method in class
javax.faces.validator.**DoubleRangeValidator**

**setTransient(boolean)** - Method in class
javax.faces.validator.**LengthValidator**

**setTransient(boolean)** - Method in class
javax.faces.validator.**LongRangeValidator**

**setTransient(boolean)** - Method in class
javax.faces.validator.**MethodExpressionValidator**

**setType(String)** - Method in class
javax.faces.component.html.**HtmlCommandButton**
    Set the value of the type property.

**setType(String)** - Method in class
javax.faces.component.html.**HtmlCommandLink**
    Set the value of the type property.

**setType(String)** - Method in class
javax.faces.component.html.**HtmlOutputLink**
    Set the value of the type property.

**setType(String)** - Method in class
javax.faces.convert.**DateTimeConverter**
    Set the type of value to be formatted or parsed.

**setType(String)** - Method in class
javax.faces.convert.**NumberConverter**
    Set the number type to be used when formatting and parsing numbers.

**setType(String)** - Method in interface
javax.xml.registry.infomodel.**EmailAddress**
    Sets the type for this object.

**setType(String)** - Method in interface
javax.xml.registry.infomodel.**PostalAddress**
    Sets the type of address (for example, "headquarters") as a String.

**setType(String)** - Method in interface
javax.xml.registry.infomodel.**TelephoneNumber**
    Sets the type of telephone number (for example, "fax").
**setType(String)** - Method in interface javax.xml.registry.infomodel.User
Sets the type for this User.

**setupResponseWriter()** - Method in class javax.faces.webapp.UICOMPONENTCLASSICTAGBASE
Set up the ResponseWriter for the current response, if this has not been done already.

**setURL(String)** - Method in class javax.faces.component.UIGRAPHIC
Set the image URL for this UIGraphic.

**setURL(URL)** - Method in class javax.xml.bind.helpers.VALIDATIONEVENTLOCATORIMPL
Set the URL field on this event locator.

**setUrl(String)** - Method in interface javax.xml.registry.infomodel.TELPHONENUMBER
Sets the URL that can dial this number electronically.

**setUrl(URL)** - Method in interface javax.xml.registry.infomodel.USER
Sets the URL to the web page for this User.

**setURLName(URLName)** - Method in class javax.mail.SERVICE
Set the URLName representing this service.

**setUsageDescription(InternationalString)** - Method in interface javax.xml.registry.infomodel.SPECIFICATIONLINK
Sets the description of usage parameters.

**setUsageParameters(Collection)** - Method in interface javax.xml.registry.infomodel.SPECIFICATIONLINK
Sets any usage parameters.

**setUsemap(String)** - Method in class javax.faces.component.html.HTMLGRAPHICIMAGE
Set the value of the usemap property.

**setUserVersion(String)** - Method in interface javax.xml.registry.infomodel.VERSIONABLE
Sets the user specified revision number for this version of the Versionable object.

**setValid(boolean)** - Method in interface javax.faces.component.EDITABLEVALUEHOLDER
Set a flag indicating whether the local value of this component is valid (no conversion error has occurred).

**setValid(boolean)** - Method in class javax.faces.component.UIINPUT

**setValidateURI(boolean)** - Method in interface javax.xml.registry.infomodel.URIVALIDATOR
Sets whether to do URI validation for this object.

**setValidating(boolean)** - Method in class javax.xml.bind.helpers.ABSTRACTUNMARSHALLERIMPL
Specifies whether or not the Unmarshaller should validate during unmarshal operations.

**setValidating(boolean)** - Method in interface javax.xml.bind.UNMARSHALLER
**Deprecated. since JAXB2.0, please see Unmarshaller.setSchema(javax.xml.validation.Schema)**

**setValidator(MethodBinding)** - Method in interface
javax.faces.component.EditableValueHolder

Deprecated. Use EditableValueHolder.addValidator(javax.faces.validator.Validator) instead, obtaining the argument Validator by creating an instance of MethodExpressionValidator.

**setValidator(MethodBinding)** - Method in class javax.faces.component.UIInput

Deprecated. Use UIInput.addValidator(javax.faces.validator.Validator) instead, obtaining the argument Validator by creating an instance of MethodExpressionValidator.

**setValidatorId(String)** - Method in class javax.faces.webapp.ValidatorTag

Deprecated. Set the identifier of the Validator instance to be created.

**setValidatorMessage(String)** - Method in class javax.faces.component.UIInput

Override any ValueExpression set for the "validatorMessage" with the literal argument provided to this method.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.ArrayELResolver

If the base object is a Java language array, attempts to set the value at the given index with the given value.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.BeanELResolver

If the base object is not null, attempts to set the value of the given property on this bean.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.CompositeELResolver

Attempts to set the value of the given property object on the given base object.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.EELResolver

Attempts to set the value of the given property object on the given base object.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.ListELResolver

If the base object is a list, attempts to set the value at the given index with the given value.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.MapELResolver

If the base object is a map, attempts to set the value associated with the given key, as specified by the property argument.

**setValue(ELContext, Object, Object, Object)** - Method in class javax.el.ResourceBundleELResolver

If the base object is a ResourceBundle, throw a PropertyNotWritableException.

**setValue(ELContext, Object)** - Method in class javax.el.ValueExpression
Evaluates the expression relative to the provided context, and sets the result to the provided value.

`setValue(Object)` - Method in class `javax.faces.component.UICommand`  
Sets the value property of the `UICommand`.

`setValue(Object)` - Method in class `javax.faces.component.UIData`  
Set the value of the `UIData`.

`setValue(Object)` - Method in class `javax.faces.component.UIGraphic`  
Sets the value property of the `UIGraphic`.

`setValue(Object)` - Method in class `javax.faces.component.UIInput`  

`setValue(Object)` - Method in class `javax.faces.component.UIOutput`  

`setValue(Object)` - Method in class `javax.faces.component.UIParameter`  
Sets the value property of the `UIParameter`.

`setValue(Object)` - Method in class `javax.faces.component.UISelectItem`  
Sets the value property of the `UISelectItem`.

`setValue(Object)` - Method in class `javax.faces.component.UISelectItems`  
Sets the value property of the `UISelectItems`.

`setValue(Object)` - Method in interface `javax.faces.component.ValueHolder`  
Set the value of this `UIComponent` (if any).

`setValue(Object, Object, Object)` - Method in class `javax.faces.el.PropertyResolver`  
Deprecated. Set the specified value of the specified property on the specified base object.

`setValue(Object, int, Object)` - Method in class `javax.faces.el.PropertyResolver`  
Deprecated. Set the value at the specified index of the specified base object.

`setValue(FacesContext, Object)` - Method in class `javax.faces.el.ValueBinding`  
Deprecated. Set the value of the property represented by this `ValueBinding`, relative to the specified `FacesContext`.

`setValue(Object)` - Method in class `javax.faces.model.SelectItem`  
Set the value of this item, to be delivered to the model if this item is selected by this user.

`setValue(String)` - Method in class `javax.faces.webapp.AttributeTag`  
Deprecated. Set the attribute value.

`setValue(String)` - Method in class `javax.servlet.http.Cookie`  
Assigns a new value to a cookie after the cookie is created.

`setValue(ELContext, Object, Object, Object)` - Method in class `javax.servlet.jsp.el.ImplicitObjectELResolver`  
If the base object is null, and the property matches the name of a JSP implicit object, throws `PropertyNotWritableException` to indicate that implicit objects cannot be overwritten.

`setValue(ELContext, Object, Object, Object)` - Method in class `javax.servlet.jsp.el.ScopedAttributeELResolver`
If the base object is null, sets an existing scoped attribute to the new value, or creates a new scoped attribute if one does not exist by this name.

**setValue(String, Object)** - Method in class
javax.servlet.jsp.tagext.TagSupport
Associate a value with a String key.

**setValue(T)** - Method in class javax.xml.bind.JAXBElem
Set the content model and attributes of this xml element.

**setValue(String)** - Method in interface
javax.xml.registry.infomodel.Classification
Sets the taxonomy value for this external Classification.

**setValue(String)** - Method in interface
javax.xml.registry.infomodel.Concept
Sets the value (usually a taxonomy value) associated with this Concept.

**setValue(String)** - Method in interface
docs.methods.LinkedHashMap
Sets the value for the specified Locale.

**setValue(String)** - Method in interface
javax.xml.soap.Node
If this is a Text node then this method will set its value, otherwise it sets the value of the immediate (Text) child of this node.

**setValueBinding(String, ValueBinding)** - Method in class
javax.faces.component.UIComponent
*Deprecated. This has been replaced by*

**setValueBinding(String, ValueBinding)** - Method in class
javax.faces.component.UIComponentBase
*Deprecated. This has been replaced by*

**setValueBinding(String, ValueBinding)** - Method in class
javax.faces.component.UIData
*Deprecated. This has been replaced by*

**setValueBinding(String, ValueBinding)** - Method in class
javax.faces.component.UIGraphic
*Deprecated. This has been replaced by*

setValueBinding(String, ValueBinding) - Method in class
javax.faces.component.UISelectBoolean

Deprecated. This has been replaced by

setValueBinding(String, ValueBinding) - Method in class
javax.faces.component.UISelectMany

Deprecated. This has been replaced by

setValueChangeListener(MethodBinding) - Method in interface
javax.faces.component.EditableValueHolder

Deprecated. Use
EditableValueHolder.addValueChangeListener(javax.faces.event.ValueChangeListener)
instead, obtaining the argument ValueChangeListener by creating
an instance of MethodExpressionValueChangeListener.

setValueChangeListener(MethodBinding) - Method in class
javax.faces.component.UIInput

Deprecated. Use
UIInput.addValueChangeListener(javax.faces.event.ValueChangeListener)
instead, obtaining the argument ValueChangeListener by creating
an instance of MethodExpressionValueChangeListener.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UICOMPONENT

Set the ValueExpression used to calculate the value for the
specified attribute or property name, if any.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UIData

Set the ValueExpression used to calculate the value for the
specified attribute or property name, if any.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UIGraphic

Store any ValueExpression specified for url under value
instead; otherwise, perform the default superclass processing
for this method.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UISelectBoolean

Store any ValueExpression specified for selected under value
instead; otherwise, perform the default superclass processing
for this method.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UISelectMany

Store any ValueExpression specified for selectedValues under
value instead; otherwise, perform the default superclass processing
for this method.

setValueExpression(String, ValueExpression) - Method in class
javax.faces.component.UIData

Set the values for this Slot.
**setValueType(int)** - Method in interface javax.xml.registry.info.model.ClassificationScheme
Sets the value type for this object.

**setVar(String)** - Method in class javax.faces.component.UIData
Set the request-scope attribute under which the data object for the current row will be exposed when iterating.

**setVariable(String, ValueExpression)** - Method in class javax.el.VariableMapper
Assign a ValueExpression to an EL variable, replacing any previously assignment to the same variable.

**setVariableResolver(VariableResolver)** - Method in class javax.faces.application.Application
**Deprecated.** The recommended way to affect the execution of the EL is to provide an `<el-resolver>` element at the right place in the application configuration resources which will be considered in the normal course of expression evaluation. This method now will cause the argument resolver to be wrapped inside an implementation of `ELResolver` and exposed to the EL resolution system as if the user had called `Application.addELResolver(javax.el.ELResolver)`.

Sets the version of the cookie protocol this cookie complies with.

**setViewHandler(ViewHandler)** - Method in class javax.faces.application.Application
Set the `ViewHandler` instance that will be utilized during the Restore View and Render Response phases of the request processing lifecycle.

**setId(String)** - Method in class javax.faces.component.UIData
Set the view identifier for this view.

**setViewRoot(UIDataViewRoot)** - Method in class javax.faces.context.FacesContext
Set the root component that is associated with this request.

**setWarnClass(String)** - Method in class javax.faces.component.html.HtmlMessage
Set the value of the warnClass property.

**setWarnStyle(String)** - Method in class javax.faces.component.html.HtmlMessage
Set the value of the warnStyle property.

**setWidth(String)** - Method in class javax.faces.component.html.HtmlDataItem
Set the value of the width property.

**setWidth(String)** - Method in class
javax.faces.component.html.HtmlGraphicImage
   Set the value of the width property.

setWidth(String) - Method in class javax.faces.component.html.HtmlPanelGrid
   Set the value of the width property.

setWrappedData(Object) - Method in class javax.faces.model.ArrayDataModel
   Set the object representing the data collection wrapped by this DataModel.

setWrappedData(Object) - Method in class javax.faces.model.ListDataModel

setWrappedData(Object) - Method in class javax.faces.model.ResultSetDataModel

setWrappedData(Object) - Method in class javax.faces.model.ScalarDataModel

setXid(Xid) - Method in class javax.resource.spi.work.ExecutionContext
   set a transaction context.

setXMLReporter(XMLReporter) - Method in class javax.xml.stream.XMLInputFactory
   The reporter that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

setXMLResolver(XMLResolver) - Method in class javax.xml.stream.XMLInputFactory
   The resolver that will be set on any XMLStreamReader or XMLEventReader created by this factory instance.

SEVERITY_ERROR - Static variable in class javax.faces.application.FacesMessage
   Message severity level indicating that an error has occurred.

SEVERITY_FATAL - Static variable in class javax.faces.application.FacesMessage
   Message severity level indicating that a serious error has occurred.

SEVERITY_INFO - Static variable in class javax.faces.application.FacesMessage
   Message severity level indicating an informational message rather than an error.

SEVERITY_WARN - Static variable in class javax.faces.application.FacesMessage
   Message severity level indicating that an error might have occurred.
**SharedByteArrayInputStream** - Class in `javax.mail.util`
A ByteArrayInputStream that implements the SharedInputStream interface, allowing the underlying byte array to be shared between multiple readers.

**SharedByteArrayInputStream(byte[])** - Constructor for class `javax.mail.util.SharedByteArrayInputStream`
Create a SharedByteArrayInputStream representing the entire byte array.

**SharedByteArrayInputStream(byte[], int, int)** - Constructor for class `javax.mail.util.SharedByteArrayInputStream`
Create a SharedByteArrayInputStream representing the part of the byte array from offset for length bytes.

**SharedFileInputStream** - Class in `javax.mail.util`
A SharedFileInputStream is a BufferedInputStream that buffers data from the file and supports the mark and reset methods.

**SharedFileInputStream(File)** - Constructor for class `javax.mail.util.SharedFileInputStream`
Creates a SharedFileInputStream for the file.

**SharedFileInputStream(String)** - Constructor for class `javax.mail.util.SharedFileInputStream`
Creates a SharedFileInputStream for the named file.

**SharedFileInputStream(File, int)** - Constructor for class `javax.mail.util.SharedFileInputStream`
Creates a SharedFileInputStream with the specified buffer size.

**SharedFileInputStream(String, int)** - Constructor for class `javax.mail.util.SharedFileInputStream`
Creates a SharedFileInputStream with the specified buffer size.

**SharedInputStream** - Interface in `javax.mail.internet`
An InputStream that is backed by data that can be shared by multiple readers may implement this interface.

**SharingViolationException** - Exception in `javax.resource.spi`
This is thrown to indicate a connection sharing violation.

**SharingViolationException()** - Constructor for exception `javax.resource.spi.SharingViolationException`
Constructs a new instance with null as its detail message.

**SharingViolationException(String)** - Constructor for exception `javax.resource.spi.SharingViolationException`
Constructs a new instance with the specified detail message.

**SharingViolationException(Throw)able** - Constructor for exception `javax.resource.spi.SharingViolationException`
Constructs a new throwable with the specified cause.

**SharingViolationException(String, Throwable)** - Constructor for exception `javax.resource.spi.SharingViolationException`
Constructs a new throwable with the specified detail message and cause.

**SharingViolationException(String, String)** - Constructor for exception `javax.resource.spi.SharingViolationException`
Constructs a new throwable with the specified detail message and error code.

**SHORT_ID** - Static variable in class
javax.faces.convert.ShortConverter
The message identifier of the FacesMessage to be created if the conversion to Short fails.

ShortConverter - Class in javax.faces.convert
Converter implementation for java.lang.Short (and short primitive) values.

ShortConverter() - Constructor for class
javax.faces.convert.ShortConverter

ShortHolder - Class in javax.xml.rpc.holders

ShortHolder() - Constructor for class
javax.xml.rpc.holders.ShortHolder

ShortHolder(short) - Constructor for class
javax.xml.rpc.holders.ShortHolder

shortname - Variable in class
javax.servlet.jsp.tagext.TagLibraryInfo
The preferred short name (prefix) as indicated in the TLD.

ShortWrapperHolder - Class in javax.xml.rpc.holders

ShortWrapperHolder() - Constructor for class
javax.xml.rpc.holders.ShortWrapperHolder

ShortWrapperHolder(Short) - Constructor for class
javax.xml.rpc.holders.ShortWrapperHolder

SimpleTag - Interface in javax.servlet.jsp.tagext
Interface for defining Simple Tag Handlers.

SimpleTagSupport - Class in javax.servlet.jsp.tagext
A base class for defining tag handlers implementing SimpleTag.

SimpleTagSupport() - Constructor for class
javax.servlet.jsp.tagext.SimpleTagSupport
Sole constructor.

SimpleThreadModel - Interface in javax.servlet
Deprecated. As of Java Servlet API 2.4, with no direct replacement.

size() - Method in class javax.activation.MimeTypeParameterList
Return the number of name-value pairs in this list.

size() - Method in class javax.mail.internet.ParameterList
Return the number of parameters in this list.

SizeTerm - Class in javax.mail.search
This class implements comparisons for Message sizes.

SizeTerm(int, int) - Constructor for class
javax.mail.search.SizeTerm
Constructor.

skip(long) - Method in class javax.mail.util.SharedFileInputStream
See the general contract of the skip method of InputStream.

SKIP_BODY - Static variable in interface
javax.servlet.jsp.tagext.Tag
Skip body evaluation.

SKIP_PAGE - Static variable in interface
javax.servlet.jsp.tagext.Tag
Skip the rest of the page.

SkipPageException - Exception in javax.servlet.jsp
Exception to indicate the calling page must cease evaluation.

SkipPageException() - Constructor for exception
javax.servlet.jsp.SkipPageException
Creates a SkipPageException with no message.

SkipPageException(String) - Constructor for exception
javax.servlet.jsp.SkipPageException
Creates a SkipPageException with the provided message.

SkipPageException(String, Throwable) - Constructor for exception
javax.servlet.jsp.SkipPageException
Creates a SkipPageException with the provided message and root cause.

SkipPageException( Throwable) - Constructor for exception
javax.servlet.jsp.SkipPageException
Creates a SkipPageException with the provided root cause.

Slot - Interface in javax.xml.registry.infomodel
Slot instances provide a dynamic way to add arbitrary attributes to RegistryObject instances.

SLOT - Static variable in interface
javax.xml.registry.LifeCycleManager

SOAP11HTTP_BINDING - Static variable in interface
javax.xml.ws.soap.SOAPBinding
A constant representing the identity of the SOAP 1.1 over HTTP binding.

SOAP11HTTP_MTOM_BINDING - Static variable in interface
javax.xml.ws.soap.SOAPBinding
A constant representing the identity of the SOAP 1.1 over HTTP binding with MTOM enabled by default.

SOAP12HTTP_BINDING - Static variable in interface
javax.xml.ws.soap.SOAPBinding
A constant representing the identity of the SOAP 1.2 over HTTP binding.

SOAP12HTTP_MTOM_BINDING - Static variable in interface
javax.xml.ws.soap.SOAPBinding
A constant representing the identity of the SOAP 1.2 over HTTP binding with MTOM enabled by default.

SOAP_1_1_CONTENT_TYPE - Static variable in interface
javax.xml.soap.SOAPConstants
The media type of the Content-Type MIME header in SOAP 1.1.

SOAP_1_1_PROTOCOL - Static variable in interface
javax.xml.soap.SOAPConstants
Used to create MessageFactory instances that create SOAPMessages whose behavior supports the SOAP 1.1 specification.
**SOAP 1.2 CONTENT TYPE** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   The media type of the Content-Type MIME header in SOAP 1.2.

**SOAP 1.2 PROTOCOL** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   Used to create MessageFactory instances that create SOAPMessages whose behavior supports the SOAP 1.2 specification.

**SOAP_ARRAY** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:Array type.

**SOAP_BASE64** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:base64 type.

**SOAP_BOOLEAN** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:boolean type.

**SOAP_BYTE** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:byte type.

**SOAP_DATAENCODINGUNKNOWN_FAULT** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   SOAP 1.2 DataEncodingUnknown Fault.

**SOAP_DOUBLE** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:double type.

**SOAP_ENV_PREFIX** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   The default namespace prefix for http://www.w3.org/2003/05/soap-envelope.

**SOAP_FLOAT** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:float type.

**SOAP_INT** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:int type.

**SOAP_LONG** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:long type.

**SOAP_MUSTUNDERSTAND_FAULT** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   SOAP 1.2 MustUnderstand Fault.

**SOAP_RECEIVER_FAULT** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   SOAP 1.2 Receiver Fault.

**SOAP_SENDER_FAULT** - Static variable in interface javax.xml.soap.SOAPOutputProperties
   SOAP 1.2 Sender Fault.

**SOAP_SHORT** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:short type.

**SOAP_STRING** - Static variable in class javax.xml.rpc.encoding.XMLType
   The name of the SOAP-ENC:string type.
SOAP_VERSIONMISMATCH_FAULT - Static variable in interface javax.xml.soap.SOAPConstants
SOAP 1.2 VersionMismatch Fault

SOAPACTION_URI_PROPERTY - Static variable in interface javax.xml.rpc.Call
Standard property for SOAPAction.

SOAPACTION_URI_PROPERTY - Static variable in interface javax.xml.ws.BindingProvider
Standard property for SOAPAction.

SOAPACTION_USE_PROPERTY - Static variable in interface javax.xml.rpc.Call
Standard property for SOAPAction.

SOAPACTION_USE_PROPERTY - Static variable in interface javax.xml.ws.BindingProvider
Standard property for SOAPAction.

SOAPBinding - Annotation Type in javax.jws.soap
Specifies the mapping of the Web Service onto the SOAP message protocol.

SOAPBinding - Interface in javax.xml.ws.soap
The SOAPBinding interface is an abstraction for the SOAP binding.

SOAPBinding.ParameterStyle - Enum in javax.jws.soap
The style of mapping parameters onto SOAP messages

SOAPBinding.Style - Enum in javax.jws.soap
The SOAP binding style

SOAPBinding.Use - Enum in javax.jws.soap
The SOAP binding use

SOAPBody - Interface in javax.xml.soap
An object that represents the contents of the SOAP body element in a SOAP message.

SOAPBodyElement - Interface in javax.xml.soap
A SOAPBodyElement object represents the contents in a SOAPBody object.

SOAPConnection - Class in javax.xml.soap
A point-to-point connection that a client can use for sending messages directly to a remote party (represented by a URL, for instance).

SOAPConnection() - Constructor for class javax.xml.soap.SOAPConnection

SOAPConnectionFactory - Class in javax.xml.soap
A factory for creating SOAPConnection objects.

SOAPConnectionFactory() - Constructor for class javax.xml.soap.SOAPConnectionFactory

SOAPConstants - Interface in javax.xml.soap
The definition of constants pertaining to the SOAP protocol.

SOAPElement - Interface in javax.xml.soap
An object representing an element of a SOAP message that is allowed but not specifically prescribed by a SOAP
**SOAPElementFactory** - Class in *javax.xml.soap*
Deprecated. - Use *javax.xml.soap.SOAPFactory* for creating SOAPElements.

**SOAPEnvelope** - Interface in *javax.xml.soap*
The container for the SOAPHeader and SOAPBody portions of a SOAPPart object.

**SOAPException** - Exception in *javax.xml.soap*
An exception that signals that a SOAP exception has occurred.

**SOAPException()** - Constructor for exception
*javax.xml.soap.SOAPException*
Constructs a SOAPException object with no reason or embedded Throwable object.

**SOAPException(String)** - Constructor for exception
*javax.xml.soap.SOAPException*
Constructs a SOAPException object with the given String as the reason for the exception being thrown.

**SOAPException(String, Throwable)** - Constructor for exception
*javax.xml.soap.SOAPException*
Constructs a SOAPException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded exception.

**SOAPException(Throwables)** - Constructor for exception
*javax.xml.soap.SOAPException*
Constructs a SOAPException object initialized with the given Throwable object.

**SOAPFactory** - Class in *javax.xml.soap*
SOAPFactory is a factory for creating various objects that exist in the SOAP XML tree.

**SOAPFactory()** - Constructor for class *javax.xml.soap.SOAPFactory*

**SOAPFault** - Interface in *javax.xml.soap*
An element in the SOAPBody object that contains error and/or status information.

**SOAPFaultElement** - Interface in *javax.xml.soap*
A representation of the contents in a SOAPFault object.

**SOAPFaultException** - Exception in *javax.xml.rpc.soap*
The SOAPFaultException exception represents a SOAP fault.

**SOAPFaultException(QName, String, String, Detail)** - Constructor for exception
*javax.xml.rpc.soap.SOAPFaultException*
Constructor for the SOAPFaultException

**SOAPFaultException** - Exception in *javax.xml.ws.soap*
The SOAPFaultException exception represents a SOAP 1.1 or 1.2 fault.

**SOAPFaultException(SOAPFault)** - Constructor for exception
*javax.xml.ws.soap.SOAPFaultException*
Constructor for SOAPFaultException

**SOAPHandler<T extends SOAPMessageContext>** - Interface in *javax.xml.ws.handler.soap*
The SOAPHandler class extends Handler to provide typesafety for...
the message context parameter and add a method to obtain access to the headers that may be processed by the handler.

**SOAPHeader** - Interface in javax.xml.soap
A representation of the SOAP header element.

**SOAPHeaderElement** - Interface in javax.xml.soap
An object representing the contents in the SOAP header part of the SOAP envelope.

**SOAPMessage** - Class in javax.xml.soap
The root class for all SOAP messages.

**SOAPMessage()** - Constructor for class javax.xml.soap.SOAPOperation

**SOAPMessageContext** - Interface in javax.xml.rpc.handler.soap
The interface javax.xml.rpc.soap.SOAPMessageContext provides access to the SOAP message for either RPC request or response.

**SOAPMessageContext** - Interface in javax.xml.ws.handler.soap
The interface SOAPMessageContext provides access to the SOAP message for either RPC request or response.

**SOAPMessageHandler** - Annotation Type in javax.jws.soap
Deprecated. As of JSR-181 2.0 with no replacement.

**SOAPMessageHandlers** - Annotation Type in javax.jws.soap
Deprecated. As of JSR-181 2.0 with no replacement.

**SOAPPart** - Class in javax.xml.soap
The container for the SOAP-specific portion of a SOAPMessage object.

**SOAPPart()** - Constructor for class javax.xml.soap.SOAPOperation

**SORT_BY_DATE_ASC** - Static variable in interface javax.xml.registry.FindQualifier

**SORT_BY_DATE_DESC** - Static variable in interface javax.xml.registry.FindQualifier

**SORT_BY_NAME_ASC** - Static variable in interface javax.xml.registry.FindQualifier

**SORT_BY_NAME_DESC** - Static variable in interface javax.xml.registry.FindQualifier

**SORT_CODE_SLOT** - Static variable in interface javax.xml.registry.infomodel.Slot
Name for pre-defined Slot used in PostalAddress by JAXR UDDI provider.

**SOUNDEX** - Static variable in interface javax.xml.registry.FindQualifier
Optional qualifier that allows matching strings by their sounds.

**SPACE** - Static variable in interface javax.xml.stream.XMLStreamConstants
The characters are white space (see [XML], 2.10 "White Space Handling").
**SPECIFICATION_LINK** - Static variable in interface
javax.xml.registry.*LifeCycleManager*

**SpecificationLink** - Interface in *javax.xml.registry.infomodel*
A SpecificationLink provides the linkage between a ServiceBinding and one of its technical specifications that describes how to use the service using the ServiceBinding.

**SqlResultSetMapping** - Annotation Type in *javax.persistence*
This annotation is used to specify the mapping of the result of a native SQL query.

**SqlResultSetMappings** - Annotation Type in *javax.persistence*
This annotation is used to define one or more SqlResultSetMapping.

**STABILITY_DYNAMIC** - Static variable in interface
javax.xml.registry.infomodel.*RegistryEntry*
RegistryEntry may change at any time.

**STABILITY_DYNAMIC_COMPATIBLE** - Static variable in interface
javax.xml.registry.infomodel.*RegistryEntry*
RegistryEntry may change at any time, however the changes will be backward compatible.

**STABILITY_STATIC** - Static variable in interface
javax.xml.registry.infomodel.*RegistryEntry*
RegistryEntry will not change.

**standaloneSet()** - Method in interface
javax.xml.stream.events.*StartDocument*
Returns true if the standalone attribute was set in the encoding declaration of the document.

**standaloneSet()** - Method in class
javax.xml.stream.util.*StreamReaderDelegate*

**standaloneSet()** - Method in interface
javax.xml.stream.*XMLStreamReader*
Checks if standalone was set in the document

**START** - Static variable in class
javax.enterprise.deploy.shared.*CommandType*
The DeploymentManger action operation being processed is start.

**start(TargetModuleID[])** - Method in interface
javax.enterprise.deploy.spi.*DeploymentManager*
Start the application running.

**start()** - Method in interface java.jms.*Connection*
Starts (or restarts) a connection's delivery of incoming messages.

**start()** - Method in interface java.jms.*ServerSession*
Cause the Session's run method to be called to process messages that were just assigned to it.

**start** - Variable in class
javax.mail.util.*SharedByteArrayInputStream*
Position within shared buffer that this stream starts at.

**start** - Variable in class java.mail.util.*SharedFileInputStream*
The file offset of the start of data in this subset of the
file.

**start(BootstrapContext)** - Method in interface javax.resource.spi.ResourceAdapter
This is called when a resource adapter instance is bootstrapped.

**start(Xid, int)** - Method in interface javax.transaction.xa.XAResource
Starts work on behalf of a transaction branch specified in xid.

**START DOCUMENT** - Static variable in interface javax.xml.stream.XMLStreamConstants
Indicates an event is a start document

**START ELEMENT** - Static variable in interface javax.xml.stream.XMLStreamConstants
Indicates an event is a start element

**START_TIMED_OUT** - Static variable in exception javax.resource.spi.work.WorkException
Indicates start timeout expiration.

**startDocument()** - Method in class javax.faces.context.ResponseWriter
Write whatever text should begin a response.

**startDocument()** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call ResponseWriter.startDocument() on the wrapped ResponseWriter object.

**StartDocument** - Interface in javax.xml.stream.events
An interface for the start document event

**startElement(String, UIComponent)** - Method in class javax.faces.context.ResponseWriter
Write the start of an element, up to and including the element name.

**startElement(String, UIComponent)** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call ResponseWriter.startElement(String, javax.faces.component.UIComponent) on the wrapped ResponseWriter object.

**StartElement** - Interface in javax.xml.stream.events
The StartElement interface provides access to information about start elements.

**startWork(Work)** - Method in interface javax.resource.spi.work.WorkManager
Accepts a Work instance for processing.

**startWork(Work, long, ExecutionContext, WorkListener)** - Method in interface javax.resource.spi.work.WorkManager
Accepts a Work instance for processing.

**STATE_SAVING_METHOD_CLIENT** - Static variable in class javax.faces.application.StateManager
Constant value for the initialization parameter named by the STATE_SAVING_METHOD_PARAM_NAME that indicates state saving
should take place on the client.

**STATE_SAVING_METHOD_PARAM_NAME** - Static variable in class `javax.faces.application.StateManager`

The ServletContext init parameter consulted by the StateManager to tell where the state should be saved.

**STATE_SAVING_METHOD_SERVER** - Static variable in class `javax.faces.application.StateManager`

Constant value for the initialization parameter named by the STATE_SAVING_METHOD_PARAM_NAME that indicates state saving should take place on the server.

**Stateful** - Annotation Type in `javax.ejb`

Component-defining annotation for a stateful session bean.

**StatefulSessionBeanStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a stateful session bean.

**StateHolder** - Interface in `javax.faces.component`

This interface is implemented by classes that need to save their state between requests.

**Stateless** - Annotation Type in `javax.ejb`

Component-defining annotation for a stateless session bean.

**StatelessSessionBeanStats** - Interface in `javax.management.j2ee.statistics`

Specifies the statistics provided by a stateless session bean.

**StateManager** - Class in `javax.faces.application`.

StateManager directs the process of saving and restoring the view between requests.

**StateManager()** - Constructor for class `javax.faces.application.StateManager`

**StateManager.SerializedView** - Class in `javax.faces.application`.

Deprecated. This class was not marked Serializable in the 1.0 version of the spec. It was also not a static inner class, so it can't be made to be Serializable. Therefore, it is being deprecated in version 1.2 of the spec. The replacement is to use an implementation dependent Object.

**StateManager.SerializedView(Object, Object)** - Constructor for class `javax.faces.application.StateManager.SerializedView`

Deprecated.

**StateManagerWrapper** - Class in `javax.faces.application`.

Provides a simple implementation of `StateManager` that can be subclassed by developers wishing to provide specialized behavior to an existing `StateManager` instance.

**StateManagerWrapper()** - Constructor for class `javax.faces.application.StateManagerWrapper`

**StateType** - Class in `javax.enterprise.deploy.shared`.

Class StateTypes defines enumeration values for the DeploymentStatus object.

**StateType(int)** - Constructor for class `javax.enterprise.deploy.shared.StateType`
Construct a new enumeration value with the given integer value.

**Statistic** - Interface in `javax.management.j2ee.statistics`
The Statistic model and its sub-models specify the data models which are required to be used to provide the performance data described by the specific attributes in the Stats models.

**Stats** - Interface in `javax.management.j2ee.statistics`
The Stats model and its submodels specify performance data attributes for each of the specific managed object types.

**Status** - Interface in `javax.transaction`
The Status interface defines static variables used for transaction status codes.

**STATUS_ACTIVE** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it is in the active state.

**STATUS_APPROVED** - Static variable in interface `javax.xml.registry.infomodel.RegistryEntry`
RegistryEntry has been submitted and approved.

**STATUS_COMMITTED** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it has been committed.

**STATUS_COMMITTING** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it is in the process of committing.

**STATUS_DEPRECATED** - Static variable in interface `javax.xml.registry.infomodel.RegistryEntry`
RegistryEntry has been deprecated.

**STATUS_FAILURE** - Static variable in interface `javax.xml.registry.JAXRResponse`
Status indicating a failure response.

**STATUS_MARKED_ROLLBACK** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it has been marked for rollback, perhaps as a result of a setRollbackOnly operation.

**STATUS_NO_TRANSACTION** - Static variable in interface `javax.transaction.Status`
No transaction is currently associated with the target object.

**STATUS_PREPARED** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it has been prepared.

**STATUS_PREPARING** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and it is in the process of preparing.

**STATUS_ROLLED_BACK** - Static variable in interface `javax.transaction.Status`
A transaction is associated with the target object and the outcome has been determined to be rollback.

**STATUS_ROLLING_BACK** - Static variable in interface javax.transaction.Status

A transaction is associated with the target object and it is in the process of rolling back.

**STATUS_SUBMITTED** - Static variable in interface javax.xml.registry.infomodel.RegistryEntry

RegistryEntry has been submitted.

**STATUS_SUCCESS** - Static variable in interface javax.xml.registry.JAXRResponse

Status indicating a successful response.

**STATUS_UNAVAILABLE** - Static variable in interface javax.xml.registry.JAXRResponse

Status indicating that the results are currently unavailable.

**STATUS_UNKNOWN** - Static variable in interface javax.transaction.Status

A transaction is associated with the target object but its current status cannot be determined.

**STATUS_WARNING** - Static variable in interface javax.xml.registry.JAXRResponse

Status indicating a successful response that included at least one warning.

**STATUS_WITHDRAWN** - Static variable in interface javax.xml.registry.infomodel.RegistryEntry

RegistryEntry has been withdrawn by the submitter.

**STOP** - Static variable in class javax.enterprise.deploy.shared.ActionType

A stop operation is being preformed on the DeploymentManager action command.

**STOP** - Static variable in class javax.enterprise.deploy.shared.CommandType

The DeploymentManager action operation being processed is stop.

**stop(TargetModuleID[])** - Method in interface javax.enterprise.deploy.spi.DeploymentManager

Stop the application running.

**stop()** - Method in interface javax.enterprise.deploy.spi.status.ProgressObject

(optional) A stop request on an in-process operation allows the operation on the current TargetModuleID to run to completion but does not process any of the remaining unprocessed TargetModuleID objects.

**stop()** - Method in interface javax.jms.Connection

Temporarily stops a connection's delivery of incoming messages.

**stop()** - Method in interface javax.resource.spi.ResourceAdapter

This is called when a resource adapter instance is undeployed or during application server shutdown.

**stop()** - Method in class javax.xml.ws.Endpoint

Stops publishing this endpoint.

**store** - Variable in class javax.mail.Folder
The parent store.

**Store** - Static variable in class javax.mail.**Provider.Type**

**Store** - Class in javax.mail
An abstract class that models a message store and its access protocol, for storing and retrieving messages.

**Store(Store, URLName)** - Constructor for class javax.mail.Store
Constructor.

**StoreClosedException** - Exception in javax.mail
This exception is thrown when a method is invoked on a Messaging object and the Store that owns that object has died due to some reason.

**StoreClosedException(Store)** - Constructor for exception javax.mail.StoreClosedException
Constructor

**StoreClosedException(Store, String)** - Constructor for exception javax.mail.StoreClosedException
Constructor

**StoreEvent** - Class in javax.mail.event
This class models notifications from the Store connection.

**StoreEvent(Store, int, String)** - Constructor for class javax.mail.event.StoreEvent
Constructor.

**StoreListener** - Interface in javax.mail.event
This is the Listener interface for Store Notifications.

**Streamable** - Interface in javax.resource.cci
Streamable interface enables a resource adapter to extract data from an input Record or set data into an output Record as a stream of bytes.

**StreamFilter** - Interface in javax.xml.stream
This interface declares a simple filter interface that one can create to filter XMLStreamReaders

**StreamMessage** - Interface in javax.jms
A StreamMessage object is used to send a stream of primitive types in the Java programming language.

**StreamReaderDelegate** - Class in javax.xml.stream.util
This is the base class for deriving an XMLStreamReader filter
This class is designed to sit between an XMLStreamReader and an application's XMLStreamReader.

**StreamReaderDelegate()** - Constructor for class javax.xml.stream.util.StreamReaderDelegate
Construct an empty filter with no parent.

**StreamReaderDelegate(XMLStreamReader)** - Constructor for class javax.xml.stream.util.StreamReaderDelegate
Construct an filter with the specified parent.

**STRING_ID** - Static variable in class javax.faces.convert.BigDecimalConverter
The message identifier of the FacesMessage to be created if the conversion of the BigDecimal value to String fails.

**STRING_ID** - Static variable in class
BigIntegerConverter
The message identifier of the FacesMessage to be created if the conversion of the BigInteger value to String fails.

STRING_ID - Static variable in class

BooleanConverter
The message identifier of the FacesMessage to be created if the conversion of the Boolean value to String fails.

STRING_ID - Static variable in class

ByteConverter
The message identifier of the FacesMessage to be created if the conversion of the Byte value to String fails.

STRING_ID - Static variable in class

CharacterConverter
The message identifier of the FacesMessage to be created if the conversion of the Character value to String fails.

STRING_ID - Static variable in class

DateTimeConverter
The message identifier of the FacesMessage to be created if the conversion of the DateTime value to String fails.

STRING_ID - Static variable in class

DoubleConverter
The message identifier of the FacesMessage to be created if the conversion of the Double value to String fails.

STRING_ID - Static variable in class

FloatConverter
The message identifier of the FacesMessage to be created if the conversion of the Float value to String fails.

STRING_ID - Static variable in class

IntegerConverter
The message identifier of the FacesMessage to be created if the conversion of the Integer value to String fails.

STRING_ID - Static variable in class

LongConverter
The message identifier of the FacesMessage to be created if the conversion of the Long value to String fails.

STRING_ID - Static variable in class

NumberConverter
The message identifier of the FacesMessage to be created if the conversion of the Number value to String fails.

STRING_ID - Static variable in class

ShortConverter
The message identifier of the FacesMessage to be created if the conversion of the Short value to String fails.

StringHolder - Class in javax.xml.rpc.holders

StringHolder() - Constructor for class

javax.xml.rpc.holders.StringHolder

StringHolder(String) - Constructor for class

javax.xml.rpc.holders.StringHolder
**StringTerm** - Class in [javax.mail.search](https://docs.oracle.com/javaee/7/api/javax/mail/search/)
   This class implements the match method for Strings.

**StringTerm(String)** - Constructor for class
   javax.mail.search.StringTerm

**StringTerm(String, boolean)** - Constructor for class
   javax.mail.search.StringTerm

**Stub** - Interface in [javax.xml.rpc](https://docs.oracle.com/javase/8/docs/api/javax/xml/rpc/)
   The interface javax.xml.rpc.Stub is the common base interface for the stub classes.

**SubjectTerm** - Class in [javax.mail.search](https://docs.oracle.com/javaee/7/api/javax/mail/search/)
   This class implements comparisons for the Message Subject header.

**SubjectTerm(String)** - Constructor for class
   javax.mail.search.SubjectTerm
   Constructor.

**SUPPORT_DTD** - Static variable in class
   javax.xml.stream.XMLInputFactory
   The property that requires the parser to support DTDs

**supports(String)** - Method in interface
   javax.security.jacc.PolicyContextHandler
   This public method returns a boolean result indicating whether or not the handler supports the context object identified by the (case-sensitive) key value.

**supportsExecuteWithInputAndOutputRecord()** - Method in interface
   javax.resource.cci.ResourceAdapterMetaData
   Returns true if the implementation class for the Interaction interface implements public boolean execute(InteractionSpec ispec, Record input, Record output) method; otherwise the method returns false.

**supportsExecuteWithInputRecordOnly()** - Method in interface
   javax.resource.cci.ResourceAdapterMetaData
   Returns true if the implementation class for the Interaction interface implements public Record execute(InteractionSpec ispec, Record input) method; otherwise the method returns false.

**supportsLocalTransactionDemarcation()** - Method in interface
   javax.resource.cci.ResourceAdapterMetaData
   Returns true if the resource adapter implements the LocalTransaction interface and supports local transaction demarcation on the underlying EIS instance through the LocalTransaction interface.

**supportsResultSetType(int)** - Method in interface
   javax.resource.cci.ResultSetInfo
   Indicates whether or not a resource adapter supports a type of ResultSet.

**supportsResultSetTypeConcurrency(int, int)** - Method in interface
   javax.resource.cci.ResultSetInfo
Indicates whether or not a resource adapter supports the concurrency type in combination with the given ResultSet type/

suspend() - Method in interface
javax.transaction. TransactionManager
Suspend the transaction currently associated with the calling thread and return a Transaction object that represents the transaction context being suspended.

SYNC_RECEIVE - Static variable in interface
javax.resource.cci. InteractionSpec
The execution of an Interaction results in a synchronous receive of an output Record.

SYNC_SEND - Static variable in interface
javax.resource.cci. InteractionSpec
Interaction Verb type: The execution of an Interaction does only a send to the target EIS instance.

SYNC_SEND_RECEIVE - Static variable in interface
javax.resource.cci. InteractionSpec
Interaction Verb type: The execution of an Interaction sends a request to the EIS instance and receives response synchronously.

Synchronization - Interface in javax.transaction
The transaction manager supports a synchronization mechanism that allows the interested party to be notified before and after the transaction completes.

SystemException - Exception in javax.transaction
The SystemException is thrown by the transaction manager to indicate that it has encountered an unexpected error condition that prevents future transaction services from proceeding.

SystemException() - Constructor for exception
javax.transaction. SystemException

SystemException(String) - Constructor for exception
javax.transaction. SystemException
Create a SystemException with a given string.

SystemException(int) - Constructor for exception
javax.transaction. SystemException
Create a SystemException with a given error code.
Table - Annotation Type in `javax.persistence`  
This annotation specifies the primary table for the annotated entity.

TableGenerator - Annotation Type in `javax.persistence`  
This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the `GeneratedValue` annotation.

Tag - Interface in `javax.servlet.jsp.tagext`  
The interface of a classic tag handler that does not want to manipulate its body.

TagAdapter - Class in `javax.servlet.jsp.tagext`  
Wraps any SimpleTag and exposes it using a Tag interface.

TagAdapter(SimpleTag) - Constructor for class `javax.servlet.jsp.tagext.TagAdapter`  
Creates a new TagAdapter that wraps the given SimpleTag and returns the parent tag when getParent() is called.

TagAttributeInfo - Class in `javax.servlet.jsp.tagext`  
Information on the attributes of a Tag, available at translation time.

TagAttributeInfo(String, boolean, String, boolean) - Constructor for class `javax.servlet.jsp.tagext.TagAttributeInfo`  
Constructor for TagAttributeInfo.

TagAttributeInfo(String, boolean, String, boolean, boolean) - JSP 2.0 Constructor for TagAttributeInfo.

TagAttributeInfo(String, boolean, String, boolean, boolean, String, boolean, boolean, String, String) - JSP 2.1 Constructor for TagAttributeInfo.

TagData - Class in `javax.servlet.jsp.tagext`  
The (translation-time only) attribute/value information for a tag instance.

TagData(Object[][]) - Constructor for class `javax.servlet.jsp.tagext.TagData`  
Constructor for TagData.

TagData(Hashtable<String, Object>) - Constructor for class `javax.servlet.jsp.tagext.TagData`  
Constructor for a TagData.

TagExtraInfo - Class in `javax.servlet.jsp.tagext`  
Optional class provided by the tag library author to describe additional translation-time information not described in the TLD.

TagExtraInfo() - Constructor for class `javax.servlet.jsp.tagext.TagExtraInfo`  
Sole constructor.

TagFileInfo - Class in `javax.servlet.jsp.tagext`
Tag information for a tag file in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

**TagFileInfo(String, String, TagInfo)** - Constructor for class
java.servlet.jsp.tagext.TagFileInfo
Constructor for TagFileInfo from data in the JSP 2.0 format for TLD.

tagFiles - Variable in class
java.servlet.jsp.tagext.TagLibraryInfo
An array describing the tag files that are defined in this tag library.

**TagInfo** - Class in java.servlet.jsp.tagext
Tag information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

**TagInfo(String, String, String, TagLibraryInfo, TagExtraInfo, TagAttributeInfo[])** - Constructor for class
java.servlet.jsp.tagext.TagInfo
Constructor for TagInfo from data in the JSP 1.1 format for TLD.

**TagInfo(String, String, String, TagLibraryInfo, TagExtraInfo, TagAttributeInfo[], String, String, TagVariableInfo[])** - Constructor for class
java.servlet.jsp.tagext.TagInfo
Constructor for TagInfo from data in the JSP 1.2 format for TLD.

**TagInfo(String, String, String, TagLibraryInfo, TagExtraInfo, TagAttributeInfo[], boolean)** - Constructor for class
java.servlet.jsp.tagext.TagInfo
Constructor for TagInfo from data in the JSP 2.0 format for TLD.

**TagLibraryInfo** - Class in java.servlet.jsp.tagext
Translation-time information associated with a taglib directive, and its underlying TLD file.

**TagLibraryInfo(String, String)** - Constructor for class
java.servlet.jsp.tagext.TagLibraryInfo
Constructor.

**TagLibraryValidator** - Class in java.servlet.jsp.tagext
Translation-time validator class for a JSP page.

**TagLibraryValidator()** - Constructor for class
java.servlet.jsp.tagext.TagLibraryValidator
Sole constructor.

tags - Variable in class java.servlet.jsp.tagext.TagLibraryInfo
An array describing the tags that are defined in this tag library.

**TagSupport** - Class in java.servlet.jsp.tagext
A base class for defining new tag handlers implementing Tag.

**TagSupport()** - Constructor for class
java.servlet.jsp.tagext.TagSupport
Default constructor, all subclasses are required to define only a public constructor with the same signature, and to call the superclass constructor.

**TagVariableInfo** - Class in `javax.servlet.jsp.tagext`
Variable information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.

**TagVariableInfo(String, String, String, boolean, int)** - Constructor for class `javax.servlet.jsp.tagext.TagVariableInfo`
Constructor for TagVariableInfo.

**Target** - Interface in `javax.enterprise.deploy.spi`
A Target interface represents a single logical core server of one instance of a J2EE platform product.

**TargetException** - Exception in `javax.enterprise.deploy.spi.exceptions`
This exception is to report bad target designators.

**TargetException(String)** - Constructor for exception `javax.enterprise.deploy.spi.exceptions.TargetException`
Creates an new TargetException object.

**TargetModuleID** - Interface in `javax.enterprise.deploy.spi`
A TargetModuleID interface represents a unique identifier for a deployed application module.

**TELEPHONE_NUMBER** - Static variable in interface `javax.xml.registry.LifeCycleManager`

**TelephoneNumber** - Interface in `javax.xml.registry.infomodel`
A simple re-usable entity class that defines attributes of a telephone number.

**Temporal** - Annotation Type in `javax.persistence`
This annotation must be specified for persistent fields or properties of type `Date` and `Calendar`.

**TemporalType** - Enum in `javax.persistence`
Type used to indicate a specific mapping of `Date` or `Calendar`.

**TemporaryQueue** - Interface in `javax.jms`
A TemporaryQueue object is a unique Queue object created for the duration of a Connection.

**TemporaryTopic** - Interface in `javax.jms`
A TemporaryTopic object is a unique Topic object created for the duration of a Connection.

**term** - Variable in class `javax.mail.search.NotTerm`
The search term to negate.

**terms** - Variable in class `javax.mail.search.AndTerm`
The array of terms on which the AND operator should be applied.

**terms** - Variable in class `javax.mail.search.OrTerm`
The array of terms on which the OR operator should be applied.

**Text** - Interface in `javax.xml.soap`
A representation of a node whose value is text.

**TextMessage** - Interface in `javax.jms`
A TextMessage object is used to send a message containing a java.lang.String.
**TIME_ID** - Static variable in class `javax.faces.convert.DateTimeConverter`
The message identifier of the `FacesMessage` to be created if the conversion to Time fails.

**TimedObject** - Interface in `javax.ejb`
The TimedObject interface contains the callback method that is used to deliver timer expiration notifications.

**Timeout** - Annotation Type in `javax.ejb`
Designates a method on a stateless session bean class or message driven bean class that should receive EJB timer expirations for that bean.

**Timer** - Interface in `javax.ejb`
The Timer interface contains information about a timer that was created through the EJB Timer Service.

**TimerHandle** - Interface in `javax.ejb`
The TimerHandle interface is implemented by all EJB timer handles.

**TimerService** - Interface in `javax.ejb`
The TimerService interface provides enterprise bean components with access to the container-provided Timer Service.

**TimeStatistic** - Interface in `javax.management.j2ee.statistics`
Specifies standard timing measurements.

**tlibversion** - Variable in class `javax.servlet.jsp.tagext.TagLibraryInfo`
The version of the tag library.

**TMENDRSCAN** - Static variable in interface `javax.transaction.xa.XAResource`
Ends a recovery scan.

**TMFAIL** - Static variable in interface `javax.transaction.xa.XAResource`
Disassociates the caller and marks the transaction branch rollback-only.

**TMJOIN** - Static variable in interface `javax.transaction.xa.XAResource`
Caller is joining existing transaction branch.

**TMNOFLAGS** - Static variable in interface `javax.transaction.xa.XAResource`
Use TMNOFLAGS to indicate no flags value is selected.

**TMONEPHASE** - Static variable in interface `javax.transaction.xa.XAResource`
Caller is using one-phase optimization.

**TMRESUME** - Static variable in interface `javax.transaction.xa.XAResource`
Caller is resuming association with a suspended transaction branch.

**TMSTARTRSCAN** - Static variable in interface `javax.transaction.xa.XAResource`
Starts a recovery scan.

**TMSUCCESS** - Static variable in interface `javax.transaction.xa.XAResource`
Disassociates caller from a transaction branch.

**TMSUSPEND** - Static variable in interface 
javax.transaction.xa.XAResource
Caller is suspending (not ending) its association with a transaction branch.

**TO** - Static variable in class javax.mail.Message.RecipientType
The "To" (primary) recipients.

**Topic** - Interface in javax.jms
A Topic object encapsulates a provider-specific topic name.

**TopicConnection** - Interface in javax.jms
A TopicConnection object is an active connection to a publish/subscribe JMS provider.

**TopicConnectionFactory** - Interface in javax.jms
A client uses a TopicConnectionFactory object to create TopicConnection objects with a publish/subscribe JMS provider.

**TopicPublisher** - Interface in javax.jms
A client uses a TopicPublisher object to publish messages on a topic.

**TopicRequestor** - Class in javax.jms
The TopicRequestor helper class simplifies making service requests.

**TopicRequestor(TopicSession, Topic)** - Constructor for class javax.jms.TopicRequestor
Constructor for the TopicRequestor class.

**TopicSession** - Interface in javax.jms
A TopicSession object provides methods for creating TopicPublisher, TopicSubscriber, and TemporaryTopic objects.

**TopicSubscriber** - Interface in javax.jms
A client uses a TopicSubscriber object to receive messages that have been published to a topic.

**toString()** - Method in class javax.activation.MimeType
Return the String representation of this object.

**toString()** - Method in class javax.activation.MimeTypeParameterList
Return a string representation of this object.

**toString()** - Method in class 
javax.enterprise.deploy.shared.ActionType
Return the string name of this ActionType or the integer value if outside the bounds of the table

**toString()** - Method in class 
javax.enterprise.deploy.shared.CommandType
Return the string name of this CommandType or the integer value if outside the bounds of the table

**toString()** - Method in class 
javax.enterprise.deploy.shared.DConfigBeanVersionType
Return the string name of this DConfigBeanVersionType or the integer value if outside the bounds of the table

**toString()** - Method in class 
javax.enterprise.deploy.shared.ModuleType
Return the string name of this ModuleType or the integer value if outside the bounds of the table
**toString()** - Method in class `javax.enterprise.deploy.shared.StateType`  
Returns the string name of this `StateType` or the integer value  
if outside the bounds of the table.

**toString()** - Method in interface `javax.enterprise.deploy.spi.TargetModuleID`  
Retrieve the identifier representing the deployed module.

**toString()** - Method in class `javax.faces.application.FacesMessage.Severity`  
Return a String representation of this `FacesMessage.Severity`  
instance.

**toString()** - Method in class `javax.faces.event.PhaseId`  
Return a String representation of this `PhaseId` instance.

**toString()** - Method in interface `javax.jms.Queue`  
Returns a string representation of this object.

**toString()** - Method in interface `javax.jms.Topic`  
Returns a string representation of this object.

**toString()** - Method in class `javax.mail.Address`  
Return a String representation of this address object.

**toString()** - Method in class `javax.mail.Folder`  
override the default `toString()`, it will return the String from Folder.getFullName() or if that is null, it will use the  
default `toString()` behavior.

**toString()** - Method in exception `javax.mail.internet.AddressException`  

**toString()** - Method in class `javax.mail.internet.ContentDisposition`  
Retrieve a RFC2045 style string representation of this  
`ContentDisposition`.

**toString()** - Method in class `javax.mail.internet.ContentType`  
Retrieve a RFC2045 style string representation of this `ContentType`.

**toString()** - Method in class `javax.mail.internet.NeWSAddress`  
Convert this address into a RFC 822 / RFC 2047 encoded address.

**toString(Address[])** - Static method in class  
`javax.mail.internet.NeWSAddress`  
Convert the given array of `NeWSAddress` objects into a comma  
separated sequence of address strings.

**toString(Address[], int)** - Static method in class  
`javax.mail.internet.NeWSAddress`  
Convert the given array of `NeWSAddress` objects into a comma  
separated sequence of address strings.

**toString()** - Method in class `javax.mail.internet.NeWSAddress`  
Convert this address into a RFC 1036 address.

**toString(Address[])** - Static method in class  
`javax.mail.internet.NeWSAddress`  
Convert the given array of `NeWSAddress` objects into a comma  
separated sequence of address strings.

**toString()** - Method in class `javax.mail.internet.ParameterList`  
Convert this ParameterList into a MIME String.
**toString(int)** - Method in class `javax.mail.internet.ParameterList`
Convert this ParameterList into a MIME String.

**toString()** - Method in class `javax.mail.Message.RecipientType`

**toString()** - Method in exception `javax.mail.MessagingException`
Override toString method to provide information on nested exceptions.

**toString()** - Method in class `javax.mail.Provider`
Overrides Object.toString()

**toString()** - Method in class `javax.mail.Provider.Type`

**toString()** - Method in class `javax.mail.Service`
Return getURLName.toString() if this service has a URLName, otherwise it will return the default toString.

**toString()** - Method in class `javax.mail.URLName`
Constructs a string representation of this URLName.

**toString()** - Method in class `javax.servlet.jsp.tagext.TagAttributeInfo`
Returns a String representation of this TagAttributeInfo, suitable for debugging purposes.

**toString()** - Method in class `javax.xml.bind.helpers.ValidationEventImpl`
Returns a string representation of this object in a format helpful to debugging.

**toString()** - Method in class `javax.xml.bind.helpers.ValidationEventLocatorImpl`
Returns a string representation of this object in a format helpful to debugging.

**toString()** - Method in exception `javax.xml.bind.JAXBException`
Returns a short description of this JAXBException.

**toString()** - Method in exception `javax.xml.bind.TypeConstraintException`
Returns a short description of this TypeConstraintException.

**toString()** - Method in interface `javax.xml.registry.Query`
Returns the String representing the query.

**toString()** - Method in class `javax.xml.rpc.ParameterMode`
Returns a String describing this ParameterMode object.

**toUnicodeString()** - Method in class `javax.mail.internet.InternetAddress`
Returns a properly formatted address (RFC 822 syntax) of Unicode characters.

**toXML()** - Method in interface `javax.xml.registry.infomodel.RegistryObject`
Returns a registry provider specific XML representation of this Object.

**Transaction** - Interface in `javax.transaction`
The Transaction interface allows operations to be performed against the transaction in the target Transaction object.

**TransactionAttribute** - Annotation Type in `javax.ejb`
When applied at the TYPE-level, designates the default
transaction attribute for all business methods of the session or message driven bean.

**TransactionAttributeType** - Enum in `javax.ejb`

**TransactionInProgressException** - Exception in `javax.jms`
This exception is thrown when an operation is invalid because a transaction is in progress.

**TransactionInProgressException(String, String)** - Constructor for exception `javax.jms.TransactionInProgressException`
Constructs a TransactionInProgressException with the specified reason and error code.

**TransactionInProgressException(String)** - Constructor for exception `javax.jms.TransactionInProgressException`
Constructs a TransactionInProgressException with the specified reason.

**TransactionManagement** - Annotation Type in `javax.ejb`
Declares whether a session bean or message driven bean has container managed transactions or bean managed transactions.

**TransactionManagementType** - Enum in `javax.ejb`

**TransactionManager** - Interface in `javax.transaction`
The TransactionManager interface defines the methods that allow an application server to manage transaction boundaries.

**TransactionRequiredException** - Exception in `javax.persistence`
Thrown by the persistence provider when a transaction is required but is not active.

**TransactionRequiredException()** - Constructor for exception `javax.persistence.TransactionRequiredException`
Constructs a new TransactionRequiredException exception with null as its detail message.

**TransactionRequiredException(String)** - Constructor for exception `javax.persistence.TransactionRequiredException`
Constructs a new TransactionRequiredException exception with the specified detail message.

**TransactionRequiredException** - Exception in `javax.transaction`
This exception indicates that a request carried a null transaction context, but the target object requires an activate transaction.

**TransactionRequiredException()** - Constructor for exception `javax.transaction.TransactionRequiredException`

**TransactionRequiredException(String)** - Constructor for exception `javax.transaction.TransactionRequiredException`

**TransactionRequiredLocalException** - Exception in `javax.ejb`
This exception indicates that a request carried a null transaction context, but the target object requires an active transaction.

**TransactionRequiredLocalException()** - Constructor for exception `javax.ejb.TransactionRequiredLocalException`
Constructs a TransactionRequiredLocalException with no detail message.

**TransactionRequiredLocalException(String)** - Constructor for exception javax.ejb.TransactionRequiredLocalException

Constructs an TransactionRequiredLocalException with the specified detailed message.

**TransactionRolledBackException** - Exception in javax.jms

This exception must be thrown when a call to Session.commit results in a rollback of the current transaction.

**TransactionRolledBackException(String, String)** - Constructor for exception javax.jms.TransactionRolledBackException

Constructs a TransactionRolledBackException with the specified reason and error code.

**TransactionRolledBackException(String)** - Constructor for exception javax.jms.TransactionRolledBackException

Constructs a TransactionRolledBackException with the specified reason.

**TransactionRolledbackException** - Exception in javax.transaction

This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back.

**TransactionRolledbackException()** - Constructor for exception javax.transaction.TransactionRolledbackException

**TransactionRolledbackException(String)** - Constructor for exception javax.transaction.TransactionRolledbackException

**TransactionRolledbackLocalException** - Exception in javax.ejb

This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back.

**TransactionRolledbackLocalException()** - Constructor for exception javax.ejb.TransactionRolledbackLocalException

Constructs a TransactionRolledbackLocalException with no detail message.

**TransactionRolledbackLocalException(String)** - Constructor for exception javax.ejb.TransactionRolledbackLocalException

Constructs a TransactionRolledbackLocalException with the specified detailed message.

**TransactionRolledbackLocalException(String, Exception)** - Constructor for exception javax.ejb.TransactionRolledbackLocalException

Constructs a TransactionRolledbackLocalException with the specified detail message and a nested exception.

**TransactionSynchronizationRegistry** - Interface in javax.transaction

This interface is intended for use by system level application server components such as persistence managers, resource adapters, as well as EJB and Web application components.

**transform(ClassLoader, String, Class<?>, ProtectionDomain, byte[])** - Method in interface javax.persistence.spi.ClassTransformer
Invoked when a class is being loaded or redefined.

**Transient** - Annotation Type in `javax.persistence`
This annotation specifies that the property or field is not persistent.

**TRANSPORT** - Static variable in class `javax.mail.Provider.Type`

**Transport** - Class in `javax.mail`
An abstract class that models a message transport.

**Transport(Session, URLName)** - Constructor for class `javax.mail.Transport`
Constructor.

**TransportAdapter** - Class in `javax.mail.event`
The adapter which receives Transport events.

**TransportAdapter()** - Constructor for class `javax.mail.event.TransportAdapter`

**TransportEvent** - Class in `javax.mail.event`
This class models Transport events.

**TransportEvent(Transport, int, Address[], Address[], Address[], Message)** - Constructor for class `javax.mail.event.TransportEvent`
Constructor.

**TransportListener** - Interface in `javax.mail.event`
This is the listener interface for Transport events

**TryCatchFinally** - Interface in `javax.servlet.jsp.tagext`
The auxiliary interface of a Tag, IterationTag or BodyTag tag handler that wants additional hooks for managing resources.

**TX_CONCURRENT_WORK_DISALLOWED** - Static variable in exception `javax.resource.spi.work.WorkException`
Indicates that concurrent work within a transaction is disallowed.

**TX_RECREATE_FAILED** - Static variable in exception `javax.resource.spi.work.WorkException`
Indicates a failure in recreating the specified transaction context.

**TYPE** - Static variable in class `javax.el.ELResolver`
The attribute name of the named attribute in the FeatureDescriptor that specifies the runtime type of the variable or property.

**type** - Variable in class `javax.mail.event.ConnectionEvent`
The event type.

**type** - Variable in class `javax.mail.event.FolderEvent`
The event type.

**type** - Variable in class `javax.mail.event.MessageChangedEvent`
The event type.

**type** - Variable in class `javax.mail.event.MessageCountEvent`
The event type.

**type** - Variable in class `javax.mail.event.StoreEvent`
The event type.

**type** - Variable in class `javax.mail.event.TransportEvent`
The event type.
**type** - Variable in class `javax.mail.Message.RecipientType`
The type of recipient, usually the name of a corresponding
Internet standard header.

**type** - Variable in class `javax.mail.search.RecipientTerm`
The recipient type.

**TYPE_MESSAGE_ID** - Static variable in class
`javax.faces.validator.DoubleRangeValidator`
The message identifier of the `FacesMessage` to be created if the
current value of this component is not of the correct type.

**TYPE_MESSAGE_ID** - Static variable in class
`javax.faces.validator.LongRangeValidator`
The message identifier of the `FacesMessage` to be created if the
current value of this component is not of the correct type.

**TypeConstraintException** - Exception in `javax.xml.bind`
This exception indicates that a violation of a dynamically
checked type constraint was detected.

**TypeConstraintException(String)** - Constructor for exception
`javax.xml.bind.TypeConstraintException`
Construct a `TypeConstraintException` with the specified detail
message.

**TypeConstraintException(String, String)** - Constructor for exception
`javax.xml.bind.TypeConstraintException`
Construct a `TypeConstraintException` with the specified detail
message and vendor specific errorCode.

**TypeConstraintException(Throwable)** - Constructor for exception
`javax.xml.bind.TypeConstraintException`
Construct a `TypeConstraintException` with a linkedException.

**TypeConstraintException(String, Throwable)** - Constructor for exception
`javax.xml.bind.TypeConstraintException`
Construct a `TypeConstraintException` with the specified detail
message and linkedException.

**TypeConstraintException(String, String, Throwable)** - Constructor
for exception `javax.xml.bind.TypeConstraintException`
Construct a `TypeConstraintException` with the specified detail
message, vendor specific errorCode, and linkedException.

**TypeMapping** - Interface in `javax.xml.rpc.encoding`
The `javax.xml.rpc.encoding.TypeMapping` is the base interface
for the representation of a type mapping.

**TypeMappingRegistry** - Interface in `javax.xml.rpc.encoding`
The interface `javax.xml.rpc.encoding.TypeMappingRegistry`
defines a registry of `TypeMapping` instances for various
encoding styles.
UIColumn - Class in javax.faces.component
UIColumn is a UIComponent that represents a single column of
data within a parent UIData component.
UIColumn() - Constructor for class javax.faces.component.UIColumn
Create a new UIColumn instance with default property values.

UICommand - Class in javax.faces.component
UICommand is a UIComponent that represents a user interface
component which, when activated by the user, triggers an
application specific "command" or "action".
UICommand() - Constructor for class javax.faces.component.UICommand
Create a new UICommand instance with default property values.

UIComponent - Class in javax.faces.component
UIComponent is the base class for all user interface components
in JavaServer Faces.
UIComponent() - Constructor for class
javax.faces.component.UIComponent

UIComponentBase - Class in javax.faces.component
UIComponentBase is a convenience base class that implements the
default concrete behavior of all methods defined by
UIComponent.
UIComponentBase() - Constructor for class
javax.faces.component.UIComponentBase

UIComponentBodyTag - Class in javax.faces.webapp
Deprecated. All component tags now implement BodyTag. This
class has been replaced by UIComponentELTag.
UIComponentBodyTag() - Constructor for class
javax.faces.webapp.UIComponentBodyTag

UIComponentClassicTagBase - Class in javax.faces.webapp
Deprecated. UIComponentClassicTagBase is the base class for all JSP tags that use
the "classic" JSP tag interface that correspond to a
UIComponent instance in the view.
UIComponentClassicTagBase() - Constructor for class
javax.faces.webapp.UIComponentClassicTagBase

UIComponentELTag - Class in javax.faces.webapp
UIComponentELTag specializes its superclass to allow for
properties that take their values from EL API expressions.
UIComponentELTag() - Constructor for class
javax.faces.webapp.UIComponentELTag

UIComponentTag - Class in javax.faces.webapp
Deprecated. Use of this class has been replaced with
UIComponentELTag, which extends UIComponentClassicTagBase to
add properties that use the EL API introduced as part of JSP 2.1.

**UIComponentTag()** - Constructor for class `javax.faces.webapp.UIComponentTag`

*Deprecated.*

**UIComponentTagBase** - Class in `javax.faces.webapp`

*UIComponentTagBase* is the base class for all JSP tags that correspond to a `UIComponent` instance in the view.

**UIComponentTagBase()** - Constructor for class `javax.faces.webapp.UIComponentTagBase`

**UID** - Static variable in class `javax.mail.UIDFolder.FetchProfileItem`

UID is a fetch profile item that can be included in a FetchProfile during a fetch request to a Folder.

**UIData** - Class in `javax.faces.component`

*UIData* is a `UIComponent` that supports data binding to a collection of data objects represented by a `DataModel` instance, which is the current value of this component itself (typically established via a `ValueExpression`).

**UIData()** - Constructor for class `javax.faces.component.UIData`

Create a new `UIData` instance with default property values.

**UIDFolder** - Interface in `javax.mail`

The UIDFolder interface is implemented by Folders that can support the "disconnected" mode of operation, by providing unique-ids for messages in the folder.

**UIDFolder.FetchProfileItem** - Class in `javax.mail`

A fetch profile item for fetching UIDs.

**UIDFolder.FetchProfileItem(String)** - Constructor for class `javax.mail.UIDFolder.FetchProfileItem`

**UIForm** - Class in `javax.faces.component`

*UIForm* is a `UIComponent` that represents an input form to be presented to the user, and whose child components represent (among other things) the input fields to be included when the form is submitted.

**UIForm()** - Constructor for class `javax.faces.component.UIForm`

Create a new `UIForm` instance with default property values.

**UIGraphic** - Class in `javax.faces.component`

*UIGraphic* is a `UIComponent` that displays a graphical image to the user.

**UIGraphic()** - Constructor for class `javax.faces.component.UIGraphic`

Create a new `UIGraphic` instance with default property values.

**UIInput** - Class in `javax.faces.component`

*UIInput* is a `UIComponent` that represents a component that both displays output to the user (like `UIOutput` components do) and processes request parameters on the subsequent request that need to be decoded.

**UIInput()** - Constructor for class `javax.faces.component.UIInput`

Create a new `UIInput` instance with default property values.
**UIMessage** - Class in `javax.faces.component`
This component is responsible for displaying messages for a specific `UIComponent`, identified by a clientId.

**UIMessage()** - Constructor for class `javax.faces.component.UIMessage`
Create a new `UIMessage` instance with default property values.

**UIMessages** - Class in `javax.faces.component`
The renderer for this component is responsible for obtaining the messages from the `FacesContext` and displaying them to the user.

**UIMessages()** - Constructor for class `javax.faces.component.UIMessages`
Create a new `UIMessages` instance with default property values.

**UINamingContainer** - Class in `javax.faces.component`
`UINamingContainer` is a convenience base class for components that wish to implement `NamingContainer` functionality.

**UINamingContainer()** - Constructor for class `javax.faces.component.UINamingContainer`
Create a new `UINamingContainer` instance with default property values.

**UIOutput** - Class in `javax.faces.component`
`UIOutput` is a `UIComponent` that has a value, optionally retrieved from a model tier bean via a value expression, that is displayed to the user.

**UIOutput()** - Constructor for class `javax.faces.component.UIOutput`
Create a new `UIOutput` instance with default property values.

**UIPanel** - Class in `javax.faces.component`
`UIPanel` is a `UIComponent` that manages the layout of its child components.

**UIPanel()** - Constructor for class `javax.faces.component.UIPanel`
Create a new `UIPanel` instance with default property values.

**UIParameter** - Class in `javax.faces.component`
`UIParameter` is a `UIComponent` that represents an optionally named configuration parameter for a parent component.

**UIParameter()** - Constructor for class `javax.faces.component.UIParameter`
Create a new `UIParameter` instance with default property values.

**UISelectBoolean** - Class in `javax.faces.component`
`UISelectBoolean` is a `UIComponent` that represents a single boolean (true or false) value.

**UISelectBoolean()** - Constructor for class `javax.faces.component.UISelectBoolean`
Create a new `UISelectBoolean` instance with default property values.

**UISelectItem** - Class in `javax.faces.component`
`UISelectItem` is a component that may be nested inside a `UISelectMany` or `UISelectOne` component, and causes the addition of a `SelectItem` instance to the list of available options for the parent component.

**UISelectItem()** - Constructor for class `javax.faces.component.UISelectItem`
Create a new **UISelectItem** instance with default property values.

**UISelectItems** - Class in `javax.faces.component`

*UISelectItems* is a component that may be nested inside a **UISelectMany** or **UISelectOne** component, and causes the addition of one or more SelectItem instances to the list of available options in the parent component.

**UISelectItems()** - Constructor for class

`javax.faces.component.UISelectItems`

Create a new **UISelectItems** instance with default property values.

**UISelectMany** - Class in `javax.faces.component`

*UISelectMany* is a **UIComponent** that represents the user's choice of a zero or more items from among a discrete set of available options.

**UISelectMany()** - Constructor for class

`javax.faces.component.UISelectMany`

Create a new **UISelectMany** instance with default property values.

**UISelectOne** - Class in `javax.faces.component`

*UISelectOne* is a **UIComponent** that represents the user's choice of zero or one items from among a discrete set of available options.

**UISelectOne()** - Constructor for class

`javax.faces.component.UISelectOne`

Create a new **UISelectOne** instance with default property values.

**UIViewRoot** - Class in `javax.faces.component`

*UIViewRoot* is the **UIComponent** that represents the root of the **UIComponent** tree.

**UIViewRoot()** - Constructor for class

`javax.faces.component.UIViewRoot`

Create a new **UIViewRoot** instance with default property values.

**UnavailableException** - Exception in `javax.resource.spi`

This is thrown to indicate that a service is unavailable.

**UnavailableException()** - Constructor for exception

`javax.resource.spi.UnavailableException`

Constructs a new instance with null as its detail message.

**UnavailableException(String)** - Constructor for exception

`javax.resource.spi.UnavailableException`

Constructs a new instance with the specified detail message.

**UnavailableException(Throweroble)** - Constructor for exception

`javax.resource.spi.UnavailableException`

Constructs a new throwable with the specified cause.

**UnavailableException(String, Throwable)** - Constructor for exception

`javax.resource.spi.UnavailableException`

Constructs a new throwable with the specified detail message and cause.

**UnavailableException(String, String)** - Constructor for exception

`javax.resource.spi.UnavailableException`

Constructs a new throwable with the specified detail message
and an error code.

**UnavailableException** - Exception in javax.servlet

Defines an exception that a servlet or filter throws to indicate that it is permanently or temporarily unavailable.

**UnavailableException(Servlet, String)** - Constructor for exception javax.servlet.UnavailableException

*Deprecated.* As of Java Servlet API 2.2, use **UnavailableException.UnavailableException(String)** instead.

**UnavailableException(int, Servlet, String)** - Constructor for exception javax.servlet.UnavailableException

*Deprecated.* As of Java Servlet API 2.2, use **UnavailableException.UnavailableException(String, int)** instead.

**UnavailableException(String)** - Constructor for exception javax.servlet.UnavailableException

Constructs a new exception with a descriptive message indicating that the servlet is permanently unavailable.

**UnavailableException(String, int)** - Constructor for exception javax.servlet.UnavailableException

Constructs a new exception with a descriptive message indicating that the servlet is temporarily unavailable and giving an estimate of how long it will be unavailable.

**UNBOUNDED BUFFER** - Static variable in class javaservlet.jsp.JspWriter

Constant indicating that the Writer is buffered and is unbounded; this is used in BodyContent.


Undoes a previous confirmation of this Association by the User associated with the caller.

**UNDEFINED** - Static variable in exception javax.resource.spi.work.WorkException

Undefined error code.

**UNDEPLOY** - Static variable in class javaservlet.enterprise.deploy.shared.CommandType

The DeploymentManager action operation being processed is undeploy.

**undeploy(TargetModuleID[])** - Method in interface javaservlet.enterprise.deploy.spi.DeploymentManager

Remove the application from the target server.

**unDeprecateObjects(Collection)** - Method in interface javax.xml.registry.LifeCycleManager

Undeprecates one or more previously deprecated objects.

**UnexpectedObjectException** - Exception in javax.xml.registry

This exception is thrown when the JAXR provider finds a Object that is out-of-place or of the wrong type within the context of a user request.

**UnexpectedObjectException()** - Constructor for exception javax.xml.registry.UnexpectedObjectException

Constructs a JAXRException object with no reason or embedded Throwable.
**UnexpectedObjectException(String)** - Constructor for exception
javax.xml.registry.**UnexpectedObjectException**
Constructs a JAXRException object with the given String as the reason for the exception being thrown.

**UnexpectedObjectException(String, Throwable)** - Constructor for exception
javax.xml.registry.**UnexpectedObjectException**
Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

**UnexpectedObjectException(Throwables)** - Constructor for exception
javax.xml.registry.**UnexpectedObjectException**
Constructs a JAXRException object initialized with the given Throwable object.

**unfold(String)** - Static method in class
javax.mail.internet.**MimeUtility**
Unfold a folded header.

**UNIQUE_ID_PREFIX** - Static variable in class
javax.faces.component.**UIViewRoot**
The prefix that will be used for identifiers generated by the createUniqueId() method.

**UNIQUE_ID_PREFIX** - Static variable in class
javax.faces.webapp.**UIComponentClassicTagBase**
Used as the prefix for ids.

**UniqueConstraint** - Annotation Type in **javax.persistence**
This annotation is used to specify that a unique constraint is to be included in the generated DDL for a primary or secondary table.

**UNKNOWN** - Static variable in interface
javax.resource.spi.work.**WorkManager**
A constant to indicate an unknown start delay duration or other unknown values.

**unmarshal(String)** - Method in class
javax.xml.bind.annotation.adapters.**CollapsedStringAdapter**
Removes leading and trailing whitespaces of the string given as the parameter, then truncate any sequence of tab, CR, LF, and SP by a single whitespace character '\'.

**unmarshal(String)** - Method in class
javax.xml.bind.annotation.adapters.**HexBinaryAdapter**

**unmarshal(String)** - Method in class
javax.xml.bind.annotation.adapters.**NormalizedStringAdapter**
Removes leading and trailing whitespaces of the string given as the parameter, then replace any tab, CR, and LF by a whitespace character '\'.

**unmarshal(ValueType)** - Method in class
javax.xml.bind.annotation.adapters.**XmlAdapter**
Convert a value type to a bound type.

**unmarshal(XMLOutput)** - Method in class javax.xml.bind.**Binder**
Unmarshal XML infoset view to a JAXB object tree.

**unmarshal(XMLOutput, Class<T>)** - Method in class
javax.xml.bind.**Binder**  
Unmarshal XML root element by provided declaredType to a JAXB object tree.

**unmarshal(Source)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(XMLReader, InputSource)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**
    Unmarshals an object by using the specified XMLReader and the InputSource.

**unmarshal(InputSource)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(URL)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(File)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(InputStream)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(Reader)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(XMLEventReader)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(XMLStreamReader)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(Node, Class<T>)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(Source, Class<T>)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(XMLStreamReader, Class<T>)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(XMLEventReader, Class<T>)** - Method in class  
javax.xml.bind.helpers.**AbstractUnmarshallerImpl**

**unmarshal(File)** - Method in interface javax.xml.bind.**Unmarshaller**  
Unmarshal XML data from the specified file and return the resulting content tree.

**unmarshal(InputStream)** - Method in interface  
javax.xml.bind.**Unmarshaller**  
Unmarshal XML data from the specified InputStream and return the resulting content tree.
unmarshal(Reader) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified Reader and return the resulting content tree.

unmarshal(URL) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified URL and return the resulting content tree.

unmarshal(InputSource) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified SAX InputSource and return the resulting content tree.

unmarshal(Node) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal global XML data from the specified DOM tree and return the resulting content tree.

unmarshal(Node, Class<T>) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data by JAXB mapped declaredType and return the resulting content tree.

unmarshal(Source) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified XML Source and return the resulting content tree.

unmarshal(Source, Class<T>) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified XML Source by declaredType and return the resulting content tree.

unmarshal(XMLStreamReader) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal XML data from the specified pull parser and return the resulting content tree.

unmarshal(XMLStreamReader, Class<T>) - Method in interface javax.xml.bind.Unmarshaller
Unmarshal root element to JAXB mapped declaredType and return the resulting content tree.

UnmarshalException - Exception in javax.xml.bind
This exception indicates that an error has occurred while performing an unmarshal operation that prevents the JAXB Provider from completing the operation.

UnmarshalException(String) - Constructor for exception javax.xml.bind.UnmarshalException
Construct an UnmarshalException with the specified detail message.

UnmarshalException(String, String) - Constructor for exception javax.xml.bind.UnmarshalException
Construct an UnmarshalException with the specified detail message and vendor specific errorCode.

**UnmarshalException(Throwable)** - Constructor for exception
```
javax.xml.bind.UnmarshalException
```
Construct an UnmarshalException with a linkedException.

**UnmarshalException(String, Throwable)** - Constructor for exception
```
javax.xml.bind.UnmarshalException
```
Construct an UnmarshalException with the specified detail message and linkedException.

**UnmarshalException(String, String, Throwable)** - Constructor for exception
```
javax.xml.bind.UnmarshalException
```
Construct an UnmarshalException with the specified detail message, vendor specific errorCode, and linkedException.

**Unmarshaller** - Interface in **javax.xml.bind**
The Unmarshaller class governs the process of deserializing XML data into newly created Java content trees, optionally validating the XML data as it is unmarshalled.

**Unmarshaller.Listener** - Class in **javax.xml.bind**
Register an instance of an implementation of this class with Unmarshaller to externally listen for unmarshal events.

**Unmarshaller.Listener()** - Constructor for class
```
javax.xml.bind.Unmarshaller.Listener
```

**UnmarshallerEventHandler** - Interface in **javax.xml.bind**
Unmarshaller implemented as SAX ContentHandler.

**unregisterTypeMapping(String)** - Method in interface
```
javax.xml.rpc.encoding.TypeMappingRegistry
```
Unregisters a TypeMapping instance, if present, from the specified encodingStyleURI.

**unsetEntityContext()** - Method in interface **javax.ejb.EntityBean**
Unset the associated entity context.

**unsubscribe(String)** - Method in interface **javax.jms.Session**
Unsubscribes a durable subscription that has been created by a client.

**unsubscribe(String)** - Method in interface **javax.jms.TopicSession**
Unsubscribes a durable subscription that has been created by a client.

**UnsupportedCapabilityException** - Exception in **javax.xml.registry**
This exception must be thrown when a JAXR client attempts to invoke an API method that is not supported by the capability profile that is supported by the JAXR provider.

**UnsupportedCapabilityException()** - Constructor for exception
```
javax.xml.registry.UnsupportedCapabilityException
```
Constructs a JAXRException object with no reason or embedded Throwable.

**UnsupportedCapabilityException(String)** - Constructor for exception
```
javax.xml.registry.UnsupportedCapabilityException
```
Constructs a JAXRException object with the given String as the reason for the exception being thrown.
UnsupportedCapabilityException(String, Throwable) - Constructor for exception javax.xml.registry.UnsupportedCapabilityException
    Constructs a JAXRException object with the given String as the reason for the exception being thrown and the given Throwable object as an embedded Throwable.

UnsupportedCapabilityException(Throwable) - Constructor for exception javax.xml.registry.UnsupportedCapabilityException
    Constructs a JAXRException object initialized with the given Throwable object.

UnsupportedDataTypeException - Exception in javax.activation
    Signals that the requested operation does not support the requested data type.

UnsupportedDataTypeException() - Constructor for exception javax.activation.UnsupportedDataTypeException
    Constructs an UnsupportedDataTypeException with no detail message.

UnsupportedDataTypeException(String) - Constructor for exception javax.activation.UnsupportedDataTypeException
    Constructs an UnsupportedDataTypeException with the specified message.

UPDATE_MESSAGE_ID - Static variable in class javax.faces.component.UIInput
    The message identifier of the FacesMessage to be created if a model update error occurs, and the thrown exception has no message.

UPDATE_MODEL_VALUES - Static variable in class javax.faces.event.PhaseId
    Identifier that indicates an interest in events queued for the Update Model Values phase of the request processing lifecycle.

updateHeaders() - Method in class javax.mail.internet.MimeBodyPart
    Examine the content of this body part and update the appropriate MIME headers.

updateHeaders() - Method in class javax.mail.internet.MimeMessage
    Called by the saveChanges method to actually update the MIME headers.

updateHeaders() - Method in class javax.mail.internet.MimeMultipart
    Update headers.

updateHeaders() - Method in class javax.mail.internet.PreencodedMimeBodyPart
    Force the Content-Transfer-Encoding header to use the encoding that was specified when this object was created.

updateJAXB(XmlNode) - Method in class javax.xml.bind.Binder
    Takes an XML node and updates its associated JAXB object and its descendants.

updateMessageID() - Method in class javax.mail.internet.MimeMessage
    Update the Message-ID header.

updateModel(FacesContext) - Method in class javax.faces.component.UIInput
    Perform the following algorithm to update the model data associated with this UIInput, if any, as appropriate.
**updatesAreDetected(int)** - Method in interface javax.resource.cci.ResultSetInfo
Indicates whether or not a visible row update can be detected by calling the method ResultSet.rowUpdated.

**updateXML(Object)** - Method in class javax.xml.bind.Binder
Takes an JAXB object and updates its associated XML node and its descendants.

**updateXML(Object, XmlNode)** - Method in class javax.xml.bind.Binder
Changes in JAXB object tree are updated in its associated XML parse tree.

**uri** - Variable in class javax.servlet.jsp.tagext.TagLibraryInfo
The value of the uri attribute from the taglib directive for this library.

**URI NS SOAP 1 1 ENVELOPE** - Static variable in interface javax.xml.soap.SOAPConstants
The namespace identifier for the SOAP 1.1 envelope.

**URI NS SOAP 1 2 ENCODING** - Static variable in interface javax.xml.soap.SOAPConstants
The namespace identifier for the SOAP 1.2 encoding.

**URI NS SOAP 1 2 ENVELOPE** - Static variable in interface javax.xml.soap.SOAPConstants
The namespace identifier for the SOAP 1.2 envelope.

**URI NS SOAP ENCODING** - Static variable in interface javax.xml.soap.SOAPConstants
The namespace identifier for the SOAP 1.1 encoding.

**URI NS SOAP ENVELOPE** - Static variable in interface javax.xml.soap.SOAPConstants
The namespace identifier for the SOAP 1.1 envelope, All SOAPElements in this namespace are defined by the SOAP 1.1 specification.

**URI SOAP 1 2 ROLE NEXT** - Static variable in interface javax.xml.soap.SOAPConstants
The URI identifying the next application processing a SOAP request as the intended role for a SOAP 1.2 header entry (see section 2.2 of part 1 of the SOAP 1.2 specification).

**URI SOAP 1 2 ROLE NONE** - Static variable in interface javax.xml.soap.SOAPConstants
The URI specifying the role None in SOAP 1.2.

**URI SOAP 1 2 ROLE ULTIMATE RECEIVER** - Static variable in interface javax.xml.soap.SOAPConstants
The URI identifying the ultimate receiver of the SOAP 1.2 message.

**URI SOAP ACTOR NEXT** - Static variable in interface javax.xml.soap.SOAPConstants
The URI identifying the next application processing a SOAP request as the intended actor for a SOAP 1.1 header entry (see section 4.2.2 of the SOAP 1.1 specification).

**URIVValidator** - Interface in javax.xml.registry.infomodel
Defines common behavior expected of any class that validates URIs.
**url** - Variable in class `javax.mail.Service`  
The URLName of this service.

**URLDataSource** - Class in `javax.activation`  
The URLDataSource class provides an object that wraps a URL object in a DataSource interface.

**URLDataSource(URL)** - Constructor for class `javax.activation.URLDataSource`  
URLDataSource constructor.

**URLName** - Class in `javax.mail`  
The name of a URL.

**URLName(String, String, int, String, String, String)** - Constructor for class `javax.mail.URLName`  
Creates a URLName object from the specified protocol, host, port number, file, username, and password.

**URLName(URL)** - Constructor for class `javax.mail.URLName`  
Construct a URLName from a java.net.URL object.

**URLName(String)** - Constructor for class `javax.mail.URLName`  
Construct a URLName from the string.

**URLStats** - Interface in `javax.management.j2ee.statistics`  
Specifies the statistics provided by a URL resource.

**urn** - Variable in class `javax.servlet.jsp.tagext.TagLibraryInfo`  
The "reliable" URN indicated in the TLD.

**usage** - Variable in class `javax.mail.Quota.Resource`  
The current usage of the resource.

**USER** - Static variable in class `javax.mail.Flags.Flag`  
A special flag that indicates that this folder supports user defined flags.

**User** - Interface in `javax.xml.registry.infomodel`  
User instances are RegistryObjects that are used to provide information about registered users within the registry.

**USER** - Static variable in interface `javax.xml.registry.LifeCycleManager`  

**USERNAME_PROPERTY** - Static variable in interface `javax.xml.rpc.Call`  
Standard property: User name for authentication  
Type: java.lang.String

**USERNAME_PROPERTY** - Static variable in interface `javax.xml.rpc.Stub`  
Standard property: User name for authentication.

**USERNAME_PROPERTY** - Static variable in interface `javax.xml.ws.BindingProvider`  
Standard property: User name for authentication.

**UserTransaction** - Interface in `javax.transaction`  
The UserTransaction interface defines the methods that allow an application to explicitly manage transaction boundaries.
V1_3 - Static variable in class javax.enterprise.deploy.shared.DConfigBeanVersionType
J2EE Platform version 1.3

V1_3_1 - Static variable in class javax.enterprise.deploy.shared.DConfigBeanVersionType
J2EE Platform version 1.3.1
THIS CONSTANT SHOULD NEVER BE USED.

V1_4 - Static variable in class javax.enterprise.deploy.shared.DConfigBeanVersionType
J2EE Platform version 1.4

V5 - Static variable in class javax.enterprise.deploy.shared.DConfigBeanVersionType
Java EE Platform version 5

validate(FacesContext) - Method in class javax.faces.component.UIInput
Perform the following algorithm to validate the local value of this UIInput.

validate(FacesContext, UIComponent, Object) - Method in class javax.faces.validator.DoubleRangeValidator

validate(FacesContext, UIComponent, Object) - Method in class javax.faces.validator.LengthValidator

validate(FacesContext, UIComponent, Object) - Method in class javax.faces.validator.LongRangeValidator

validate(FacesContext, UIComponent, Object) - Method in interface javax.faces.validator.Validator
Perform the correctness checks implemented by this Validator against the specified UIComponent.

validate() - Method in class javax.mail.internet.InternetAddress
Validate that this address conforms to the syntax rules of RFC 822.

validate() - Method in interface javax.resource.spi.ActivationSpec
This method may be called by a deployment tool to validate the overall activation configuration information provided by the endpoint deployer.

validate(TagData) - Method in class javax.servlet.jsp.tagext.TagExtraInfo
Translation-time validation of the attributes.

validate(TagData) - Method in class javax.servlet.jsp.tagext.TagInfo
Translation-time validation of the attributes.

validate(String, String, PageData) - Method in class
javax.servlet.jsp.tagext.**TagLibraryValidator**
Validate a JSP page.

**validate(Object)** - Method in interface javax.xml.bind.**Validator**
Deprecated. since JAXB2.0

**validateRoot(Object)** - Method in interface javax.xml.bind.**Validator**
Deprecated. since JAXB2.0

**validateValue(FacesContext, Object)** - Method in class javax.faces.component.**UIInput**
Set the "valid" property according to the below algorithm.

**validateValue(FacesContext, Object)** - Method in class javax.faces.component.**UISelectMany**
In addition to the standard validation behavior inherited from UIInput, ensure that any specified values are equal to one of the available options.

**validateValue(FacesContext, Object)** - Method in class javax.faces.component.**UISelectOne**
In addition to the standard validation behavior inherited from UIInput, ensure that any specified value is equal to one of the available options.

**validating** - Variable in class javax.xml.bind.helpers.**AbstractUnmarshallerImpl**
whether or not the unmarshaller will validate

**ValidatingManagedConnectionFactory** - Interface in javax.resource.spi
This interface is implemented by a ManagedConnectionFactory instance that supports the ability to validate ManagedConnection objects.

**ValidationEvent** - Interface in javax.xml.bind
This event indicates that a problem was encountered while validating the incoming XML data during an unmarshal operation, while performing on-demand validation of the Java content tree, or while marshalling the Java content tree back to XML data.

**ValidationEventCollector** - Class in javax.xml.bind.util
ValidationEventHandler implementation that collects all events.

**ValidationEventCollector()** - Constructor for class javax.xml.bind.util.**ValidationEventCollector**

**ValidationEventHandler** - Interface in javax.xml.bind
A basic event handler interface for validation errors.

**ValidationEventImpl** - Class in javax.xml.bind.helpers
Default implementation of the ValidationEvent interface.

**ValidationEventImpl(int, String, ValidationEventLocator)** - Constructor for class javax.xml.bind.helpers.**ValidationEventImpl**
Create a new ValidationEventImpl.

**ValidationEventImpl(int, String, ValidationEventLocator, Throwable)** - Constructor for class javax.xml.bind.helpers.**ValidationEventImpl**
Create a new ValidationEventImpl.

**ValidationEventLocator** - Interface in javax.xml.bind
Encapsulate the location of a ValidationEvent.

**ValidationEventLocatorImpl** - Class in javax.xml.bind.helpers
Default implementation of the ValidationEventLocator interface. 

**ValidationEventLocatorImpl()** - Constructor for class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

- Creates an object with all fields unavailable.

**ValidationEventLocatorImpl(Locator)** - Constructor for class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

- Constructs an object from an `org.xml.sax.Locator`.

**ValidationEventLocatorImpl(SAXParseException)** - Constructor for class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

- Constructs an object from the location information of a `SAXParseException`.

**ValidationEventLocatorImpl(Node)** - Constructor for class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

- Constructs an object that points to a DOM Node.

**ValidationEventLocatorImpl(Object)** - Constructor for class `javax.xml.bind.helpers.ValidationEventLocatorImpl`

- Constructs an object that points to a JAXB content object.

**ValidationException** - Exception in `javax.xml.bind`

- This exception indicates that an error has occurred while performing a validate operation.

**ValidationException(String)** - Constructor for exception `javax.xml.bind.ValidationException`

- Construct a `ValidationException` with the specified detail message.

**ValidationException(String, String)** - Constructor for exception `javax.xml.bind.ValidationException`

- Construct a `ValidationException` with the specified detail message and vendor specific errorCode.

**ValidationException(Throwables)** - Constructor for exception `javax.xml.bind.ValidationError`

- Construct a `ValidationException` with a linkedException.

**ValidationException(String, Throwable)** - Constructor for exception `javax.xml.bind.ValidationError`

- Construct a `ValidationException` with the specified detail message and linkedException.

**ValidationMessage** - Class in `javax.servlet.jsp.tagext`

- A validation message from either TagLibraryValidator or TagExtraInfo.

**ValidationMessage(String, String)** - Constructor for class `javax.servlet.jsp.tagext.ValidationError`

- Create a `ValidationMessage`.

**Validator** - Interface in `javax.faces.validator`

- A `Validator` implementation is a class that can perform validation (correctness checks) on a `EditableValueHolder`.

**Validator** - Interface in `javax.xml.bind`

- Deprecated. since JAXB 2.0
**VALIDATOR ID** - Static variable in class `javax.faces.validator.DoubleRangeValidator`
The standard converter id for this converter.

**VALIDATOR ID** - Static variable in class `javax.faces.validator.LengthValidator`
The standard validator id for this validator.

**VALIDATOR ID** - Static variable in class `javax.faces.validator.LongRangeValidator`
The standard converter id for this converter.

**ValidatorELTag** - Class in `javax.faces.webapp`

ValidatorELTag is a base class for all JSP custom actions that create and register a Validator instance on the `EditableValueHolder` associated with our most immediate surrounding instance of a tag whose implementation class is a subclass of `UIComponentTag`.

**ValidatorELTag()** - Constructor for class `javax.faces.webapp.ValidatorELTag`

**ValidatorException** - Exception in `javax.faces.validator`

A `ValidatorException` is an exception thrown by the validate() method of a `Validator` to indicate that validation failed.

**ValidatorException(FacesMessage)** - Constructor for exception `javax.faces.validator.ValidatorException`

Construct a new exception with the specified message and no root cause.

**ValidatorException(FacesMessage, Throwable)** - Constructor for exception `javax.faces.validator.ValidatorException`

Construct a new exception with the specified detail message and root cause.

**ValidatorTag** - Class in `javax.faces.webapp`

Deprecated. This has been partially replaced by `ValidatorELTag`. The remainder of the functionality, namely, the binding facility and the implementation of the `ValidatorTag.createValidator()` method, is now an implementation detail.

**ValidatorTag()** - Constructor for class `javax.faces.webapp.ValidatorTag`

Deprecated.

**validSent** - Variable in class `javax.mail.event.TransportEvent`

**validSent** - Variable in exception `javax.mail.SendFailedException`

**validUnsent** - Variable in class `javax.mail.event.TransportEvent`

**validUnsent** - Variable in exception `javax.mail.SendFailedException`

**value** - Variable in class `javax.mail.Header`
The value of the header.

**value** - Variable in class `javax.xml.bind.JAXBElement` xml element value.
value - Variable in class javax.xml.rpc.holders.BigDecimalHolder
value - Variable in class javax.xml.rpc.holders.BigIntegerHolder
value - Variable in class javax.xml.rpc.holders.BooleanHolder
value - Variable in class javax.xml.rpc.holders.BooleanWrapperHolder
value - Variable in class javax.xml.rpc.holders.ByteArrayHolder
value - Variable in class javax.xml.rpc.holders.ByteHolder
value - Variable in class javax.xml.rpc.holders.ByteWrapperHolder
value - Variable in class javax.xml.rpc.holders.CalendarHolder
value - Variable in class javax.xml.rpc.holders.DoubleHolder
value - Variable in class javax.xml.rpc.holders.DoubleWrapperHolder
value - Variable in class javax.xml.rpc.holders.FloatHolder
value - Variable in class javax.xml.rpc.holders.FloatWrapperHolder
value - Variable in class javax.xml.rpc.holders.IntegerWrapperHolder
value - Variable in class javax.xml.rpc.holders.IntHolder
value - Variable in class javax.xml.rpc.holders.LongHolder
value - Variable in class javax.xml.rpc.holders.LongWrapperHolder
value - Variable in class javax.xml.rpc.holders.ObjectHolder
value - Variable in class javax.xml.rpc.holders.QNameHolder
value - Variable in class javax.xml.rpc.holders.ShortHolder
value - Variable in class javax.xml.rpc.holders.ShortWrapperHolder
value - Variable in class javax.xml.rpc.holders.StringHolder
value - Variable in class javax.xml.ws.Holder

The value contained in the holder.

VALUE_TYPE_EMBEDDED_PATH - Static variable in interface javax.xml.registry.infomodel.ClassificationScheme
Each taxonomy value in ClassificationScheme embeds the full path from scheme to that Concept.
**VALUE TYPE NON UNIQUE** - Static variable in interface javax.xml.registry.infomodel.ClassificationScheme
Taxonomy values in ClassificationScheme may be repeated within the same scheme.

**VALUE TYPE UNIQUE** - Static variable in interface javax.xml.registry.infomodel.ClassificationScheme
Each taxonomy value in ClassificationScheme is unique.

**ValueBinding** - Class in javax.faces.el
Deprecated. This has been replaced by ValueExpression.
**ValueBinding()** - Constructor for class javax.faces.el.ValueBinding
Deprecated.

**valueBound(HttpSessionBindingEvent)** - Method in interface javax.servlet.http.HttpSessionBindingListener
Notifies the object that it is being bound to a session and identifies the session.

**ValueChangeEvent** - Class in javax.faces.event
A ValueChangeEvent is a notification that the local value of the source component has been change as a result of user interface activity.

**ValueChangeEvent(UIComponent, Object, Object)** - Constructor for class javax.faces.event.ValueChangeEvent
Construct a new event object from the specified source component, old value, and new value.

**ValueChangeListener** - Interface in javax.faces.event
A listener interface for receiving ValueChangeEvent s.

**ValueExpression** - Class in javax.el
An Expression that can get or set a value.
**ValueExpression()** - Constructor for class javax.el.ValueExpression

**ValueHolder** - Interface in javax.faces.component
ValueHolder is an interface that may be implemented by any concrete UIComponent that wishes to support a local value, as well as access data in the model tier via a value expression, and support conversion between String and the model tier data's native data type.

**valueOf(String)** - Static method in enum javax.annotation.Resource.AuthenticationType
Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum javax.ejb.TransactionAttributeType
Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum javax.ejb.TransactionManagementType
Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum javax.jws.soap.SOAPBinding.ParameterStyle
Returns the enum constant of this type with the specified name.

**valueOf(String)** - Static method in enum javax.jws.soap.SOAPBinding.Style
Returns the enum constant of this type with the specified name.
valueOf(String) - Static method in enum javax.jws.soap.SOAPBinding.Use
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.jws.WebParam.Mode
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.CascadeType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.DiscriminatorType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.EnumType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.FetchType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.FlushModeType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.GenerationType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.InheritanceType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.LockModeType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.PersistenceContextType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.persistence.spi.PersistenceUnitTransactionType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.xml.bind.annotation.XmlAccessOrder
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.xml.bind.annotation.XmlAccessType
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.xml.bind.annotation.XmlNsForm
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.xml.ws.handler.MessageContext.Scope
Returns the enum constant of this type with the specified name.

valueOf(String) - Static method in enum javax.xml.ws.Service.Mode
Returns the enum constant of this type with the specified name.
values() - Static method in enum javax.annotation.Resource.AuthenticationType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.ejb.TransactionAttributeType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.ejb.TransactionManagementType
Returns an array containing the constants of this enum type, in the order they're declared.

VALUES - Static variable in class javax.faces.application.FacesMessage
Immutable List of valid FacesMessage.Severity instances, in ascending order of their ordinal value.

VALUES - Static variable in class javax.faces.event.PhaseId
List of valid PhaseId instances, in ascending order of their ordinal value.

values() - Static method in enum javax.jws.soap.SOAPBinding.ParameterStyle
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.jws.soap.SOAPBinding.Style
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.jws.soap.SOAPBinding.Use
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.jws.WebParam.Mode
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.CascadeType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.DiscriminatorType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.EnumType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.FetchType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.FlushModeType
Returns an array containing the constants of this enum type, in the order they're declared.

values() - Static method in enum javax.persistence.GenerationType
Returns an array containing the constants of this enum type, in the order they're declared.
values() - Static method in enum javax.persistence.InheritanceType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.persistence.LockModeType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.persistence.PersistenceContextType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.persistence.spi.PersistenceUnitTransactionType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.persistence.TemporalType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.xml.bind.annotation.XmlAccessOrder
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.xml.bind.annotation.XmlAccessType
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.xml.bind.annotation.XmlNsForm
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.xml.ws.handler.MessageContext.Scope
Returns an array containing the constants of this enum type, in
the order they're declared.
values() - Static method in enum javax.xml.ws.Service.Mode
Returns an array containing the constants of this enum type, in
the order they're declared.
VALUES_MAP - Static variable in class
javax.faces.application.FacesMessage
Immutable Map of valid FacesMessage.Severity instances, keyed
by name.
valueUnbound(HttpSessionBindingEvent) - Method in interface
javax.servlet.http.HttpSessionBindingListener
Notifies the object that it is being unbound from a session and
identifies the session.
VariableInfo - Class in javax.servlet.jsp.tagext
Information on the scripting variables that are
created/modified by a tag (at run-time).
VariableInfo(String, String, boolean, int) - Constructor for class
javax.servlet.jsp.tagext.VariableInfo
Constructor These objects can be created (at translation time)
by the TagExtraInfo instances.

**VariableMapper** - Class in *javax.el*
The interface to a map between EL variables and the EL expressions they are associated with.

**VariableMapper()** - Constructor for class *javax.el.VariableMapper*

**VariableResolver** - Class in *javax.faces.el*
Deprecated. This has been replaced by *ELResolver* when operating with a null base argument.

**VariableResolver()** - Constructor for class *javax.faces.el.VariableResolver*
Deprecated.

**VariableResolver** - Interface in *javax.servlet.jsp.el*
Deprecated. As of JSP 2.1, replaced by *ELResolver*

**Version** - Annotation Type in *javax.persistence*
This annotation specifies the version field or property of an entity class that serves as its optimistic lock value.

**Versionable** - Interface in *javax.xml.registry.infomodel*
The Versionable interface defines the behavior common to classes that are capable of creating versions of their instances.

**VERSIONABLE** - Static variable in interface *javax.xml.registry.LifeCycleManager*

**VIEW_STATE_PARAM** - Static variable in class *javax.faces.render.ResponseStateManager*
Implementations must use this value as the name and id of the client parameter in which to save the state between requests.

**ViewExpiredException** - Exception in *javax.faces.application*
Implementations must throw this *FacesException* when attempting to restore the view

**ViewExpiredException(String)** - Constructor for exception

**ViewExpiredException(String, String)** - Constructor for exception

**ViewExpiredException(Throwable, String)** - Constructor for exception

**ViewExpiredException(String, Throwable, String)** - Constructor for exception

Construct a new exception with no detail message or root cause.
**ViewHandler** - Class in `javax.faces.application`  
*ViewHandler* is the pluggability mechanism for allowing implementations of or applications using the JavaServer Faces specification to provide their own handling of the activities in the *Render Response* and *Restore View* phases of the request processing lifecycle.

**ViewHandler()** - Constructor for class `javax.faces.application.ViewHandler`  

**ViewHandlerWrapper** - Class in `javax.faces.application`  
Provides a simple implementation of *ViewHandler* that can be subclassed by developers wishing to provide specialized behavior to an existing *ViewHandler* instance.

**ViewHandlerWrapper()** - Constructor for class `javax.faces.application.ViewHandlerWrapper`
W3CDomHandler - Class in javax.xml.bind.annotation
   - Default constructor.
   - Constructor for class W3CDomHandler
      Constructor that allows applications to specify which DOM implementation to be used.

WAR - Static variable in class javax.enterprise.deploy.shared.ModuleType
      The module is an Web Application archive.

WARNING - Static variable in interface javax.xml.bind.ValidationEvent
   Conditions that are not errors or fatal errors as defined by the XML 1.0 recommendation

WebEndpoint - Annotation Type in javax.xml.ws
   Used to annotate the getPortName() methods of a generated service interface.

WebFault - Annotation Type in javax.xml.ws
   Used to annotate service specific exception classes to customize to the local and namespace name of the fault element and the name of the fault bean.

WebMethod - Annotation Type in javax.jws
   Customizes a method that is exposed as a Web Service operation.

WebParam - Annotation Type in javax.jws
   Customizes the mapping of an individual parameter to a Web Service message part and XML element.

WebParam.Mode - Enum in javax.jws
   The direction in which the parameter flows

WebResourcePermission - Class in javax.security.jacc
   Class for Servlet web resource permissions.

WebResourcePermission(String, String) - Constructor for class javax.security.jacc.WebResourcePermission
   Creates a new WebResourcePermission with the specified name and actions.

WebResourcePermission(String, String[]) - Constructor for class javax.security.jacc.WebResourcePermission
   Creates a new WebResourcePermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods.

WebResourcePermission(HttpServletRequest) - Constructor for class javax.security.jacc.WebResourcePermission
   Creates a new WebResourcePermission from the HttpServletRequest object.

WebResult - Annotation Type in javax.jws
Customizes the mapping of the return value to a WSDL part and XML element.

- **WebRoleRefPermission** - Class in `javax.security.jacc`
  - Class for Servlet `isUserInRole (String reference)` permissions.
- **WebRoleRefPermission(String, String)** - Constructor for class `javax.security.jacc.WebRoleRefPermission`
  - Creates a new WebRoleRefPermission with the specified name and actions.

- **WebService** - Annotation Type in `javax.jws`
  - Marks a Java class as implementing a Web Service, or a Java interface as defining a Web Service interface.
- **WebServiceClient** - Annotation Type in `javax.xml.ws`
  - Used to annotate a generated service interface.
- **WebServiceContext** - Interface in `javax.xml.ws`
  - A WebServiceContext makes it possible for a web service endpoint implementation class to access message context and security information relative to a request being served.
- **WebServiceException** - Exception in `javax.xml.ws`
  - The WebServiceException class is the base exception class for all JAX-WS API runtime exceptions.
- **WebServiceException()** - Constructor for exception `javax.xml.ws.WebServiceException`
  - Constructs a new exception with null as its detail message.
- **WebServiceException(String)** - Constructor for exception `javax.xml.ws.WebServiceException`
  - Constructs a new exception with the specified detail message.
- **WebServiceException(String, Throwable)** - Constructor for exception `javax.xml.ws.WebServiceException`
  - Constructs a new exception with the specified detail message and cause.
- **WebServiceException(Throwerable)** - Constructor for exception `javax.xml.ws.WebServiceException`
  - Constructs a new WebServiceException with the specified cause and a detail message of (cause==null ?
- **WebServicePermission** - Class in `javax.xml.ws`
  - This class defines web service permissions.
- **WebServicePermission(String)** - Constructor for class `javax.xml.ws.WebServicePermission`
  - Creates a new permission with the specified name.
- **WebServicePermission(String, String)** - Constructor for class `javax.xml.ws.WebServicePermission`
  - Creates a new permission with the specified name and actions.
- **WebServiceProvider** - Annotation Type in `javax.xml.ws`
  - Used to annotate a Provider implementation class.
- **WebServiceRef** - Annotation Type in `javax.xml.ws`
  - TheWebServiceRef annotation is used to define a reference to a web service and (optionally) an injection target for it.
- **WebServiceRefs** - Annotation Type in `javax.xml.ws`
  - The WebServiceRefs annotation allows multiple web service references to be declared at the class level.
**WebUserDataPermission** - Class in `javax.security.jacc`

Class for Servlet Web user data permissions.

**WebUserDataPermission(String, String)** - Constructor for class `javax.security.jacc.WebUserDataPermission`

Creates a new WebUserDataPermission with the specified name and actions.

**WebUserDataPermission(String, String[], String)** - Constructor for class `javax.security.jacc.WebUserDataPermission`

Creates a new WebUserDataPermission with name corresponding to the URLPatternSpec, and actions composed from the array of HTTP methods and the transport type.

**WebUserDataPermission(HttpServletRequest)** - Constructor for class `javax.security.jacc.WebUserDataPermission`

Creates a new WebUserDataPermission from the HttpServletRequest object.

**Work** - Interface in `javax.resource.spi.work`

This models a Work instance that would be executed by a WorkManager upon submission.

**WORK_ACCEPTED** - Static variable in class `javax.resource.spi.work.WorkEvent`

Indicates Work instance has been accepted.

**WORK_COMPLETED** - Static variable in class `javax.resource.spi.work.WorkEvent`

Indicates Work instance has completed execution.

**WORK_REJECTED** - Static variable in class `javax.resource.spi.work.WorkEvent`

Indicates Work instance has been rejected.

**WORK_STARTED** - Static variable in class `javax.resource.spi.work.WorkEvent`

Indicates Work instance has started execution.

**workAccepted(WorkEvent)** - Method in class `javax.resource.spi.work.WorkAdapter`

Invoked when a Work instance has been accepted.

**workAccepted(WorkEvent)** - Method in interface `javax.resource.spi.work.WorkListener`

Invoked when a Work instance has been accepted.

**WorkAdapter** - Class in `javax.resource.spi.work`

This class is provided as a convenience for easily creating WorkListener instances by extending this class and overriding only those methods of interest.

**WorkAdapter()** - Constructor for class `javax.resource.spi.work.WorkAdapter`

**workCompleted(WorkEvent)** - Method in class `javax.resource.spi.work.WorkAdapter`

Invoked when a Work instance has completed execution.

**workCompleted(WorkEvent)** - Method in interface `javax.resource.spi.work.WorkListener`

Invoked when a Work instance has completed execution.

**WorkCompletedException** - Exception in `javax.resource.spi.work`
This exception is thrown by a WorkManager to indicate that a submitted Work instance has completed with an exception.

**WorkCompletedException()** - Constructor for exception

`javax.resource.spi.work.WorkCompletedException`

Constructs a new instance with null as its detail message.

**WorkCompletedException(String)** - Constructor for exception

`javax.resource.spi.work.WorkCompletedException`

Constructs a new instance with the specified detail message.

**WorkCompletedException(Throwables)** - Constructor for exception

`javax.resource.spi.work.WorkCompletedException`

Constructs a new throwables with the specified cause.

**WorkCompletedException(String, Throwable)** - Constructor for exception

`javax.resource.spi.work.WorkCompletedException`

Constructs a new throwables with the specified detail message and cause.

**WorkCompletedException(String, String)** - Constructor for exception

`javax.resource.spi.work.WorkCompletedException`

Constructs a new throwables with the specified detail message and error code.

**WorkEvent** - Class in `javax.resource.spi.work`

This class models the various events that occur during the processing of a Work instance.

**WorkEvent(Object, int, Work, WorkException)** - Constructor for class

`javax.resource.spi.work.WorkEvent`

Constructor.

**WorkEvent(Object, int, Work, WorkException, long)** - Constructor for class

`javax.resource.spi.work.WorkEvent`

Constructor.

**WorkException** - Exception in `javax.resource.spi.work`

A common base class for all Work processing related exceptions.

**WorkException()** - Constructor for exception

`javax.resource.spi.work.WorkException`

Constructs a new instance with null as its detail message.

**WorkException(String)** - Constructor for exception

`javax.resource.spi.work.WorkException`

Constructs a new instance with the specified detail message.

**WorkException(Throwables)** - Constructor for exception

`javax.resource.spi.work.WorkException`

Constructs a new throwables with the specified cause.

**WorkException(String, Throwable)** - Constructor for exception

`javax.resource.spi.work.WorkException`

Constructs a new throwables with the specified detail message and cause.

**WorkException(String, String)** - Constructor for exception

`javax.resource.spi.work.WorkException`

Constructs a new throwables with the specified detail message and error code.

**WorkListener** - Interface in `javax.resource.spi.work`

This models a WorkListener instance which would be notified by the WorkManager when the various Work processing events (work
accepted, work rejected, work started, work completed) occur.

**WorkManager** - Interface in `javax.resource.spi.work`
This interface models a WorkManager which provides a facility to submit Work instances for execution.

**workRejected(WorkEvent)** - Method in class `javax.resource.spi.work.WorkAdapter`
Invoked when a Work instance has been rejected.

**workRejected(WorkEvent)** - Method in interface `javax.resource.spi.work.WorkListener`
Invoked when a Work instance has been rejected.

**WorkRejectedException** - Exception in `javax.resource.spi.work`
This exception is thrown by a WorkManager to indicate that a submitted Work instance has been rejected.

**WorkRejectedException()** - Constructor for exception
Constructs a new instance with null as its detail message.

**WorkRejectedException(String)** - Constructor for exception
Constructs a new instance with the specified detail message.

**WorkRejectedException(ThrowException)** - Constructor for exception
Constructs a new throwable with the specified cause.

**WorkRejectedException(String, Throwable)** - Constructor for exception
 Constructs a new throwable with the specified detail message and cause.

**WorkRejectedException(String, String)** - Constructor for exception
Constructs a new throwable with the specified detail message and an error code.

**workStarted(WorkEvent)** - Method in class `javax.resource.spi.work.WorkAdapter`
Invoked when a Work instance has started execution.

**write(char[], int, int)** - Method in class `javax.faces.context.ResponseWriterWrapper`
The default behavior of this method is to call `Writer.write(char[], int, int)` on the wrapped `ResponseWriter` object.

**write(OutputStream)** - Method in interface `javax.resource.cci.Streamable`
Write fields of a Streamable object to an OutputStream

**WRITE XML DECLARATION** - Static variable in class `javax.xml.soap.SOAPMessage`
Specifies whether the SOAP Message will contain an XML declaration when it is sent.

**writeAsEncodedUnicode(Writer)** - Method in interface `javax.xml.stream.events.XMLEvent`
This method will write the XMLEvent as per the XML 1.0 specification as Unicode characters.

**writeAttribute(String, Object, String)** - Method in class javax.faces.context.ResponseWriter
Write an attribute name and corresponding value, after converting that text to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered.

**writeAttribute(String, Object, String)** - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call ResponseWriter.writeAttribute(String, Object, String) on the wrapped ResponseWriter object.

**writeAttribute(String, String)** - Method in interface javax.xml.stream.XMLStreamWriter
Writes an attribute to the output stream without a prefix.

**writeAttribute(String, String, String, String)** - Method in interface javax.xml.stream.XMLStreamWriter
Writes an attribute to the output stream

**writeBoolean(boolean)** - Method in interface javax.jms.BytesMessage
Writes a boolean to the bytes message stream as a 1-byte value.

**writeBoolean(boolean)** - Method in interface javax.jms.StreamMessage
Writes a boolean to the stream message.

**writeByte(byte)** - Method in interface javax.jms.BytesMessage
Writes a byte to the bytes message stream as a 1-byte value.

**writeByte(byte)** - Method in interface javax.jms.StreamMessage
Writes a byte to the stream message.

**writeBytes(byte[])** - Method in interface javax.jms.BytesMessage
Writes a byte array to the bytes message stream.

**writeBytes(byte[], int, int)** - Method in interface javax.jms.BytesMessage
Writes a portion of a byte array to the bytes message stream.

**writeBytes(byte[])** - Method in interface javax.jms.StreamMessage
Writes a byte array field to the stream message.

**writeBytes(byte[], int, int)** - Method in interface javax.jms.StreamMessage
Writes a portion of a byte array as a byte array field to the stream message.

**writeCData(String)** - Method in interface javax.xml.stream.XMLStreamWriter
Writes a CData section

**writeChar(char)** - Method in interface javax.jms.BytesMessage
Writes a char to the bytes message stream as a 2-byte value, high byte first.

**writeChar(char)** - Method in interface javax.jms.StreamMessage
Writes a char to the stream message.

**writeCharacters(String)** - Method in interface
javax.xml.stream.XMLStreamWriter
Write text to the output
writeCharacters(char[], int, int) - Method in interface
class javax.xml.stream.XMLStreamWriter
Write text to the output
writeComment(Object) - Method in class
class javax.faces.context.ResponseWriter
Write a comment containing the specified text, after converting
that text to a String (if necessary), and after performing any
escaping appropriate for the markup language being rendered.
writeComment(Object) - Method in class
class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call
ResponseWriter.writeComment(Object) on the wrapped
ResponseWriter object.
writeComment(String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Writes an xml comment with the data enclosed
writeDefaultNamespace(String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Writes the default namespace to the stream
writeDouble(double) - Method in interface class javax.jms.BytesMessage
Converts the double argument to a long using the
doubleToLongBits method in class Double, and then writes that
long value to the bytes message stream as an 8-byte quantity,
high byte first.
writeDouble(double) - Method in interface class javax.jms.StreamMessage
Writes a double to the stream message.
writeDTD(String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Write a DTD section.
writeEJBHome(EJBHome, ObjectOutputStream) - Method in interface
class java.ejb.spi.HandleDelegate
Serialize the EJBHome reference corresponding to a HomeHandle.
writeEJBOBJECT(EJBObject, ObjectOutputStream) - Method in interface
class java.ejb.spi.HandleDelegate
Serialize the EJBObject reference corresponding to a Handle.
writeEmptyElement(String, String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Writes an empty element tag to the output
writeEmptyElement(String, String, String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Writes an empty element tag to the output
writeEmptyElement(String) - Method in interface
class javax.xml.stream.XMLStreamWriter
Writes an empty element tag to the output
writeEndElement() - Method in interface
javax.xml.stream.XMLStreamWriter
  Writes an end tag to the output relying on the internal state of the writer to determine the prefix and local name of the event.

writeEntityRef(String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes an entity reference

writeExternal(ObjectOutput) - Method in class javax.activation.MimeType
  The object implements the writeExternal method to save its contents by calling the methods of DataOutput for its primitive values or calling the writeObject method of ObjectOutputStream for objects, strings and arrays.

writeFloat(float) - Method in interface javax.jms.BytesMessage
  Converts the float argument to an int using the floatToIntBits method in class Float, and then writes that int value to the bytes message stream as a 4-byte quantity, high byte first.

writeFloat(float) - Method in interface javax.jms.StreamMessage
  Writes a float to the stream message.

writeInt(int) - Method in interface javax.jms.BytesMessage
  Writes an int to the bytes message stream as four bytes, high byte first.

writeInt(int) - Method in interface javax.jms.StreamMessage
  Writes an int to the stream message.

writeLong(long) - Method in interface javax.jms.BytesMessage
  Writes a long to the bytes message stream as eight bytes, high byte first.

writeLong(long) - Method in interface javax.jms.StreamMessage
  Writes a long to the stream message.

writeNamespace(String, String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a namespace to the output stream If the prefix argument to this method is the empty string, "xmlns", or null this method will delegate to writeDefaultNamespace

writeObject(Object) - Method in interface javax.jms.BytesMessage
  Writes an object to the bytes message stream.

writeObject(Object) - Method in interface javax.jms.StreamMessage
  Writes an object to the stream message.

writeOut(Writer) - Method in class javax.servlet.jsp.tagext.BodyContent
  Write the contents of this BodyContent into a Writer.

writeProcessingInstruction(String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a processing instruction

writeProcessingInstruction(String, String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a processing instruction

writeShort(short) - Method in interface javax.jms.BytesMessage
  Writes a short to the bytes message stream as two bytes, high byte first.
writeShort(short) - Method in interface javax.jms.StreamMessage
  Writes a short to the stream message.
writeStartDocument() - Method in interface javax.xml.stream.XMLStreamWriter
  Write the XML Declaration.
writeStartDocument(String) - Method in interface javax.xml.stream.XMLStreamWriter
  Write the XML Declaration.
writeStartDocument(String, String) - Method in interface javax.xml.stream.XMLStreamWriter
  Write the XML Declaration.
writeStartElement(String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a start tag to the output.
writeStartElement(String, String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a start tag to the output.
writeStartElement(String, String, String) - Method in interface javax.xml.stream.XMLStreamWriter
  Writes a start tag to the output.
writeState(FacesContext, Object) - Method in class javax.faces.application.StateManager
  Save the state represented in the specified state Object instance, in an implementation dependent manner.
writeState(FacesContext, StateManager.SerializedView) - Method in class javax.faces.application.StateManager
  Deprecated. This method has been replaced by
  StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object)
  The default implementation of this method does nothing.
writeState(FacesContext, Object) - Method in class javax.faces.application.StateManagerWrapper
  The default behavior of this method is to call
  StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object) on the wrapped StateManager object.
writeState(FacesContext, StateManager.SerializedView) - Method in class javax.faces.application.StateManagerWrapper
  The default behavior of this method is to call
  StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object) on the wrapped StateManager object.
writeState(FacesContext) - Method in class javax.faces.application.ViewHandler
  Take any appropriate action to either immediately write out the current state information (by calling
  StateManager.writeState(javax.faces.context.FacesContext, java.lang.Object), or noting where state information should
  later be written.
writeState(FacesContext) - Method in class javax.faces.application.ViewHandlerWrapper
  The default behavior of this method is to call
ViewHandler.writeState(javax.faces.context.FacesContext) on the wrapped ViewHandler object.
writeState(FacesContext, Object) - Method in class javax.faces.render.ResponseStateManager
writeState(FacesContext, StateManager.SerializedView) - Method in class javax.faces.render.ResponseStateManager
   Deprecated. This method has been replaced by ResponseStateManager.writeState(javax.faces.context.FacesContext, java.lang.Object)
   The default implementation of this method does nothing.
writeString(String) - Method in interface javax.jms.StreamMessage
   Writes a String to the stream message.
writeText(Object, String) - Method in class javax.faces.context.ResponseWriter
   Write an object, after converting it to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered.
writeText(Object, UIComponent, String) - Method in class javax.faces.context.ResponseWriter
   Write an object, after converting it to a String (if necessary), and after performing any escaping appropriate for the markup language being rendered.
writeText(char[], int, int) - Method in class javax.faces.context.ResponseWriter
   Write text from a character array, after any performing any escaping appropriate for the markup language being rendered.
writeText(Object, String) - Method in class javax.faces.context.ResponseWriterWrapper
   The default behavior of this method is to call ResponseWriter.writeText(Object, String) on the wrapped ResponseWriter object.
writeText(Object, UIComponent, String) - Method in class javax.faces.context.ResponseWriterWrapper
   The default behavior of this method is to call ResponseWriter.writeText(Object, UIComponent, String) on the wrapped ResponseWriter object.
writeText(char[], int, int) - Method in class javax.faces.context.ResponseWriterWrapper
   The default behavior of this method is to call ResponseWriter.writeText(char[], int, int) on the wrapped ResponseWriter object.
writeTo(Object, String, OutputStream) - Method in interface javax.activation.DataContentHandler
   Convert the object to a byte stream of the specified MIME type and write it to the output stream.
writeTo(OutputStream) - Method in class javax.activation.DataHandler
   Write the data to an OutputStream.
writeTo(OutputStream) - Method in class javax.mail.internet.MimeBodyPart
Output the body part as an RFC 822 format stream.
writeTo(OutputStream) - Method in class javax.mail.internet.MimeMessage
Output the message as an RFC 822 format stream.
writeTo(OutputStream, String[]) - Method in class javax.mail.internet.MimeMessage
Output the message as an RFC 822 format stream, without specified headers.
writeTo(OutputStream) - Method in class javax.mail.internet.MimeMultipart
Iterates through all the parts and outputs each MIME part separated by a boundary.
writeTo(OutputStream) - Method in class javax.mail.internet.PreencodedMimeBodyPart
Output the body part as an RFC 822 format stream.
writeTo(OutputStream) - Method in class javax.mail.Multipart
Output an appropriately encoded bytestream to the given OutputStream.
writeTo(OutputStream) - Method in interface javax.mail.Part
Output a bytestream for this Part.
writeTo(OutputStream) - Method in class javax.xml.soap.SOAPMessage
Writes this SOAPMessage object to the given output stream.
writeURIAttribute(String, Object, String) - Method in class javax.faces.context.ResponseWriter
Write a URI attribute name and corresponding value, after converting that text to a String (if necessary), and after performing any encoding appropriate to the markup language being rendered.
writeURIAttribute(String, Object, String) - Method in class javax.faces.context.ResponseWriterWrapper
The default behavior of this method is to call
ResponseWriter.writeURIAttribute(String, Object, String) on the wrapped ResponseWriter object.
writeUTF(String) - Method in interface javax.jms.BytesMessage
Writes a string to the bytes message stream using UTF-8 encoding in a machine-independent manner.
WSDL_DESCRIPTION - Static variable in interface javax.xml.ws.handler.MessageContext
WSDL_INTERFACE - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: name of wsdl interface (2.0) or port type (1.1).
WSDL_OPERATION - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: name of WSDL operation.
WSDL_PORT - Static variable in class javax.xml.ws.Endpoint
Standard property: name of WSDL port.
WSDL_PORT - Static variable in interface javax.xml.ws.handler.MessageContext
Standard property: name of WSDL port.

**WSDL_SERVICE** - Static variable in class `javax.xml.ws.Endpoint`
Standard property: name of WSDL service.

**WSDL_SERVICE** - Static variable in interface `javax.xml.ws.handler.MessageContext`
Standard property: name of WSDL service.
XA_HEURCOM - Static variable in exception
javax.transaction.xa.XAException
   The transaction branch has been heuristically committed.
XA_HEURHAZ - Static variable in exception
javax.transaction.xa.XAException
   The transaction branch may have been heuristically completed.
XA_HEURMIX - Static variable in exception
javax.transaction.xa.XAException
   The transaction branch has been heuristically committed and rolled back.
XA_HEURRRB - Static variable in exception
javax.transaction.xa.XAException
   The transaction branch has been heuristically rolled back.
XA NOMIGRATE - Static variable in exception
javax.transaction.xa.XAException
   Resumption must occur where the suspension occurred.
XA_OK - Static variable in interface
javax.transaction.xa.XAResource
   The transaction work has been prepared normally.
XA_RBBASE - Static variable in exception
javax.transaction.xa.XAException
   The inclusive lower bound of the rollback codes.
XA_RBCOMMFAIL - Static variable in exception
javax.transaction.xa.XAException
   Indicates that the rollback was caused by a communication failure.
XA_RBDDEADLOCK - Static variable in exception
javax.transaction.xa.XAException
   A deadlock was detected.
XA_RBEND - Static variable in exception
javax.transaction.xa.XAException
   The inclusive upper bound of the rollback error code.
XA_RBINTEGRITY - Static variable in exception
javax.transaction.xa.XAException
   A condition that violates the integrity of the resource was detected.
XA_RBOOTHER - Static variable in exception
javax.transaction.xa.XAException
   The resource manager rolled back the transaction branch for a reason not on this list.
XA_RBPROTO - Static variable in exception
javax.transaction.xa.XAException
   A protocol error occurred in the resource manager.
XA_RBBROLLBACK - Static variable in exception
javax.transaction.xa.XAException
   Indicates that the rollback was caused by an unspecified
reason.

**XA_RBTIMEOUT** - Static variable in exception
javax.transaction.xa.XAException
A transaction branch took too long.

**XA_RBTRANSIENT** - Static variable in exception
javax.transaction.xa.XAException
May retry the transaction branch.

**XA_RDONLY** - Static variable in exception
javax.transaction.xa.XAException
The transaction branch was read-only and has been committed.

**XA_RDONLY** - Static variable in interface
javax.transaction.xa.XAResource
The transaction branch has been read-only and has been committed.

**XA_RETRY** - Static variable in exception
javax.transaction.xa.XAException
Routine returned with no effect and may be reissued.

**XACoNnection** - Interface in *javax.jms*
The XACoNnection interface extends the capability of Connection by providing an XASession (optional).

**XACoNnectionFactory** - Interface in *javax.jms*
The XACoNNECTIONFactory interface is a base interface for the XAQueueConnectionFactory and XATopicConnectionFactory interfaces.

**XAER_ASYNC** - Static variable in exception
javax.transaction.xa.XAException
There is an asynchronous operation already outstanding.

**XAER_DUPID** - Static variable in exception
javax.transaction.xa.XAException
The XID already exists.

**XAER_INVAL** - Static variable in exception
javax.transaction.xa.XAException
Invalid arguments were given.

**XAER_NOTA** - Static variable in exception
javax.transaction.xa.XAException
The XID is not valid.

**XAER_OUTSIDE** - Static variable in exception
javax.transaction.xa.XAException
The resource manager is doing work outside a global transaction.

**XAER_PROTO** - Static variable in exception
javax.transaction.xa.XAException
Routine was invoked in an improper context.

**XAER_RMFAIL** - Static variable in exception
javax.transaction.xa.XAException
A resource manager error has occurred in the transaction branch.

**XAER_RMFAIL** - Static variable in exception
javax.transaction.xa.XAException
Resource manager is unavailable.
**XAException** - Exception in *javax.transaction.xa*

The XAException is thrown by the Resource Manager (RM) to inform the Transaction Manager of an error encountered by the involved transaction.

**XAException()** - Constructor for exception

`javax.transaction.xa.XAException`

Create an XAException.

**XAException(String)** - Constructor for exception

`javax.transaction.xa.XAException`

Create an XAException with a given string.

**XAException(int)** - Constructor for exception

`javax.transaction.xa.XAException`

Create an XAException with a given error code.

**XAQueueConnection** - Interface in *javax.jms*

An XAQueueConnection provides the same create options as QueueConnection (optional).

**XAQueueConnectionFactory** - Interface in *javax.jms*

An XAQueueConnectionFactory provides the same create options as a QueueConnectionFactory (optional).

**XAQueueSession** - Interface in *javax.jms*

An XAQueueSession provides a regular QueueSession, which can be used to create QueueReceiver, QueueSender, and QueueBrowser objects (optional).

**XAResource** - Interface in *javax.transaction.xa*

The XAResource interface is a Java mapping of the industry standard XA interface based on the X/Open CAE Specification (Distributed Transaction Processing: The XA Specification).

**XASession** - Interface in *javax.jms*

The XASession interface extends the capability of Session by adding access to a JMS provider's support for the Java Transaction API (JTA) (optional).

**XATerminator** - Interface in *javax.resource.spi*

The XATerminator interface is used for transaction completion and crash recovery flows.

**XATopicConnection** - Interface in *javax.jms*

An XATopicConnection provides the same create options as TopicConnection (optional).

**XATopicConnectionFactory** - Interface in *javax.jms*

An XATopicConnectionFactory provides the same create options as a TopicConnectionFactory (optional).

**XATopicSession** - Interface in *javax.jms*

An XATopicSession provides a regular TopicSession.

**Xid** - Interface in *javax.transaction.xa*

The Xid interface is a Java mapping of the X/Open transaction identifier XID structure.

**XmlAccessOrder** - Enum in *javax.xml.bind.annotation*

Used by XmlAccessorOrder to control the ordering of properties and fields in a JAXB bound class.

**XmlAccessorOrder** - Annotation Type in *javax.xml.bind.annotation*

Controls the ordering of fields and properties in a class.
**XmlAccessorType** - Annotation Type in `javax.xml.bind.annotation`
Controls whether fields or JavaBean properties are serialized by default.

**XmlAccessType** - Enum in `javax.xml.bind.annotation`
Used by XmlAccessorType to control serialization of fields or properties.

**XmlAdapter<ValueType,BoundType>** - Class in `javax.xml.bind.annotation.adapters`
Adapts a Java type for custom marshaling.

**XmlAdapter()** - Constructor for class `javax.xml.bind.annotation.adapters.XmlAdapter`
Do-nothing constructor for the derived classes.

**XmlAnyAttribute** - Annotation Type in `javax.xml.bind.annotation`
Maps a JavaBean property to a map of wildcard attributes.

**XmlAnyElement** - Annotation Type in `javax.xml.bind.annotation`
Maps a JavaBean property to XML infoset representation and/or JAXB element.

**XmlAttachmentRef** - Annotation Type in `javax.xml.bind.annotation`
Marks a field/property that its XML form is a uri reference to mime content.

**XmlAttribute** - Annotation Type in `javax.xml.bind.annotation`
Maps a JavaBean property to a XML attribute.

**XmlElement** - Annotation Type in `javax.xml.bind.annotation`
Maps a JavaBean property to a XML element derived from property name.

**XmlElement.DEFAULT** - Class in `javax.xml.bind.annotation`
Used in `XmlElement.type()` to signal that the type be inferred from the signature of the property.

**XmlElement.DEFAULT()** - Constructor for class `javax.xml.bind.annotation.XmlElement.DEFAULT`

**XmlElementDecl** - Annotation Type in `javax.xml.bind.annotation`
Maps a factory method to a XML element.

**XmlElementDecl.GLOBAL** - Class in `javax.xml.bind.annotation`
Used in `XmlElementDecl.scope()` to signal that the declaration is in the global scope.

**XmlElementDecl.GLOBAL()** - Constructor for class `javax.xml.bind.annotation.XmlElementDecl.GLOBAL`

**XmlElementRef** - Annotation Type in `javax.xml.bind.annotation`
Maps a JavaBean property to a XML element derived from property's type.

**XmlElementRef.DEFAULT** - Class in `javax.xml.bind.annotation`
Used in `XmlElementRef.type()` to signal that the type be inferred from the signature of the property.

**XmlElementRef.DEFAULT()** - Constructor for class `javax.xml.bind.annotation.XmlElementRef.DEFAULT`

**XmlElementRefs** - Annotation Type in `javax.xml.bind.annotation`
Marks a property that refers to classes with **XmlElement** or
JAXBElement.

**XmlElements** - Annotation Type in **javax.xml.bind.annotation**
A container for multiple `@XmlElement` annotations.

**XmlElementWrapper** - Annotation Type in **javax.xml.bind.annotation**
Generates a wrapper element around XML representation.

**XmlEnum** - Annotation Type in **javax.xml.bind.annotation**
Maps an enum type `Enum` to XML representation.

**XmlEnumValue** - Annotation Type in **javax.xml.bind.annotation**
Maps an enum constant in `Enum` type to XML representation.

**XmlEvent** - Interface in **javax.xml.stream.events**
This is the base event interface for handling markup events.

**XmlEventAllocator** - Interface in **javax.xml.stream.util**
This interface defines a class that allows a user to register a way to allocate events given an XMLStreamReader.

**XmlEventConsumer** - Interface in **javax.xml.stream.util**
This interface defines an event consumer interface.

**XmlEventFactory** - Class in **javax.xml.stream**
This interface defines a utility class for creating instances of XMLEvents.

**XmlEventFactory()** - Constructor for class
`javax.xml.stream.XMLEventFactory`.

**XMLEventReader** - Interface in **javax.xml.stream**
This is the top level interface for parsing XML Events.

**XMLEventWriter** - Interface in **javax.xml.stream**
This is the top level interface for writing XML documents.

**XmlID** - Annotation Type in **javax.xml.bind.annotation**
Maps a JavaBean property to XML ID.

**XmlIDREF** - Annotation Type in **javax.xml.bind.annotation**
Maps a JavaBean property to XML IDREF.

**XmlInlineBinaryData** - Annotation Type in **javax.xml.bind.annotation**
Disable consideration of XOP encoding for datatypes that are bound to base64-encoded binary data in XML.

**XMLInputFactory** - Class in **javax.xml.stream**
Defines an abstract implementation of a factory for getting streams.

**XMLInputFactory()** - Constructor for class
`javax.xml.stream.XMLInputFactory`.

**XmlJavaTypeAdapter** - Annotation Type in **javax.xml.bind.annotation.adapters**
Use an adapter that implements `XmlAdapter` for custom marshaling.

**XmlJavaTypeAdapter.DEFAULT** - Class in **javax.xml.bind.annotation.adapters**
Used in `XmlJavaTypeAdapter.type()` to signal that the type be inferred from the signature of the field, property, parameter or the class.

**XmlJavaTypeAdapter.DEFAULT()** - Constructor for class
`javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter.DEFAULT`
XmlJavaTypeAdapters - Annotation Type in javax.xml.bind.annotation.adapters
A container for multiple @XmlJavaTypeAdapter annotations.

XmlList - Annotation Type in javax.xml.bind.annotation
Used to map a property to a list simple type.

XmlMimeType - Annotation Type in javax.xml.bind.annotation
Associates the MIME type that controls the XML representation of the property.

XmlMixed - Annotation Type in javax.xml.bind.annotation
Annotate a JavaBean multi-valued property to support mixed content.

XmlNs - Annotation Type in javax.xml.bind.annotation
Associates a namespace prefix with a XML namespace URI.

XmlNsForm - Enum in javax.xml.bind.annotation
Enumeration of XML Schema namespace qualifications.

XMLOutputFactory - Class in javax.xml.stream
Defines an abstract implementation of a factory for getting XMLEventWriters and XMLStreamWriter.

XMLOutputFactory() - Constructor for class javax.xml.stream.XMLOutputFactory

XmlRegistry - Annotation Type in javax.xml.bind.annotation
Marks a class that has XmlElementDecl.

XMLReporter - Interface in javax.xml.stream
This interface is used to report non-fatal errors.

XMLResolver - Interface in javax.xml.stream
This interface is used to resolve resources during an XML parse.

XmlRootElement - Annotation Type in javax.xml.bind.annotation
Maps a class or an enum type to an XML element.

XmlSchema - Annotation Type in javax.xml.bind.annotation
Maps a package name to a XML namespace.

XmlSchemaType - Annotation Type in javax.xml.bind.annotation
Maps a Java type to a simple schema built-in type.

XmlSchemaType.DEFAULT - Class in javax.xml.bind.annotation
Used in XmlSchemaType.type() to signal that the type be inferred from the signature of the property.

XmlSchemaType.DEFAULT() - Constructor for class javax.xml.bind.annotation.XmlSchemaType.DEFAULT

XmlSchemaTypes - Annotation Type in javax.xml.bind.annotation
A container for multiple @XmlSchemaType annotations.

XMLStreamConstants - Interface in javax.xml.stream
This interface declares the constants used in this API.

XMLStreamException - Exception in javax.xml.stream
The base exception for unexpected processing errors.

XMLStreamException() - Constructor for exception javax.xml.stream.XMLStreamException
Default constructor
**XMLStreamException(String)** - Constructor for exception

`javax.xml.stream.XMLStreamException`  
Construct an exception with the associated message.

**XMLStreamException(throwable)** - Constructor for exception

`javax.xml.stream.XMLStreamException`  
Construct an exception with the associated exception.

**XMLStreamException(String, Throwable)** - Constructor for exception

`javax.xml.stream.XMLStreamException`  
Construct an exception with the associated message and exception.

**XMLStreamException(String, Location, Throwable)** - Constructor for exception

`javax.xml.stream.XMLStreamException`  
Construct an exception with the associated message, exception and location.

**XMLStreamReader** - Interface in `javax.xml.stream`

The `XMLStreamReader` interface allows forward, read-only access to XML.

**XMLStreamWriter** - Interface in `javax.xml.stream`

The `XMLStreamWriter` interface specifies how to write XML.

**XmlTransient** - Annotation Type in `javax.xml.bind.annotation`

Prevents the mapping of a JavaBean property to XML representation.

**XmlType** - Annotation Type in `javax.xml.bind.annotation`

Maps a class or an enum type to a XML Schema type.

**XMLType** - Class in `javax.xml.rpc.encoding`

Constants for common XML Schema and SOAP 1.1 types.

**XMLType()** - Constructor for class `javax.xml.rpc.encoding.XMLType`

**XmlType.DEFAULT** - Class in `javax.xml.bind.annotation`

Used in `XmlType.factoryClass()` to signal that either factory method is not used or that it’s in the class with this `XmlType` itself.

**XmlType.DEFAULT()** - Constructor for class `javax.xml.bind.annotation.XmlType.DEFAULT`

**XmlValue** - Annotation Type in `javax.xml.bind.annotation`

Enables mapping a class to a XML Schema complex type with a simpleContent or a XML Schema simple type.

**XpathEvent** - Class in `javax.enterprise.deploy.model`

An Event class describing ConfigBeans being added/subtracted from a server configuration.

**XpathEvent(DDBean, Object)** - Constructor for class `javax.enterprise.deploy.model.XpathEvent`

A description of a change in the ConfigBean tree.

**XpathListener** - Interface in `javax.enterprise.deploy.model`

The listener interface for receiving XpathEvents

**XSD_BASE64** - Static variable in class
javax.xml.rpc.encoding.XMLType
   The name of the xsd:base64Binary type.
 XSD_BOOLEAN - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:boolean type.
 XSD_BYTE - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:byte type.
 XSD_DATETIME - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:dateTime type.
 XSD_DECIMAL - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:decimal type.
 XSD_DOUBLE - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:double type.
 XSD_FLOAT - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:float type.
 XSD_HEX_BINARY - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:hexBinary type.
 XSD_INT - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:int type.
 XSD_INTEGER - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:integer type.
 XSD_LONG - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:long type.
 XSD_QNAME - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:QName type.
 XSD_SHORT - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:short type.
 XSD_STRING - Static variable in class
class javax.xml.rpc.encoding.XMLType
   The name of the xsd:string type.
getProperty(String) - Method in interface javax.xml.rpc.Stub
 Gets the value of a specific configuration property.

getPropertyNames() - Method in interface javax.xml.rpc.Stub
 Returns an Iterator view of the names of the properties that can be configured on this stub instance.

_jspService(HttpServletRequest, HttpServletResponse) - Method in interface javax.servlet.jsp.HttpJspPage
 The _jspService() method corresponds to the body of the JSP page.

_setProperty(String, Object) - Method in interface javax.xml.rpc.Stub
 Sets the name and value of a configuration property for this Stub instance.
How This API Document Is Organized

This API (Application Programming Interface) document has pages corresponding to the items in the navigation bar, described as follows.

Overview

The Overview page is the front page of this API document and provides a list of all packages with a summary for each. This page can also contain an overall description of the set of packages.

Package

Each package has a page that contains a list of its classes and interfaces, with a summary for each. This page can contain four categories:

- Interfaces (italic)
- Classes
- Enums
- Exceptions
- Errors
- Annotation Types

Class/Interface

Each class, interface, nested class and nested interface has its own separate page. Each of these pages has three sections consisting of a class/interface description, summary tables, and detailed member descriptions:

- Class inheritance diagram
- Direct Subclasses
- All Known Subinterfaces
- All Known Implementing Classes
• Class/interface declaration
• Class/interface description

• Nested Class Summary
• Field Summary
• Constructor Summary
• Method Summary

• Field Detail
• Constructor Detail
• Method Detail

Each summary entry contains the first sentence from the detailed description for that item. The summary entries are alphabetical, while the detailed descriptions are in the order they appear in the source code. This preserves the logical groupings established by the programmer.

Annotation Type

Each annotation type has its own separate page with the following sections:

• Annotation Type declaration
• Annotation Type description
• Required Element Summary
• Optional Element Summary
• Element Detail

Enum

Each enum has its own separate page with the following sections:

• Enum declaration
• Enum description
• Enum Constant Summary
• Enum Constant Detail

Tree (Class Hierarchy)
There is a [Class Hierarchy](#) page for all packages, plus a hierarchy for each package. Each hierarchy page contains a list of classes and a list of interfaces. The classes are organized by inheritance structure starting with `java.lang.Object`. The interfaces do not inherit from `java.lang.Object`.

- When viewing the Overview page, clicking on "Tree" displays the hierarchy for all packages.
- When viewing a particular package, class or interface page, clicking "Tree" displays the hierarchy for only that package.

**Deprecated API**

The [Deprecated API](#) page lists all of the API that have been deprecated. A deprecated API is not recommended for use, generally due to improvements, and a replacement API is usually given. Deprecated APIs may be removed in future implementations.

**Index**

The [Index](#) contains an alphabetic list of all classes, interfaces, constructors, methods, and fields.

**Prev/Next**

These links take you to the next or previous class, interface, package, or related page.

**Frames/No Frames**

These links show and hide the HTML frames. All pages are available with or without frames.

**Serialized Form**

Each serializable or externalizable class has a description of its serialization fields and methods. This information is of interest to re-implementors, not to developers using the API. While there is no link in
the navigation bar, you can get to this information by going to any serialized class and clicking "Serialized Form" in the "See also" section of the class description.

**Constant Field Values**

The [Constant Field Values](#) page lists the static final fields and their values.

*This help file applies to API documentation generated using the standard doclet.*
### Serialized Form

**Package** javax.activation

**Class**

javax.activation.ActivationDataFlavor

extends DataFlavor

implements Serializable

### Serialization Methods

**readExternal**

```java
default void readExternal(ObjectInput arg0)
throws IOException,
ClassNotFoundException
```

**Throws:**

IOException

ClassNotFoundException

**writeExternal**

```java
default void writeExternal(ObjectOutput arg0)
throws IOException
```

**Throws:**

IOException

**Class**

javax.activation.MimeType
extends Object implements Serializable

Serialization Methods

readExternal

```java
public void readExternal(ObjectInput in)
    throws IOException,
        ClassNotFoundException
```

The object implements the readExternal method to restore its contents by calling the methods of DataInput for primitive types and readObject for objects, strings and arrays. The readExternal method must read the values in the same sequence and with the same types as were written by writeExternal.

Throws:

- ClassNotFoundException - If the class for an object being restored cannot be found.
- IOException

writeExternal

```java
public void writeExternal(ObjectOutput out)
    throws IOException
```

The object implements the writeExternal method to save its contents by calling the methods of DataOutput for its primitive values or calling the writeObject method of ObjectOutput for objects, strings and arrays.

Throws:

- IOException - Includes any I/O exceptions that may occur

Class

javax.activation.MimeTypeParseException
extends Exception implements Serializable

Class
javax.activation.UnsupportedDataTypeException
extends IOException implements Serializable

Package javax.ejb

Class javax.ejb.AccessLocalException
extends EJBException implements Serializable

Class javax.ejb.ConcurrentAccessException
extends EJBException implements Serializable

Class javax.ejb.CreateException
extends Exception implements Serializable

Class javax.ejb.DuplicateKeyException
extends CreateException implements Serializable
Class `javax.ejb.EJBAccessException` extends `EJBException` implements `Serializable`

Class `javax.ejb.EJBException` extends `RuntimeException` implements `Serializable`

Serialized Fields

causeException

`Exception` causeException

Class `javax.ejb.EJBTransactionRequiredException` extends `EJBException` implements `Serializable`

Class `javax.ejb.EJBTransactionRolledbackException` extends `EJBException` implements `Serializable`
<table>
<thead>
<tr>
<th>Class</th>
<th>extends</th>
<th>implements</th>
<th>Serializable</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.ejb.FinderException</code></td>
<td><code>Exception</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb.NoSuchEJBException</code></td>
<td><code>EJBException</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb.NoSuchEntityException</code></td>
<td><code>EJBException</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb.NoSuchObjectLocalException</code></td>
<td><code>EJBException</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb.ObjectNotFoundException</code></td>
<td><code>FinderException</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>javax.ejb.RemoveException</code></td>
<td><code>Exception</code></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Class `javax.ejb.TransactionRequiredLocalException` extends `EJBException` implements `Serializable`

Class `javax.ejb.TransactionRolledbackLocalException` extends `EJBException` implements `Serializable`

Package `javax.el`

Class `javax.el.ELContextEvent` extends `EventObject` implements `Serializable`

Class `javax.el.ELException` extends `RuntimeException` implements `Serializable`

Class `javax.el.Expression` extends `Object` implements `Serializable`

Class `javax.el.MethodExpression` extends `Expression` implements
<table>
<thead>
<tr>
<th>Serializable</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
<tr>
<td><strong>Package</strong></td>
</tr>
<tr>
<td><strong>Class</strong></td>
</tr>
</tbody>
</table>
javax.enterprise.deploy.model.exceptions.DDBeanCreateException extends Exception implements Serializable

Package
javax.enterprise.deploy.spi.exceptions

Class
javax.enterprise.deploy.spi.exceptions.BeanNotFoundException extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.ClientExecuteException extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.ConfigurationException extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.DConfigBeanVersionUnsupportedException extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.DeploymentManagerCreationException
extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.InvalidModuleException
extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.OperationUnsupportedException
extends Exception implements Serializable

Class
javax.enterprise.deploy.spi.exceptions.TargetException
extends Exception implements Serializable

Package
javax.enterprise.deploy.spi.status

Class
javax.enterprise.deploy.spi.status.ProgressEvent
extends EventObject implements Serializable

Serialized Fields

statuscode
<table>
<thead>
<tr>
<th>DeploymentStatus</th>
<th>statuscode</th>
</tr>
</thead>
<tbody>
<tr>
<td>targetModuleID</td>
<td></td>
</tr>
<tr>
<td>TargetModuleID</td>
<td>targetModuleID</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>javax.faces</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>javax.faces.FacesException</td>
</tr>
<tr>
<td>extends</td>
<td>RuntimeException</td>
</tr>
<tr>
<td>implements</td>
<td>Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serialized Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>cause</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Throwable</th>
<th>cause</th>
</tr>
</thead>
<tbody>
<tr>
<td>The underlying exception that caused this exception.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>javax.faces.application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>javax.faces.application.FacesMessage</td>
</tr>
<tr>
<td>extends</td>
<td>Object</td>
</tr>
<tr>
<td>implements</td>
<td>Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>serialVersionUID:</th>
<th>-1180773928220076822L</th>
</tr>
</thead>
<tbody>
<tr>
<td>serialVersionUID</td>
<td>-1180773928220076822L</td>
</tr>
</tbody>
</table>

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
</tbody>
</table>
Serialized Fields

severity

FacesMessage.Severity severity

summary

String summary

detail

String detail

Class

javax.faces.application.ViewExpiredException

extends FacesException

implements Serializable

Serialized Fields

viewId

String viewId

The view identifier of the view that could not be restored.

Package javax.faces.convert

Class
| javax.faces.convert.ConverterException | extends FacesException | implements Serializable |

**Serialized Fields**

facesMessage

```java
FacesMessage facesMessage
```

**Package** javax.faces.el

**Class**

| javax.faces.el.EvaluationException | extends FacesException | implements Serializable |

**Class**

| javax.faces.el.MethodNotFoundException | extends EvaluationException | implements Serializable |

**Class**

<p>| javax.faces.el.PropertyNotFoundException | extends EvaluationException | implements Serializable |</p>
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.faces.el.ReferenceSyntaxException</code></td>
<td>extends <code>EvaluationException</code></td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
<tr>
<td><code>javax.faces.event.AbortProcessingException</code></td>
<td>extends <code>FacesException</code></td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
<tr>
<td><code>javax.faces.event.ActionEvent</code></td>
<td>extends <code>FacesEvent</code></td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
<tr>
<td><code>javax.faces.event.FacesEvent</code></td>
<td>extends <code>EventObject</code></td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

Serialized Fields
phaseld

**PhaseId** phaseId

---

**Class** `javax.faces.event.PhaseEvent` extends **EventObject** implements **Serializable**

**Serialized Fields**

```
context
```

**FacesContext** context

---

phaseld

**PhaseId** phaseId

---

**Class** `javax.faces.event.ValueChangeEvent` extends **FacesEvent** implements **Serializable**

**Serialized Fields**

```
oldValue
```

**Object** oldValue

The previous local value of the source **UIComponent**.
The current local value of the source `UIComponent`.

<table>
<thead>
<tr>
<th>Package</th>
<th>javax.faces.model</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td><code>javax.faces.model.DataModelEvent</code> extends <code>EventObject</code> implements Serializable</td>
</tr>
</tbody>
</table>

Serialized Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>data</code></td>
<td><code>Object</code></td>
</tr>
<tr>
<td><code>index</code></td>
<td><code>int</code></td>
</tr>
</tbody>
</table>

Class `javax.faces.model.SelectItem` extends `Object` implements Serializable

`serialVersionUID`: 876782311414654999L

Serialized Fields
description

String description

disabled

boolean disabled

label

String label

value

Object value

escape

boolean escape

Holds value of property escape.

Class

javax.faces.model.SelectItemGroup

extends SelectItem

implements Serializable

Serialized Fields

selectItems
**Package javax.faces.validator**

**Class**

```java
javax.faces.validator.ValidatorException
extends FacesException
implements Serializable
```

**Serialized Fields**

- `message` (FacesMessage)

**Package javax.faces.webapp**

**Class**

```java
javax.faces.webapp.AttributeTag
extends TagSupport
implements Serializable
```

**serialVersionUID**: `-7782950243436672334L`

**Serialized Fields**

- `name` (String)
Deprecated.

The name of the attribute to be created, if not already present.

value

String value

 Deprecated.

The value to be associated with this attribute, if it is created.

Class `javax.faces.webapp.ConverterELTag` extends `TagSupport` implements `Serializable`

Class `javax.faces.webapp.ConverterTag` extends `TagSupport` implements `Serializable`

`serialVersionUID`: -5909792518081427720L

Serialized Fields

`converterId`

String `converterId`

 Deprecated.

The identifier of the `Converter` instance to be created.
String binding

Deprecated.

The ValueExpression that evaluates to an object that implements Converter.

Class javax.faces.webapp.FacetTag extends TagSupport implements Serializable

Serialized Fields

name

String name

The name of this facet. This will be used as the facet name for our UIComponentTag child in our UIComponentTag parent's facet list.

Class javax.faces.webapp.ValidatorELTag extends TagSupport implements Serializable

Class javax.faces.webapp.ValidatorTag extends TagSupport implements Serializable
Serializable

SerialVersionUID: -5562623615418158868L

Serialized Fields

validatorId

String validatorId

Deprecated.

The identifier of the Validator instance to be created.

binding

String binding

Deprecated.

The ValueExpression that evaluates to an object that implements Validator.

Package javax.jms

Class javax.jms.IllegalStateException extends JMSException implements Serializable

Class javax.jms.InvalidClientIDException
`extends JMSException` implements `Serializable`

**Class**

`javax.jms.InvalidDestinationException` extends `JMSException` implements `Serializable`  

**Class**

`javax.jms.InvalidSelectorException` extends `JMSException` implements `Serializable`  

**Class**

`javax.jms.JMSException` extends `Exception` implements `Serializable`

**Serialized Fields**

- `errorCode`
  
  `String` `errorCode`
  
  Vendor-specific error code.

- `linkedException`
  
  `Exception` `linkedException`
Exception reference.

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.jms.JMSSecurityException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JMSEException implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.jms.MessageEOFException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JMSEException implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.jms.MessageFormatException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JMSEException implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.jms.MessageNotReadableException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JMSEException implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.jms.MessageNotWriteableException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JMSEException implements Serializable</td>
</tr>
</tbody>
</table>
Class
javax.jms.ResourceAllocationException extends JMSException implements Serializable

Class
javax.jms.TransactionInProgressException extends JMSException implements Serializable

Class
javax.jms.TransactionRolledBackException extends JMSException implements Serializable

Package javax.mail

Class javax.mail.Address extends Object implements Serializable

serialVersionUID: -5822459626751992278L

Class
javax.mail.AuthenticationFailedException
extends MessagingException implements Serializable

serialVersionUID: 492080754054436511L

Class javax.mail.Flags extends Object implements Serializable

serialVersionUID: 6243590407214169028L

Serialized Fields

system_flags

int system_flags

user_flags

Hashtable<K, V> user_flags

Class javax.mail.FolderClosedException extends MessagingException implements Serializable

serialVersionUID: 1687879213433302315L

Class javax.mail.FolderNotFoundException extends MessagingException
implements Serializable

serialVersionUID: 472612108891249403L

Class javax.mail.IllegalWriteException extends MessagingException implements Serializable

serialVersionUID: 3974370223328268013L

Class javax.mail.Message.RecipientType extends Object implements Serializable

serialVersionUID: -7479791750606340008L

Serialization Methods

readResolve

protected Object readResolve() throws ObjectStreamException

When deserializing a RecipientType, we need to make sure to return only one of the known static final instances defined in this class. Subclasses must implement their own readResolve method that checks for their known instances before calling this super method.

Throws: ObjectStreamException

Serialized Fields

type
String type

The type of recipient, usually the name of a corresponding Internet standard header.

Class `javax.mail.MessageRemovedException`
extends `MessagingException` implements Serializable

serialVersionUID: 1951292550679528690L

Class `javax.mail.MessagingException`
extends `Exception` implements Serializable

serialVersionUID: -7569192289819959253L

Serialized Fields

next

`Exception` next

The next exception in the chain.
extends `MessagingException` implements Serializable

serialVersionUID: -3757386618726131322L

Class `javax.mail.NoSuchProviderException` extends `MessagingException` implements Serializable

serialVersionUID: 8058319293154708827L

Class `javax.mail.ReadOnlyFolderException` extends `MessagingException` implements Serializable

serialVersionUID: 5711829372799039325L

Class `javax.mail.SendFailedException` extends `MessagingException` implements Serializable

serialVersionUID: -6457531621682372913L

Class `javax.mail.StoreClosedException` extends `MessagingException`
implement Serializable

c serialVersionUID: -3145392336120082655L

Package javax.mail.event

Class javax.mail.event.ConnectionEvent extends MailEvent implements Serializable

c serialVersionUID: -1855480171284792957L

Serialized Fields

type

int type

   The event type.

Class javax.mail.event.FolderEvent extends MailEvent implements Serializable

c serialVersionUID: 5278131310563694307L

Serialized Fields
The event type.

Class `javax.mail.event.MailEvent` extends `EventObject` implements `Serializable`

```
serialVersionUID: 1846275636325456631L
```

Class `javax.mail.event.MessageChangedEvent` extends `MailEvent` implements `Serializable`

```
serialVersionUID: -4974972972105535108L
```

Serialized Fields

```
type
```

```
int type
```

The event type.
javax.mail.event.MessageCountEvent extends MailEvent implements Serializable

serialVersionUID: -7447022340837897369L

Serialized Fields

type

int type

The event type.

removed

boolean removed

If true, this event is the result of an explicit expunge by this client, and the messages in this folder have been renumbered to account for this. If false, this event is the result of an expunge by external sources.

Class javax.mail.event.StoreEvent extends MailEvent implements Serializable

serialVersionUID: 1938704919992515330L
Serialized Fields

type

```java
int type
```

The event type.

message

```java
String message
```

The message text to be presented to the user.

Class `javax.mail.event.TransportEvent` extends `MailEvent` implements `Serializable`

`serialVersionUID`: `-4729852364684273073L`

<table>
<thead>
<tr>
<th>Serialized Fields</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
</tr>
</tbody>
</table>

```java
int type
```

The event type.
Package `javax.mail.internet`

Class `javax.mail.internet.AddressException` extends `ParseException` implements `Serializable`

Serialization UID: 9134583443539323120L

Serialized Fields

- `ref`: String
  - The string being parsed.

- `pos`: int
  - The index in the string where the error occurred, or -1 if not known.

Class `javax.mail.internet.InternetAddress` extends `Address` implements `Serializable`
Serialized Fields

address

String address

personal

String personal

The personal name.

encodedPersonal

String encodedPersonal

The RFC 2047 encoded version of the personal name.

This field and the personal field track each other, so if a subclass sets one of these fields directly, it should set the other to null, so that it is suitably recomputed.

Class

javax.mail.internet.MailDateFormat

extends SimpleDateFormat

implements Serializable

serialVersionUID: -8148227605210628779L
Class `javax.mail.internet.MimeMessage.RecipientType` extends `Message.RecipientType` implements `Serializable`

serialVersionUID: -5468290701714395543L

### Serialization Methods

**readResolve**

protected `Object readResolve()` throws `ObjectStreamException`

Throws: `ObjectStreamException`

Class `javax.mail.internet.NewsAddress` extends `Address` implements `Serializable`

serialVersionUID: -4203797299824684143L

### Serialized Fields

`newsgroup`

`String newsgroup`

`host`
String host

Class javax.mail.internet.ParseException extends MessagingException implements Serializable

serialVersionUID: 7649991205183658089L

Package javax.mail.search

Class javax.mail.search.AddressStringTerm extends StringTerm implements Serializable

serialVersionUID: 3086821234204980368L

Class javax.mail.search.AddressTerm extends SearchTerm implements Serializable

serialVersionUID: 2005405551929769980L

Serialized Fields

address
The address.

Class `javax.mail.search.AndTerm` extends `SearchTerm` implements `Serializable`

`serialVersionUID`: -3583274505380989582L

**Serialized Fields**

`terms`

`SearchTerm[] terms`

The array of terms on which the AND operator should be applied.

Class `javax.mail.search.BodyTerm` extends `StringTerm` implements `Serializable`

`serialVersionUID`: -488862527916911385L

Class `javax.mail.search.ComparisonTerm` extends `SearchTerm` implements
Serializable

serialVersionUID: 1456646953666474308L

Serialized Fields

comparison

int comparison

The comparison.

Class **javax.mail.search.DateTerm** extends **ComparisonTerm** implements **Serializable**

serialVersionUID: 4818873430063720043L

Serialized Fields

date

**Date** date

The date.

Class **javax.mail.search.FlagTerm** extends **SearchTerm** implements **Serializable**
### Serializable

**serialVersionUID**: -142991500302030647L

### Serialized Fields

**set**

```java
boolean set
```

Indicates whether to test for the presence or absence of the specified Flag. If `true`, then test whether all the specified flags are present, else test whether all the specified flags are absent.

**flags**

```java
Flags flags
```

Flags object containing the flags to test.

---

### Class `javax.mail.search.FromStringTerm`

**extends**: `AddressStringTerm`

**implements**: `Serializable`

**serialVersionUID**: 5801127523826772788L

### Class `javax.mail.search.FromTerm`

**extends**: `AddressTerm`

**implements**: `Serializable`
Serializable

serialVersionUID: 5214730291502658665L

Class `javax.mail.search.HeaderTerm` extends `StringTerm` implements Serializable

serialVersionUID: 8342514650333389122L

Serialized Fields

`headerName`

`String` `headerName`

The name of the header.

Class `javax.mail.search.IntegerComparisonTerm` extends `ComparisonTerm` implements Serializable

serialVersionUID: -6963571240154302484L

Serialized Fields

`number`
int number

  The number.

Class `javax.mail.search.MessageIDTerm` extends `StringTerm` implements `Serializable`

serialVersionUID: -2121096296454691963L

Class `javax.mail.search.MessageNumberTerm` extends `IntegerComparisonTerm` implements `Serializable`

serialVersionUID: -5379625829658623812L

Class `javax.mail.search.NotTerm` extends `SearchTerm` implements `Serializable`

serialVersionUID: 7152293214217310216L

Serialized Fields

term

  `SearchTerm` term
The search term to negate.

Class `javax.mail.search.OrTerm` extends `SearchTerm` implements Serializable

`serialVersionUID`: 5380534067523646936L

**Serialized Fields**

- `terms`:
  
  `SearchTerm[] terms`

  The array of terms on which the OR operator should be applied.

Class `javax.mail.search.ReceivedDateTerm` extends `DateTerm` implements Serializable

`serialVersionUID`: -2756695246195503170L

Class `javax.mail.search.RecipientStringTerm` extends `AddressStringTerm` implements Serializable
Serializable Fields

**type**

Message.RecipientType  type

The recipient type.

Class `javax.mail.search.RecipientTerm`
extends **AddressTerm**
implements Serializable

Serializable Fields

**type**

Message.RecipientType  type

The recipient type.

Class `javax.mail.search.SearchException`
extends **MessagingException**
implements Serializable
Class **javax.mail.search.SearchTerm** extends **Object** implements **Serializable**

serialVersionUID: -7092886778226268686L

Class **javax.mail.search.SentDateTerm** extends **DateTerm** implements **Serializable**

serialVersionUID: -6652358452205992789L

Class **javax.mail.search.SizeTerm** extends **IntegerComparisonTerm** implements **Serializable**

serialVersionUID: -2556219451005103709L

Class **javax.mail.search.StringTerm** extends **SearchTerm** implements **Serializable**

serialVersionUID: 1274042129007696269L

Serialized Fields

**pattern**
String pattern

The pattern.

ignoreCase

boolean ignoreCase

Ignore case when comparing?

Class `javax.mail.search.SubjectTerm` extends `StringTerm` implements `Serializable`

serialVersionUID: 7481568618055573432L

Package `javax.management.j2ee`

Package `javax.persistence`

Class `javax.persistence.EntityExistsException` extends `PersistenceException` implements `Serializable`
<table>
<thead>
<tr>
<th>Class</th>
<th><code>javax.persistence.EntityNotFoundException</code> extends <code>PersistenceException</code> implements <code>Serializable</code></th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td><code>javax.persistence.NonUniqueResultException</code> extends <code>PersistenceException</code> implements <code>Serializable</code></td>
</tr>
<tr>
<td>Class</td>
<td><code>javax.persistence.NoResultException</code> extends <code>PersistenceException</code> implements <code>Serializable</code></td>
</tr>
<tr>
<td>Class</td>
<td><code>javax.persistence.OptimisticLockException</code> extends <code>PersistenceException</code> implements <code>Serializable</code></td>
</tr>
</tbody>
</table>

**Serialized Fields**

- `entity Object` entity

  The object that caused the exception
Class
`javax.persistence.PersistenceException` extends `RuntimeException` implements `Serializable`

Class
`javax.persistence.RollbackException` extends `PersistenceException` implements `Serializable`

Class
`javax.persistence.TransactionRequiredException` extends `PersistenceException` implements `Serializable`

Package `javax.resource`

Class
`javax.resource.NotSupportedException` extends `ResourceException` implements `Serializable`
**javax.resource.ResourceException**
extends **Exception** implements **Serializable**

**Serialized Fields**

**errorCode**

`String errorCode`

Vendor specific error code

**linkedException**

`Exception linkedException`

Reference to another exception

---

**Package javax.resource.cci**

**Class**

`javax.resource.cci.ResourceWarning`
extends **ResourceException** implements **Serializable**

---

**Package javax.resource.spi**

**Class**
<table>
<thead>
<tr>
<th>Class</th>
<th>javax.resource.spi.ApplicationServerInternalException extends ResourceException implements Serializable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>javax.resource.spi.CommException extends ResourceException implements Serializable</td>
</tr>
<tr>
<td></td>
<td>Class javax.resource.spi.ConnectionEvent extends EventObject implements Serializable</td>
</tr>
</tbody>
</table>

**Serialized Fields**

**exception**

Exception associated with the ConnectionEvent instance.

**id**

int id

Type of the event
connectionHandle

Object connectionHandle

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.resource.spi.EISSystemException</th>
<th>extends ResourceException</th>
<th>implements Serializable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>javax.resource.spi.IllegalStateException</td>
<td>extends ResourceException</td>
<td>implements Serializable</td>
</tr>
<tr>
<td>Class</td>
<td>javax.resource.spi.InvalidPropertyException</td>
<td>extends ResourceException</td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

Serialized Fields

invalidProperties

PropertyDescriptor[] invalidProperties

| Class                      | javax.resource.spi.LocalTransactionException | extends ResourceException | implements Serializable |
extends ResourceException implements Serializable

Class
javax.resource.spi.ResourceAdapterInternalException
extends ResourceException implements Serializable

Class
javax.resource.spi.ResourceAllocationException
extends ResourceException implements Serializable

Class
javax.resource.spi.SecurityException
extends ResourceException implements Serializable

Class
javax.resource.spi.SharingViolationException
extends ResourceException implements Serializable

Class
javax.resource.spi.UnavailableException
extends ResourceException implements Serializable

Package javax.resource.spi.security

Class javax.resource.spi.security.PasswordCredential
extends Object implements Serializable

Serialized Fields

userName
String userName

password
char[] password

mcf
ManagedConnectionFactory mcf

Package javax.resource.spi.work

Class javax.resource.spi.work.WorkCompletedException
extends `WorkException` implements `Serializable`

Class
`javax.resource.spi.work.WorkEvent`
extends `EventObject` implements `Serializable`

Serialized Fields

```plaintext
type

int type

The event type.

work

Work work

The work object on which the event occurred.

exc

WorkException exc

The exception that occurred during work processing.

startDuration

long startDuration
```
The start delay duration (in milliseconds).

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.resource.spi.work.WorkException extends ResourceException implements Serializable</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>javax.resource.spi.work.WorkRejectedException extends WorkException implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>javax.security.jacc</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.security.jacc.EJBMethodPermission extends Permission implements Serializable</th>
</tr>
</thead>
</table>

serialVersionUID: 1L

**Serialization Methods**

readObject

```java
private void readObject(ObjectInputStream s) throws IOException, ClassNotFoundException {
    // Implementation
}
```
readObject reads the serialized fields from the input stream and uses them to restore the permission. This method need not be implemented if establishing the values of the serialized fields (as is done by defaultReadObject) is sufficient to initialize the permission.

**Throws:**
- `IOException`
- `ClassNotFoundException`

### writeObject

```java
private void writeObject(ObjectOutputStream s) throws IOException
```

writeObject is used to establish the values of the serialized fields before they are written to the output stream and need not be implemented if the values of the serialized fields are always available and up to date. The serialized fields are written to the output stream in the same form as they would be written by defaultWriteObject.

**Throws:**
- `IOException`

### Serialization Overview

The serialized fields of this permission are defined below. Whether or not the serialized fields correspond to actual (private) fields is an implementation decision.

### Serialized Fields

**actions**

```java
String actions
```

the canonicalized actions string (as returned by getActions).
Class
javax.security.jacc.EJBRoleRefPermission
extends Permission implements Serializable

serialVersionUID: 1L

Serialization Methods

readObject

private void readObject(ObjectInputStream s)
    throws IOException, ClassNotFoundException

readObject reads the serialized fields from the input stream and uses them to restore the permission. This method need not be implemented if establishing the values of the serialized fields (as is done by defaultReadObject) is sufficient to initialize the permission.

    Throws:
    IOException
    ClassNotFoundException

writeObject

private void writeObject(ObjectOutputStream s)
    throws IOException

writeObject is used to establish the values of the serialized fields before they are written to the output stream and need not be implemented if the values of the serialized fields are always available and up to date. The serialized fields are written to the output stream in the same form as they would be written by defaultWriteObject.

    Throws:
IOException

Serialization Overview

The serialized fields of this permission are defined below. Whether or not the serialized fields correspond to actual (private) fields is an implementation decision.

Serialized Fields

actions

String actions

the canonicalized actions string (as returned by getActions).

Class

class javax.security.jacc.PolicyContextException
extends Exception implements Serializable

serialVersionUID: 1L

Serialization Methods

readObject

private void readObject(ObjectInputStream s)
throws IOException,
ClassNotFoundException

readObject reads the serialized fields from the input stream and uses them to restore the permission. This method need not be implemented if establishing the values of the serialized fields (as is done by defaultReadObject) is sufficient to initialize the permission.

Throws:
   IOException
   ClassNotFoundException

writeObject

private void writeObject(ObjectOutputStream s) throws IOException

writeObject is used to establish the values of the serialized fields before they are written to the output stream and need not be implemented if the values of the serialized fields are always available and up to date. The serialized fields are written to the output stream in the same form as they would be written by defaultWriteObject.

Throws:
   IOException

Serialization Overview

The serialized fields of this permission are defined below. Whether or not the serialized fields correspond to actual (private) fields is an implementation decision.

Serialized Fields

actions

String actions

the canonicalized actions string (as returned by getActions).
Class
javax.security.jacc.WebRoleRefPermission
extends Permission implements Serializable

serialVersionUID: 1L

Serialization Methods

readObject

private void readObject(ObjectInputStream s)
throws IOException, ClassNotFoundException

readObject reads the serialized fields from the input stream and uses them to restore the permission. This method need not be implemented if establishing the values of the serialized fields (as is done by defaultReadObject) is sufficient to initialize the permission.

Throws:
  IOException
  ClassNotFoundException

writeObject

private void writeObject(ObjectOutputStream s)
throws IOException

writeObject is used to establish the values of the serialized fields before they are written to the output stream and need not be implemented if the values of the serialized fields are always available and up to date. The serialized fields are written to the output stream in the same form as they would be written by defaultWriteObject.
Serialization Overview

The serialized fields of this permission are defined below. Whether or not the serialized fields correspond to actual (private) fields is an implementation decision.

Serialized Fields

actions

String actions

the canonicalized actions string (as returned by getActions).

Class

javax.security.jacc.WebUserDataPermission
extends Permission implements Serializable

serialVersionUID: 1L

Serialization Methods

readObject

private void readObject(ObjectInputStream s)
throws IOException, ClassNotFoundException

readObject reads the serialized fields from the input stream and uses them to restore the permission. This method need not be
implemented if establishing the values of the serialized fields (as is done by defaultReadObject) is sufficient to initialize the permission.

Throws:

- `IOException`
- `ClassNotFoundException`

writeObject

```java
private void writeObject(ObjectOutputStream s) throws IOException
```

writeObject is used to establish the values of the serialized fields before they are written to the output stream and need not be implemented if the values of the serialized fields are always available and up to date. The serialized fields are written to the output stream in the same form as they would be written by defaultWriteObject.

Throws:

- `IOException`

### Serialization Overview

The serialized fields of this permission are defined below. Whether or not the serialized fields correspond to actual (private) fields is an implementation decision.

### Serialized Fields

**actions**

```java
String actions
```

the canonicalized actions string (as returned by getActions).

Package `javax.servlet`
Class `javax.servlet.GenericServlet` extends `Object` implements `Serializable`
## Serialized Fields

**rootCause**

*Throwable* rootCause

### Class

`javax.servlet.ServletRequestAttributeEvent`

extends `ServletRequestEvent`

implements `Serializable`

### Serialized Fields

**name**

*String* name

**value**

*Object* value

### Class `javax.servlet.ServletRequestEvent`

extends `EventObject`

implements `Serializable`

### Serialized Fields

**request**

*ServletRequest* request
Class
javax.servletUnavailableException
extends ServletException implements Serializable

Serialized Fields

servlet
Servlet servlet

permanent
boolean permanent

seconds
int seconds

Package javax.servlet.http

Class javax.servlet.http.HttpServlet
extends GenericServlet implements Serializable

Class javax.servlet.http.HttpSessionBindingEvent
extends HttpSessionEvent implements Serializable

Serialized Fields

name

String name

value

Object value

Class
javax.servlet.http.HttpSessionEvent
extends EventObject implements Serializable

Package javax.servlet.jsp

Class javax.servlet.jsp.JspException
extends Exception implements Serializable

Class javax.servlet.jsp.JspTagException
extends JspException implements
### Serializable

#### Class

**javax.servlet.jsp.SkipPageException**
- extends **JspException**
- implements Serializable

#### Package javax.servlet.jsp.el

#### Class **javax.servlet.jsp.el.ELException**
- extends **Exception**
- implements Serializable

#### Serialized Fields

- **mRootCause**
  - **Throwable**
  - Deprecated.

#### Class **javax.servlet.jsp.el.ELParseException**
- extends **ELException**
- implements Serializable
Package `javax.servlet.jsp.tagext`

Class
`javax.servlet.jsp.tagext.BodyTagSupport`
extends `TagSupport` implements `Serializable`

Serialized Fields

`bodyContent`

`BodyContent bodyContent`

The current BodyContent for this BodyTag.

Class
`javax.servlet.jsp.tagext.TagSupport`
extends `Object` implements `Serializable`

Serialized Fields

`parent`

`Tag parent`

`values`

`Hashtable<K, V> values`
**id**

*String* id

The value of the id attribute of this tag; or null.

---

**pageContext**

*PageContext* pageContext

The PageContext.

---

**Package javax.transaction**

**Class**

*javax.transaction.HeuristicCommitException* extends *Exception* implements *Serializable*

**Class**

*javax.transaction.HeuristicMixedException* extends *Exception* implements *Serializable*

**Class**

*javax.transaction.HeuristicRollbackException* extends *Exception* implements *Serializable*
javax.transaction.InvalidTransactionException extends RemoteException implements Serializable

Class
javax.transaction.NotSupportedException extends Exception implements Serializable

Class
javax.transaction.RollbackException extends Exception implements Serializable

Class
javax.transaction.SystemException extends Exception implements Serializable

Serialized Fields

errorCode

int errorCode

The error code with which to create the SystemException.
The error code for the exception

Class
javafx.transaction.TransactionRequiredException extends RemoteException implements Serializable

Class
javafx.transaction.TransactionRolledbackException extends RemoteException implements Serializable

Package javax.transaction.xa

Class javafx.transaction.xa.XAException extends Exception implements Serializable

Serialized Fields

errorCode

int errorCode

The error code with which to create the SystemException.

The error code for the exception.
### Package javax.xml.bind

### Class

**javax.xml.bind.DataBindingException**

extends **RuntimeException**

implements Serializable

### Class **javax.xml.bind.JAXBElement**

extends **Object**

implements Serializable

serialVersionUID: 1L

### Serialized Fields

#### name

**QName** name

xml element tag name

#### declaredType

**Class<T>** declaredType

Java datatype binding for xml element declaration's type.

#### scope

**Class<T>** scope

Scope of xml element declaration representing this xml element
instance. Can be one of the following values: -
[JAXBElement.GlobalScope](https://javaeespecs.org/jakarta-ee/8/JAX-WS/jax-ws-impl/impl-api-spec.html#GlobalScope) for global xml element declaration. - local element declaration has a scope set to the Java class representation of complex type defintion containing xml element declaration.

---

**value**

[Object](https://docs.oracle.com/javase/8/docs/api/java/lang/Object.html) value

xml element value. Represents content model and attributes of an xml element instance.

---

**nil**

[boolean](https://docs.oracle.com/javase/8/docs/api/java/lang/Boolean.html) nil

true iff the xml element instance has xsi:nil="true".

---

Class [javadoc.xml.bind.JAXBException](https://javaeespecs.org/jakarta-ee/8/JAX-WS/jax-ws-impl/impl-api-spec.html#JAXBException) extends [Exception](https://docs.oracle.com/javase/8/docs/api/java/lang/Exception.html) implements [Serializable](https://docs.oracle.com/javase/8/docs/api/java/io/Serializable.html)

serialVersionUID: -5621384651494307979L

---

**Serialized Fields**

**errorCode**

[String](https://docs.oracle.com/javase/8/docs/api/java/lang/String.html) errorCode

Vendor specific error code

---

**linkedException**
Throwable

Exception reference

Class `javax.xml.bind.MarshalException` extends `JAXBException` implements `Serializable`

Class `javax.xml.bind.PropertyException` extends `JAXBException` implements `Serializable`

Class `javax.xml.bind.TypeConstraintException` extends `RuntimeException` implements `Serializable`

Serialized Fields

`errorCode`  

`String errorCode`  

Vendor specific error code

`linkedException`

`Throwable linkedException`  

Exception reference
<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.bind.UnmarshalException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JAXBException</td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.bind.ValidationException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JAXBException</td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

| Package | javax.xml.registry |

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.registry.DeleteException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends RegistryException</td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.registry.FindException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends RegistryException</td>
</tr>
<tr>
<td></td>
<td>implements Serializable</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.registry.InvalidRequestException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JAXRException</td>
</tr>
<tr>
<td></td>
<td>implements</td>
</tr>
<tr>
<td>Serializable</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td></td>
</tr>
</tbody>
</table>

**Class** `javax.xml.registry.JAXRException` extends `Exception` implements `Serializable`

**Serialized Fields**

- `cause` is a `Throwable`

**Class** `javax.xml.registry.RegistryException` extends `JAXRException` implements `Serializable`

**Serialized Fields**

- `errorObjectKey` is a `Key`

**Class** `javax.xml.registry.SaveException` extends `RegistryException` implements `Serializable`
<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.registry.UnexpectedObjectException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JAXRException</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.registry.UnsupportedCapabilityException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends JAXRException</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Package</th>
<th>javax.xml.rpc</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.rpc.JAXRPCException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends RuntimeException</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Serialized Fields</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>cause</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Throwable cause</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>javax.xml.rpc.ServiceException</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>extends Exception</td>
</tr>
</tbody>
</table>
### Serialized Fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>cause</code></td>
<td><code>Throwable</code></td>
<td></td>
</tr>
</tbody>
</table>

```java
class HandlerInfo extends Object implements Serializable {
    // Fields defined here...
}
```
Class
javax.xml.rpc.soap.SOAPFaultException
extends RuntimeException implements Serializable

Serialized Fields

faultcode
QName faultcode

faultstring
String faultstring

cfaultactor
String faultactor

detail
Detail detail

Package javax.xml.soap

Class javax.xml.soap.SOAPException
extends Exception implements Serializable
Serialized Fields

cause

Throwable cause

Package javax.xml.stream

Class javax.xml.stream.FactoryConfigurationException
extends Error implements Serializable

Serialized Fields

nested

Exception nested

Class javax.xml.stream.XMLStreamException
extends Exception implements Serializable

Serialized Fields

nested

Throwable nested
### Package `javax.xml.ws`

Class `javax.xml.ws.ProtocolException`<br>extends `WebServiceException`<br>implements `Serializable`  

Class `javax.xml.ws.WebServiceException`<br>extends `RuntimeException`<br>implements `Serializable`  

Class `javax.xml.ws.WebServicePermission`<br>extends `BasicPermission`<br>implements `Serializable`  

serialVersionUID: -146474640053770988L

### Package `javax.xml.ws.http`

Class `javax.xml.ws.http.HTTPException`<br>extends `ProtocolException`<br>
Serializable

Serialized Fields

statusCode
int statusCode

Package javax.xml.ws.soap

Class
javax.xml.ws.soap.SOAPFaultException
extends ProtocolException
implements Serializable

Serialized Fields

fault
SOAPFault fault

Overview Package Class Tree Deprecated Index Help
PREV NEXT FRAMES NO FRAMES

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.xml.bind.helpers`

**JAXB Provider Use Only:** Provides partial default implementations for some of the `javax.xml.bind` interfaces.

See:  [Description](#)

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AbstractMarshallerImpl</strong></td>
<td>Partial default Marshaller implementation.</td>
</tr>
<tr>
<td><strong>AbstractUnmarshallerImpl</strong></td>
<td>Partial default Unmarshaller implementation.</td>
</tr>
<tr>
<td><strong>DefaultValidationEventHandler</strong></td>
<td>JAXB 1.0 only default validation event handler.</td>
</tr>
<tr>
<td><strong>NotIdentifiableEventImpl</strong></td>
<td>Default implementation of the NotIdentifiableEvent interface.</td>
</tr>
<tr>
<td><strong>ParseConversionEventImpl</strong></td>
<td>Default implementation of the ParseConversionEvent interface.</td>
</tr>
<tr>
<td><strong>PrintConversionEventImpl</strong></td>
<td>Default implementation of the PrintConversionEvent interface.</td>
</tr>
<tr>
<td><strong>ValidationEventImpl</strong></td>
<td>Default implementation of the ValidationEvent interface.</td>
</tr>
<tr>
<td><strong>ValidationEventLocatorImpl</strong></td>
<td>Default implementation of the ValidationEventLocator interface.</td>
</tr>
</tbody>
</table>
Package javax.xml.bind.helpers Description

**JAXB Provider Use Only:** Provides partial default implementations for some of the `javax.xml.bind` interfaces.

JAXB Providers can extend these classes and implement the abstract methods.
Package Specification

- JAXB Specification
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- The [JAXB Website](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package javax.xml.bind.helpers

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.bind.helpers.**AbstractMarshallerImpl** (implements javax.xml.bind.**Marshaller**)
  - javax.xml.bind.helpers.**AbstractUnmarshallerImpl** (implements javax.xml.bind.**Unmarshaller**)
  - javax.xml.bind.helpers.**DefaultValidationEventHandler** (implements javax.xml.bind.**ValidationEventHandler**)
  - javax.xml.bind.helpers.**ValidationEventImpl** (implements javax.xml.bind.**ValidationEvent**)
    - javax.xml.bind.helpers.**NotIdentifiableEventImpl** (implements javax.xml.bind.**NotIdentifiableEvent**)
    - javax.xml.bind.helpers.**ParseConversionEventImpl** (implements javax.xml.bind.**ParseConversionEvent**)
    - javax.xml.bind.helpers.**PrintConversionEventImpl** (implements javax.xml.bind.**PrintConversionEvent**)
  - javax.xml.bind.helpers.**ValidationEventLocatorImpl** (implements javax.xml.bind.**ValidationEventLocator**)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
## Package javax.ejb

The javax.ejb package contains the Enterprise JavaBeans classes and interfaces that define the contracts between the enterprise bean and its clients and between the enterprise bean and the EJB container.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EJBContext</strong></td>
<td>The EJBContext interface provides an instance with access to the container-provided runtime context of an enterprise Bean instance.</td>
</tr>
<tr>
<td><strong>EJBHome</strong></td>
<td>The EJBHome interface must be extended by all enterprise Beans' remote home interfaces.</td>
</tr>
<tr>
<td><strong>EJBLocalHome</strong></td>
<td>The EJBLocalHome interface must be extended by all enterprise Beans' local home interfaces.</td>
</tr>
<tr>
<td><strong>EJBLocalObject</strong></td>
<td>The EJBLocalObject interface must be extended by all enterprise Beans' local interfaces.</td>
</tr>
<tr>
<td><strong>EJBMetaData</strong></td>
<td>The EJBMetaData interface allows a client to obtain the enterprise Bean's meta-data information.</td>
</tr>
<tr>
<td><strong>EJBOBJECT</strong></td>
<td>The EJBOBJECT interface is extended by all enterprise Beans' remote interfaces.</td>
</tr>
<tr>
<td><strong>EnterpriseBean</strong></td>
<td>The EnterpriseBean interface must be implemented by every enterprise Bean class.</td>
</tr>
<tr>
<td><strong>EntityBean</strong></td>
<td>The EntityBean interface is implemented by every entity enterprise Bean class.</td>
</tr>
<tr>
<td><strong>EntityContext</strong></td>
<td>The EntityContext interface provides an instance with access to the container-provided runtime context of an entity</td>
</tr>
<tr>
<td>Interface</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Handle</strong></td>
<td>The Handle interface is implemented by all EJB object handles.</td>
</tr>
<tr>
<td><strong>HomeHandle</strong></td>
<td>The HomeHandle interface is implemented by all home object handles.</td>
</tr>
<tr>
<td><strong>MessageDrivenBean</strong></td>
<td>The MessageDrivenBean interface is implemented by every message-driven enterprise Bean class.</td>
</tr>
<tr>
<td><strong>MessageDrivenContext</strong></td>
<td>The MessageDrivenContext interface provides access to the runtime message-driven context that the container provides for a message-driven enterprise Bean instance.</td>
</tr>
<tr>
<td><strong>SessionBean</strong></td>
<td>The SessionBean interface is implemented by every session enterprise Bean class.</td>
</tr>
<tr>
<td><strong>SessionContext</strong></td>
<td>The SessionContext interface provides access to the runtime session context that the container provides for a session enterprise Bean instance.</td>
</tr>
<tr>
<td><strong>SessionSynchronization</strong></td>
<td>The SessionSynchronization interface allows a session Bean instance to be notified by its container of transaction boundaries.</td>
</tr>
<tr>
<td><strong>TimedObject</strong></td>
<td>The TimedObject interface contains the callback method that is used to deliver timer expiration notifications.</td>
</tr>
<tr>
<td><strong>Timer</strong></td>
<td>The Timer interface contains information about a timer that was created through the EJB Timer Service.</td>
</tr>
<tr>
<td><strong>TimerHandle</strong></td>
<td>The TimerHandle interface is implemented by all EJB timer handles.</td>
</tr>
<tr>
<td><strong>TimerService</strong></td>
<td>The TimerService interface provides enterprise bean components with access to the container-provided Timer Service.</td>
</tr>
</tbody>
</table>

**Enum Summary**
<table>
<thead>
<tr>
<th>Exception Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AccessLocalException</strong></td>
</tr>
<tr>
<td><strong>ConcurrentAccessException</strong></td>
</tr>
<tr>
<td><strong>CreateException</strong></td>
</tr>
<tr>
<td><strong>DuplicateKeyException</strong></td>
</tr>
<tr>
<td><strong>EJBAccessException</strong></td>
</tr>
<tr>
<td><strong>EJBException</strong></td>
</tr>
<tr>
<td>Exception</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td><strong>EJBException</strong></td>
</tr>
<tr>
<td><strong>EJBTransactionRequiredException</strong></td>
</tr>
<tr>
<td><strong>EJBTransactionRolledbackException</strong></td>
</tr>
<tr>
<td><strong>FinderException</strong></td>
</tr>
<tr>
<td><strong>NoSuchEJBException</strong></td>
</tr>
<tr>
<td><strong>NoSuchEntityException</strong></td>
</tr>
<tr>
<td><strong>NoSuchObjectLocalException</strong></td>
</tr>
<tr>
<td>Exception Type</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
</tr>
<tr>
<td><strong>NoSuchObjectLocalException</strong></td>
</tr>
<tr>
<td><strong>ObjectNotFoundException</strong></td>
</tr>
<tr>
<td><strong>RemoveException</strong></td>
</tr>
<tr>
<td><strong>TransactionRequiredLocalException</strong></td>
</tr>
<tr>
<td><strong>TransactionRolledbackLocalException</strong></td>
</tr>
</tbody>
</table>

**Annotation Types Summary**

<table>
<thead>
<tr>
<th>Annotation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActivationConfigProperty</strong></td>
<td>Applied to an exception to denote that it is an application exception and should be reported to the client directly (i.e., unwrapped).</td>
</tr>
<tr>
<td><strong>ApplicationException</strong></td>
<td>Indicates a dependency on the local or JNDI namespace.</td>
</tr>
<tr>
<td><strong>EJB</strong></td>
<td>Indicates a dependency on the local or JNDI namespace.</td>
</tr>
</tbody>
</table>
A remote view of an Enterprise Java Bean.

<table>
<thead>
<tr>
<th><strong>EJBs</strong></th>
<th>Declares multiple TYPE-level @EJB annotations.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Init</strong></td>
<td>Designates a method of a session bean that corresponds to the create method of an adapted Home interface or an adapted Local Home interface.</td>
</tr>
<tr>
<td><strong>Local</strong></td>
<td>When used on the bean class, declares the local business interface(s) for a session bean.</td>
</tr>
<tr>
<td><strong>LocalHome</strong></td>
<td>Declares the Local Home or adapted Local Home interface for a session bean.</td>
</tr>
<tr>
<td><strong>MessageDriven</strong></td>
<td>Component-defining annotation for a message driven bean.</td>
</tr>
<tr>
<td><strong>PostActivate</strong></td>
<td>Designates a method to receive a callback after a stateful session bean has been activated.</td>
</tr>
<tr>
<td><strong>PrePassivate</strong></td>
<td>Designates a method to receive a callback before a stateful session bean is passivated.</td>
</tr>
<tr>
<td><strong>Remote</strong></td>
<td>Declares the remote business interface(s) for a session bean.</td>
</tr>
<tr>
<td><strong>RemoteHome</strong></td>
<td>Declares the Remote Home or adapted Remote Home interface for a session bean.</td>
</tr>
<tr>
<td><strong>Remove</strong></td>
<td>Applied to a business method of a stateful session bean class.</td>
</tr>
<tr>
<td><strong>Stateful</strong></td>
<td>Component-defining annotation for a stateful session bean.</td>
</tr>
<tr>
<td><strong>Stateless</strong></td>
<td>Component-defining annotation for a stateless session bean.</td>
</tr>
<tr>
<td><strong>Timeout</strong></td>
<td>Designates a method on a stateless session bean class or message driven bean class that should receive EJB timer expirations for that bean.</td>
</tr>
</tbody>
</table>

When applied at the TYPE-level, designates
<table>
<thead>
<tr>
<th><strong>TransactionAttribute</strong></th>
<th>The default transaction attribute for all business methods of the session or message driven bean.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TransactionManagement</strong></td>
<td>Declares whether a session bean or message driven bean has container managed transactions or bean managed transactions.</td>
</tr>
</tbody>
</table>
The `javax.ejb` package contains the Enterprise JavaBeans classes and interfaces that define the contracts between the enterprise bean and its clients and between the enterprise bean and the EJB container.
Hierarchy For Package javax.ejb

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.ejb.**CreateException**
      - javax.ejb.**DuplicateKeyException**
      - javax.ejb.**FinderException**
      - javax.ejb.**ObjectNotFoundException**
      - javax.ejb.**RemoveException**
      - java.lang.**RuntimeException**
    - javax.ejb.**EJBException**
      - javax.ejb.**AccessLocalException**
      - javax.ejb.**ConcurrentAccessException**
      - javax.ejb.**EJBAccessException**
      - javax.ejb.**EJBTransactionRequiredException**
      - javax.ejb.**EJBTransactionRolledbackException**
      - javax.ejb.**NoSuchEJBException**
      - javax.ejb.**NoSuchEntityException**
      - javax.ejb.**NoSuchObjectLocalException**
      - javax.ejb.**TransactionRequiredLocalException**
      - javax.ejb.**TransactionRolledbackLocalException**
Interface Hierarchy

- `javax.ejb.EJBContext`
  - `javax.ejb.EntityContext`
  - `javax.ejb.MessageDrivenContext`
  - `javax.ejb.SessionContext`
- `javax.ejb.EJBLocalHome`
- `javax.ejb.EJBLocalObject`
- `javax.ejb.EJBMetaData`
- `javax.ejb.Handle`
- `javax.ejb.HomeHandle`
- `javax.ejb.TimerHandle`
- `javax.ejb.SessionSynchronization`
- `javax.ejb.TimedObject`
- `javax.ejb.Timer`
- `javax.ejb.TimerService`
Annotation Type Hierarchy

- `javax.ejb.ActivationConfigProperty` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.ApplicationException` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.EJB` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.EJBs` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Init` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Local` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.LocalHome` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.MessageDriven` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.PostActivate` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.PrePassivate` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Remote` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.RemoteHome` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Remove` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Stateful` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Stateless` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.Timeout` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.TransactionAttribute` (implements `java.lang.annotation.Annotation`)
- `javax.ejb.TransactionManagement` (implements `java.lang.annotation.Annotation`
# Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>,
    java.io.**Serializable**)  
    - javax.ejb.**TransactionAttributeType**
    - javax.ejb.**TransactionManagementType**
Package javax.faces.component

Fundamental APIs for user interface components.

See:   Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActionSource</strong></td>
</tr>
<tr>
<td>ActionSource is an interface that may be implemented by any concrete UIComponent that wishes to be a source of ActionEvent, including the ability to invoke application actions via the default ActionListener mechanism.</td>
</tr>
<tr>
<td><strong>ActionSource2</strong></td>
</tr>
<tr>
<td>ActionSource2 extends ActionSource and provides a JavaBeans property analogous to the &quot;action&quot; property on ActionSource.</td>
</tr>
<tr>
<td><strong>ContextCallback</strong></td>
</tr>
<tr>
<td>A simple callback interface that enables taking action on a specific UIComponent (either facet or child) in the view while preserving any contextual state for that component instance in the view.</td>
</tr>
<tr>
<td><strong>EditableValueHolder</strong></td>
</tr>
<tr>
<td>EditableValueHolder is an extension of ValueHolder that describes additional features supported by editable components, including ValueChangeEvent and Validators.</td>
</tr>
<tr>
<td><strong>NamingContainer</strong></td>
</tr>
<tr>
<td>NamingContainer is an interface that must be implemented by any UIComponent that wants to be a naming container.</td>
</tr>
<tr>
<td><strong>StateHolder</strong></td>
</tr>
<tr>
<td>This interface is implemented by classes that need to save their state between requests.</td>
</tr>
<tr>
<td><strong>ValueHolder</strong></td>
</tr>
<tr>
<td>ValueHolder is an interface that may be implemented by any concrete UIComponent that wishes to support a local value, as well as access data in the model tier via a value expression, and support conversion between String and the model tier data's native data type.</td>
</tr>
<tr>
<td>Class Summary</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td><strong>UIColumn</strong></td>
</tr>
<tr>
<td><strong>UICommand</strong></td>
</tr>
<tr>
<td><strong>UIComponent</strong></td>
</tr>
<tr>
<td><strong>UIComponentBase</strong></td>
</tr>
<tr>
<td><strong>UIData</strong></td>
</tr>
<tr>
<td><strong>UIForm</strong></td>
</tr>
<tr>
<td><strong>UIGraphic</strong></td>
</tr>
<tr>
<td><strong>UIInput</strong></td>
</tr>
<tr>
<td><strong>UIMessage</strong></td>
</tr>
<tr>
<td><strong>UIMessages</strong></td>
</tr>
<tr>
<td><strong>UINamingContainer</strong></td>
</tr>
<tr>
<td><strong>UIOutput</strong></td>
</tr>
<tr>
<td><strong>UIPanel</strong></td>
</tr>
<tr>
<td><strong>UIParameter</strong></td>
</tr>
<tr>
<td><strong>UISelectBoolean</strong></td>
</tr>
<tr>
<td><strong>UISelectItem</strong></td>
</tr>
<tr>
<td><strong>UISelectItems</strong></td>
</tr>
<tr>
<td><strong>UISelectMany</strong></td>
</tr>
<tr>
<td><strong>UISelectOne</strong></td>
</tr>
<tr>
<td><strong>UIViewRoot</strong></td>
</tr>
</tbody>
</table>
Package javax.faces.component Description

Fundamental APIs for user interface components.

For your convenience here is a UML class diagram of the classes in this package.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.faces.component

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.component.**UIComponent** (implements javax.faces.component.**StateHolder**)
  - javax.faces.component.**UIComponentBase**
    - javax.faces.component.**UIColumn**
    - javax.faces.component.**UICommand** (implements javax.faces.component.**ActionSource2**)
    - javax.faces.component.**UIData** (implements javax.faces.component.**NamingContainer**)
    - javax.faces.component.**UIForm** (implements javax.faces.component.**NamingContainer**)
    - javax.faces.component.**UIGraphic**
    - javax.faces.component.**UIMessage**
    - javax.faces.component.**UIMessages**
    - javax.faces.component.**UINamingContainer** (implements javax.faces.component.**NamingContainer**)
    - javax.faces.component.**UIOutput** (implements javax.faces.component.**ValueHolder**)
      - javax.faces.component.**UIInput** (implements javax.faces.component.**EditableValueHolder**)
        - javax.faces.component.**UISelectBoolean**
        - javax.faces.component.**UISelectMany**
        - javax.faces.component.**UISelectOne**
    - javax.faces.component.**UIPanel**
    - javax.faces.component.**UIParameter**
    - javax.faces.component.**UISelectItem**
    - javax.faces.component.**UISelectItems**
    - javax.faces.component.**UIViewRoot**
Interface Hierarchy

- javax.faces.component.ActionSource
  - javax.faces.component.ActionSource2
- javax.faces.component.ContextCallback
- javax.faces.component.NamingContainer
- javax.faces.component.StateHolder
- javax.faces.component.ValueHolder
  - javax.faces.componentEditableValueHolder

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.enterprise.deploy.shared`

Provides shared objects for Tool Vendor and Product Vendor implementation classes.

See: [Description](#)

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActionType</strong></td>
<td>Class ActionTypes defines enumeration values for the J2EE DeploymentStatus actions.</td>
</tr>
<tr>
<td><strong>CommandType</strong></td>
<td>Class CommandTypes defines enumeration values for the DeploymentStatus object.</td>
</tr>
<tr>
<td><strong>DConfigBeanVersionType</strong></td>
<td>Class DConfigBeanVersionTypes defines enumeration values for the J2EE Platform version number.</td>
</tr>
<tr>
<td><strong>ModuleType</strong></td>
<td>Class ModuleTypes defines enumeration values for the J2EE module types.</td>
</tr>
<tr>
<td><strong>StateType</strong></td>
<td>Class StateTypes defines enumeration values for the DeploymentStatus object.</td>
</tr>
</tbody>
</table>
Package `javax.enterprise.deploy.shared`

Description

Provides shared objects for Tool Vendor and Product Vendor implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.shared

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.enterprise.deploy.shared.**ActionType**
  - javax.enterprise.deploy.shared.**CommandType**
  - javax.enterprise.deploy.shared.**DConfigBeanVersionType**
  - javax.enterprise.deploy.shared.**ModuleType**
  - javax.enterprise.deploy.shared.**StateType**
Package javax.activation

The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data.

See:  Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CommandObject</strong></td>
<td>JavaBeans components that are Activation Framework aware implement this interface to find out which command verb they're being asked to perform, and to obtain the DataHandler representing the data they should operate on.</td>
</tr>
<tr>
<td><strong>DataContentHandler</strong></td>
<td>The DataContentHandler interface is implemented by objects that can be used to extend the capabilities of the DataHandler's implementation of the Transferable interface.</td>
</tr>
<tr>
<td><strong>DataContentHandlerFactory</strong></td>
<td>This interface defines a factory for DataContentHandlers.</td>
</tr>
<tr>
<td><strong>DataSource</strong></td>
<td>The DataSource interface provides the JavaBeans Activation Framework with an abstraction of an arbitrary collection of data.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActivationDataFlavor</strong></td>
<td>The ActivationDataFlavor class is a special subclass of java.awt.datatransfer.DataFlavor.</td>
</tr>
<tr>
<td><strong>CommandInfo</strong></td>
<td>The CommandInfo class is used by CommandMap implementations to describe the results of command requests.</td>
</tr>
<tr>
<td>Class/Map</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>CommandMap</td>
<td>The CommandMap class provides an interface to a registry of command objects available in the system.</td>
</tr>
<tr>
<td>DataHandler</td>
<td>The DataHandler class provides a consistent interface to data available in many different sources and formats.</td>
</tr>
<tr>
<td>FileDataSource</td>
<td>The FileDataSource class implements a simple DataSource object that encapsulates a file.</td>
</tr>
<tr>
<td>FileTypeMap</td>
<td>The FileTypeMap is an abstract class that provides a data typing interface for files.</td>
</tr>
<tr>
<td>MailcapCommandMap</td>
<td>MailcapCommandMap extends the CommandMap abstract class.</td>
</tr>
<tr>
<td>Mime Type</td>
<td>A Multipurpose Internet Mail Extension (MIME) type, as defined in RFC 2045 and 2046.</td>
</tr>
<tr>
<td>Mime Type Parameter List</td>
<td>A parameter list of a Mime Type as defined in RFC 2045 and 2046.</td>
</tr>
<tr>
<td>MimetypesFileTypeMap</td>
<td>This class extends FileTypeMap and provides data typing of files via their file extension.</td>
</tr>
<tr>
<td>URLDataSource</td>
<td>The URLDataSource class provides an object that wraps a URL object in a DataSource interface.</td>
</tr>
</tbody>
</table>

**Exception Summary**

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mime Type ParseException</td>
<td>A class to encapsulate Mime Type parsing related exceptions.</td>
</tr>
<tr>
<td>Unsupported Data Type Exception</td>
<td>Signals that the requested operation does not support the requested data type.</td>
</tr>
</tbody>
</table>
Package javax.activation Description

The JavaBeans(TM) Activation Framework is used by the JavaMail(TM) API to manage MIME data.
Hierarchy For Package javax.activation

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.activation.**CommandInfo**
  - javax.activation.**CommandMap**
    - javax.activation.**MailcapCommandMap**
  - java.awt.datatransfer.**DataFlavor** (implements java.lang.**Cloneable**, java.io.**Externalizable**)
    - javax.activation.**ActivationDataFlavor**
  - javax.activation.**DataHandler** (implements java.awt.datatransfer.**Transferable**)
  - javax.activation.**FileDataSource** (implements javax.activation.**DataSource**)
  - javax.activation.**FileTypeMap**
    - javax.activation.**MimetypesFileTypeMap**
  - javax.activation.**MimeType** (implements java.io.**Externalizable**)
  - javax.activation.**MimeTypeParameterList**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - java.io.**IOException**
        - javax.activation.**UnsupportedDataTypeException**
        - javax.activation.**MimeTypeParseException**
    - javax.activation.**URLDataSource** (implements javax.activation.**DataSource**)
## Interface Hierarchy

- javax.activation.CommandObject
- javax.activation.DataContentHandler
- javax.activation.DataContentHandlerFactory
- javax.activation.DataSource

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
The `javax.resource.spi` package contains APIs for the system contracts defined in the J2EE Connector Architecture specification.

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ActivationSpec</strong></td>
<td>This interface serves as a marker.</td>
</tr>
<tr>
<td><strong>BootstrapContext</strong></td>
<td>This provides a mechanism to pass a bootstrap context to a resource adapter instance when it is bootstrapped.</td>
</tr>
<tr>
<td><strong>ConnectionEventListener</strong></td>
<td>The <code>ConnectionEventListener</code> interface provides an event callback mechanism to enable an application server to receive notifications from a <code>ManagedConnection</code> instance.</td>
</tr>
<tr>
<td><strong>ConnectionManager</strong></td>
<td>ConnectionManager interface provides a hook for the resource adapter to pass a connection request to the application server.</td>
</tr>
<tr>
<td><strong>ConnectionRequestInfo</strong></td>
<td>The <code>ConnectionRequestInfo</code> interface enables a resource adapter to pass its own request specific data structure across the connection request flow.</td>
</tr>
<tr>
<td><strong>DissociatableManagedConnection</strong></td>
<td>This is a mix-in interface that may be optionally implemented by a <code>ManagedConnection</code></td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>LazyAssociatableConnectionManager</strong></td>
<td>This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation.</td>
</tr>
<tr>
<td><strong>LazyEnlistableConnectionManager</strong></td>
<td>This is a mix-in interface that may be optionally implemented by a ConnectionManager implementation.</td>
</tr>
<tr>
<td><strong>LazyEnlistableManagedConnection</strong></td>
<td>This is a mix-in interface that may be optionally implemented by a ManagedConnection implementation.</td>
</tr>
<tr>
<td><strong>LocalTransaction</strong></td>
<td>LocalTransaction interface provides support for transactions that are managed internal to an EIS resource manager, and do not require an external transaction manager.</td>
</tr>
<tr>
<td><strong>ManagedConnection</strong></td>
<td>ManagedConnection instance represents a physical connection to the underlying EIS.</td>
</tr>
<tr>
<td><strong>ManagedConnectionFactory</strong></td>
<td>ManagedConnectionFactory instance is a factory of both ManagedConnection and EIS-specific connection factory instances.</td>
</tr>
<tr>
<td><strong>ManagedConnectionMetaData</strong></td>
<td>The ManagedConnectionMetaData interface provides information about the underlying EIS instance associated with a</td>
</tr>
</tbody>
</table>
### ManagedConnection instance.

**ResourceAdapter**

This represents a resource adapter instance and contains operations for lifecycle management and message endpoint setup.

**ResourceAdapterAssociation**

This interface specifies the methods to associate a ResourceAdapter object with other objects that implement this interface like ManagedConnectionFactory and ActivationSpec.

**ValidatingManagedConnectionFactory**

This interface is implemented by a ManagedConnectionFactory instance that supports the ability to validate ManagedConnection objects.

**XATerminator**

The XATerminator interface is used for transaction completion and crash recovery flows.

### Class Summary

**ConnectionEvent**

The ConnectionEvent class provides information about the source of a connection related event. A ConnectionEvent instance contains the following information: Type of the connection event ManagedConnection instance that generated the connection event.

### Exception Summary
<table>
<thead>
<tr>
<th>Exception Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ApplicationServerInternalException</td>
<td>An ApplicationServerInternalException is thrown by an application server to indicate error conditions specific to an application server.</td>
</tr>
<tr>
<td>CommException</td>
<td>This indicates errors related to failed or interrupted communication with an EIS instance.</td>
</tr>
<tr>
<td>EISSystemException</td>
<td>An EISSystemException is used to indicate any EIS specific system-level error conditions.</td>
</tr>
<tr>
<td>IllegalStateException</td>
<td>An IllegalStateException is thrown from a method if the callee (resource adapter or application server for system contracts) is in an illegal or inappropriate state for the method invocation.</td>
</tr>
<tr>
<td>InvalidPropertyException</td>
<td>This exception is thrown to indicate invalid configuration property settings.</td>
</tr>
<tr>
<td>LocalTransactionException</td>
<td>A LocalTransactionException represents various error conditions related to the local transaction management contract.</td>
</tr>
<tr>
<td>ResourceAdapterInternalException</td>
<td>A ResourceAdapterInternalException indicates any system-level error conditions related to a resource adapter.</td>
</tr>
<tr>
<td>ResourceAllocationException</td>
<td>A ResourceAllocationException can be thrown by an application server or resource adapter to indicate any failure to allocate system resources (example: threads, physical connections).</td>
</tr>
<tr>
<td>SecurityException</td>
<td>A SecurityException indicates error conditions related to the security contract between an application</td>
</tr>
</tbody>
</table>
SharingViolationException

This is thrown to indicate a connection sharing violation.

UnavailableException

This is thrown to indicate that a service is unavailable.
The `javax.resource.spi` package contains APIs for the system contracts defined in the J2EE Connector Architecture specification.
Hierarchy For Package javax.resource.spi

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.resource.spi.**ConnectionEvent**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.resource.**ResourceException**
        - javax.resource.spi.**ApplicationServerInternalException**
        - javax.resource.spi.**CommException**
        - javax.resource.spi.**EISSystemException**
        - javax.resource.spi.**IllegalStateException**
        - javax.resource.spi.**InvalidPropertyException**
        - javax.resource.spi.**LocalTransactionException**
        - javax.resource.spi.**ResourceAdapterInternalException**
        - javax.resource.spi.**ResourceAllocationException**
        - javax.resource.spi.**SecurityException**
        - javax.resource.spi.**SharingViolationException**
        - javax.resource.spi.**UnavailableException**
Interface Hierarchy

- javax.resource.spi.BootstrapContext
- javax.resource.spi.ConnectionRequestInfo
- javax.resource.spi.DissociatableManagedConnection
- java.util.EventListener
  - javax.resource.spi.ConnectionEventListener
- javax.resource.spi.LazyAssociatableConnectionManager
- javax.resource.spi.LazyEnlistableConnectionManager
- javax.resource.spi.LazyEnlistableManagedConnection
- javax.resource.spi.LocalTransaction
- javax.resource.spi.ManagedConnection
- javax.resource.spi.ManagedConnectionFactory
- javax.resource.spi.ValidatingManagedConnectionFactory
- javax.resource.spi.XATerminator

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.mail`

The JavaMail™ API provides classes that model a mail system.

See:  

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MessageAware</strong></td>
<td>An interface optionally implemented by DataSources to supply information to a DataContentHandler about the message context in which the data content object is operating.</td>
</tr>
<tr>
<td><strong>MultipartDataSource</strong></td>
<td>MultipartDataSource is a DataSource that contains body parts.</td>
</tr>
<tr>
<td><strong>Part</strong></td>
<td>The Part interface is the common base interface for Messages and BodyParts.</td>
</tr>
<tr>
<td><strong>QuotaAwareStore</strong></td>
<td>An interface implemented by Stores that support quotas.</td>
</tr>
<tr>
<td><strong>UIDFolder</strong></td>
<td>The UIDFolder interface is implemented by Folders that can support the &quot;disconnected&quot; mode of operation, by providing unique-ids for messages in the folder.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Address</strong></td>
<td>This abstract class models the addresses in a message.</td>
</tr>
<tr>
<td><strong>Authenticator</strong></td>
<td>The class Authenticator represents an object that knows how to obtain authentication for a network connection.</td>
</tr>
<tr>
<td><strong>BodyPart</strong></td>
<td>This class models a Part that is contained within a Multipart.</td>
</tr>
<tr>
<td></td>
<td>Clients use a FetchProfile to list the Message attributes that it wishes to</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FetchProfile</td>
<td>prefetch from the server for a range of messages.</td>
</tr>
<tr>
<td>FetchProfile.Item</td>
<td>This inner class is the base class of all items that can be requested in a FetchProfile.</td>
</tr>
<tr>
<td>Flags</td>
<td>The Flags class represents the set of flags on a Message.</td>
</tr>
<tr>
<td>Flags.Flag</td>
<td>This inner class represents an individual system flag.</td>
</tr>
<tr>
<td>Folder</td>
<td>Folder is an abstract class that represents a folder for mail messages.</td>
</tr>
<tr>
<td>Header</td>
<td>The Header class stores a name/value pair to represent headers.</td>
</tr>
<tr>
<td>Message</td>
<td>This class models an email message.</td>
</tr>
<tr>
<td>Message.RecipientType</td>
<td>This inner class defines the types of recipients allowed by the Message class.</td>
</tr>
<tr>
<td>MessageContext</td>
<td>The context in which a piece of Message content is contained.</td>
</tr>
<tr>
<td>Multipart</td>
<td>Multipart is a container that holds multiple body parts.</td>
</tr>
<tr>
<td>PasswordAuthentication</td>
<td>The class PasswordAuthentication is a data holder that is used by Authenticator.</td>
</tr>
<tr>
<td>Provider</td>
<td>The Provider is a class that describes a protocol implementation.</td>
</tr>
<tr>
<td>Provider.Type</td>
<td>This inner class defines the Provider type.</td>
</tr>
<tr>
<td>Quota</td>
<td>This class represents a set of quotas for a given quota root.</td>
</tr>
<tr>
<td>Quota.Resource</td>
<td>An individual resource in a quota root.</td>
</tr>
<tr>
<td>Service</td>
<td>An abstract class that contains the functionality common to messaging services, such as stores and transports.</td>
</tr>
<tr>
<td>Session</td>
<td>The Session class represents a mail session and is not subclassed.</td>
</tr>
<tr>
<td><strong>Store</strong></td>
<td>An abstract class that models a message store and its access protocol, for storing and retrieving messages.</td>
</tr>
<tr>
<td>------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Transport</strong></td>
<td>An abstract class that models a message transport.</td>
</tr>
<tr>
<td><strong>UIDFolder.FetchProfileItem</strong></td>
<td>A fetch profile item for fetching UIDs.</td>
</tr>
<tr>
<td><strong>URLName</strong></td>
<td>The name of a URL.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th><strong>AuthenticationFailedException</strong></th>
<th>This exception is thrown when the connect method on a Store or Transport object fails due to an authentication failure (e.g., bad user name or password).</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>FolderClosedException</strong></td>
<td>This exception is thrown when a method is invoked on a Messaging object and the Folder that owns that object has died due to some reason.</td>
</tr>
<tr>
<td><strong>FolderNotFoundException</strong></td>
<td>This exception is thrown by Folder methods, when those methods are invoked on a non-existent folder.</td>
</tr>
<tr>
<td><strong>IllegalWriteException</strong></td>
<td>The exception thrown when a write is attempted on a read-only attribute of any Messaging object.</td>
</tr>
<tr>
<td><strong>MessageRemovedException</strong></td>
<td>The exception thrown when an invalid method is invoked on an expunged Message.</td>
</tr>
<tr>
<td><strong>MessagingException</strong></td>
<td>The base class for all exceptions thrown by the Messaging classes</td>
</tr>
<tr>
<td><strong>MethodNotSupportedException</strong></td>
<td>The exception thrown when a method is not supported by the implementation</td>
</tr>
<tr>
<td><strong>NoSuchProviderException</strong></td>
<td>This exception is thrown when Session attempts to instantiate a</td>
</tr>
<tr>
<td>Exception</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Provider that doesn't exist.</td>
<td></td>
</tr>
<tr>
<td><strong>ReadOnlyFolderException</strong></td>
<td>This exception is thrown when an attempt is made to open a folder read-write access when the folder is marked read-only.</td>
</tr>
<tr>
<td><strong>SendFailedException</strong></td>
<td>This exception is thrown when the message cannot be sent.</td>
</tr>
<tr>
<td><strong>StoreClosedException</strong></td>
<td>This exception is thrown when a method is invoked on a Messaging object and the Store that owns that object has died due to some reason.</td>
</tr>
</tbody>
</table>
The JavaMail™ API provides classes that model a mail system. The javax.mail package defines classes that are common to all mail systems. The javax.mail.internet package defines classes that are specific to mail systems based on internet standards such as MIME, SMTP, POP3, and IMAP. The JavaMail API includes the javax.mail package and subpackages.

For an overview of the JavaMail API, read the JavaMail specification [included in the download bundle](https://developers.jre.com/jre/downloads) or [available on the JavaMail web site](https://mail.on.java.com).

The code to send a plain text message can be as simple as the following:

```java
Properties props = new Properties();
props.put("mail.smtp.host", "my-mail-server");
props.put("mail.from", "me@example.com");
Session session = Session.getInstance(props, null);
try {
    MimeMessage msg = new MimeMessage(session);
    msg.setFrom();
    msg.setRecipients(Message.RecipientType.TO, "you@example.com");
    msg.setSubject("JavaMail hello world example");
    msg.setSentDate(new Date());
    msg.setText("Hello, world!\n");
    Transport.send(msg);
} catch (MessagingException mex) {
    System.out.println("send failed, exception: " + mex);
}
```

The JavaMail download bundle contains many more complete examples in the "demo" directory.

Don't forget to see the JavaMail API FAQ for answers to the most common questions. The JavaMail web site contains many additional resources.

The JavaMail API supports the following standard properties, which may be set in the Session object, or in the Properties object used to create the Session object. The properties are always set as strings; the Type column
describes how the string is interpreted. For example, use

```java
props.put("mail.debug", "true");
```

to set the `mail.debug` property, which is of type boolean.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.debug</td>
<td>boolean</td>
<td>The initial debug mode. Default is false.</td>
</tr>
<tr>
<td>mail.from</td>
<td>String</td>
<td>The return email address of the current user, used by the <code>InternetAddress</code> method <code>getLocalAddress</code>.</td>
</tr>
<tr>
<td>mail.mime.address.strict</td>
<td>boolean</td>
<td>The MimeMessage class uses the <code>InternetAddress</code> method <code>parseHeader</code> to parse headers in messages. This property controls the strict flag passed to the <code>parseHeader</code> method. The default is true.</td>
</tr>
<tr>
<td>mail.host</td>
<td>String</td>
<td>The default host name of the mail server for both Stores and Transports. Used if the <code>mail.protocol.host</code> property isn't set.</td>
</tr>
<tr>
<td>mail.store.protocol</td>
<td>String</td>
<td>Specifies the default message access protocol. The <code>Session</code> method <code>getStore()</code> returns a Store object that implements this protocol. By default the first Store provider in the configuration files is returned.</td>
</tr>
<tr>
<td>mail.transport.protocol</td>
<td>String</td>
<td>Specifies the default message access protocol. The <code>Session</code> method <code>getTransport()</code> returns a Transport object that implements this protocol. By default the first Transport provider in the configuration files is returned.</td>
</tr>
<tr>
<td>mail.user</td>
<td>String</td>
<td>The default user name to use when connecting to the mail server. Used if</td>
</tr>
</tbody>
</table>
The mail.protocol.user property isn't set.

<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.protocol.class</td>
<td>String</td>
<td>Specifies the fully qualified class name of the provider for the specified protocol. Used in cases where more than one provider for a given protocol exists; this property can be used to specify which provider to use by default. The provider must still be listed in a configuration file.</td>
</tr>
<tr>
<td>mail.protocol.host</td>
<td>String</td>
<td>The host name of the mail server for the specified protocol. Overrides the mail.host property.</td>
</tr>
<tr>
<td>mail.protocol.port</td>
<td>int</td>
<td>The port number of the mail server for the specified protocol. If not specified the protocol's default port number is used.</td>
</tr>
<tr>
<td>mail.protocol.user</td>
<td>String</td>
<td>The user name to use when connecting to mail servers using the specified protocol. Overrides the mail.user property.</td>
</tr>
</tbody>
</table>

The JavaMail API also supports several System properties; see the javax.mail.internet package documentation for details.

The JavaMail reference implementation from Sun includes protocol providers in subpackages of com.sun.mail. Note that the APIs to these protocol providers are not part of the standard JavaMail API. Portable programs will not use these APIs.

Nonportable programs may use the APIs of the Sun protocol providers by (for example) casting a returned Folder object to a com.sun.mail.imap.IMAPFolder object. Similarly for Store and Message objects returned from the standard JavaMail APIs.

The Sun protocol providers also support properties that are specific to those providers. The package documentation for the IMAP, POP3, and SMTP packages provide details.
Hierarchy For Package javax.mail

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.mail.**Address** (implements java.io.**Serializable**)
  - javax.mail.**Authenticator**
  - javax.mail.**BodyPart** (implements javax.mail.**Part**)
  - javax.mail.**FetchProfile**
  - javax.mail.**FetchProfile.Item**
    - javax.mail.**UIDFolder.FetchProfileItem**
  - javax.mail.**Flags** (implements java.lang.**Cloneable**, java.io.**Serializable**)
  - javax.mail.**Flags.Flag**
  - javax.mail.**Folder**
  - javax.mail.**Header**
  - javax.mail.**Message** (implements javax.mail.**Part**)
  - javax.mail.**Message.RecipientType** (implements java.io.**Serializable**)
  - javax.mail.**MessageContext**
  - javax.mail.**Multipart**
  - javax.mail.**PasswordAuthentication**
  - javax.mail.**Provider**
  - javax.mail.**Provider.Type**
  - javax.mail.**Quota**
  - javax.mail.**Quota.Resource**
  - javax.mail.**Service**
    - javax.mail.**Store**
    - javax.mail.**Transport**
  - javax.mail.**Session**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.mail.**MessagingException**
        - javax.mail.**AuthenticationFailedException**
        - javax.mail.**FolderClosedException**
        - javax.mail.**FolderNotFoundException**
        - javax.mail.**IllegalWriteException**
        - javax.mail.**MessageRemovedException**
        - javax.mail.**MethodNotSupportedException**
- javax.mail.NoSuchProviderException
- javax.mail.ReadOnlyFolderException
- javax.mail.SendFailedException
- javax.mail.StoreClosedException
- javax.mail.URLName
### Interface Hierarchy

- javax.activation.**DataSource**
  - javax.mail.**MultipartDataSource**
- javax.mail.**MessageAware**
- javax.mail.**Part**
- javax.mail.**QuotaAwareStore**
- javax.mail.**UIDFolder**

### Table of Contents

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Class</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

[Submit a bug or feature](#)

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.mail.internet

Classes specific to Internet mail systems.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MimePart</td>
<td>The MimePart interface models an Entity as defined by MIME (RFC2045, Section 2.4).</td>
</tr>
<tr>
<td>SharedInputStream</td>
<td>An InputStream that is backed by data that can be shared by multiple readers may implement this interface.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ContentDisposition</td>
<td>This class represents a MIME ContentDisposition value.</td>
</tr>
<tr>
<td>ContentType</td>
<td>This class represents a MIME ContentType value.</td>
</tr>
<tr>
<td>HeaderTokenizer</td>
<td>This class tokenizes RFC822 and MIME headers into the basic symbols specified by RFC822 and MIME.</td>
</tr>
<tr>
<td>HeaderTokenizer.Token</td>
<td>The Token class represents tokens returned by the HeaderTokenizer.</td>
</tr>
<tr>
<td>InternetAddress</td>
<td>This class represents an Internet email address using the syntax of RFC822.</td>
</tr>
<tr>
<td>InternetHeaders</td>
<td>InternetHeaders is a utility class that manages RFC822 style headers.</td>
</tr>
<tr>
<td>InternetHeaders.InternetHeader</td>
<td>An individual internet header.</td>
</tr>
<tr>
<td>MailDateFormat</td>
<td>Formats and parses date specification based on the draft-ietf-drums-msg-fmt-08 dated January 26,</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>MimeBodyPart</td>
<td>This class represents a MIME body part.</td>
</tr>
<tr>
<td>MimeMessage</td>
<td>This class represents a MIME style email message.</td>
</tr>
<tr>
<td>MimeMessage.RecipientType</td>
<td>This inner class extends the javax.mail.Message.RecipientType class to add additional RecipientTypes.</td>
</tr>
<tr>
<td>MimeMultipart</td>
<td>The MimeMultipart class is an implementation of the abstract Multipart class that uses MIME conventions for the multipart data.</td>
</tr>
<tr>
<td>MimePartDataSource</td>
<td>A utility class that implements a DataSource out of a MimePart.</td>
</tr>
<tr>
<td>MimeUtility</td>
<td>This is a utility class that provides various MIME related functionality.</td>
</tr>
<tr>
<td>NewsAddress</td>
<td>This class models an RFC1036 newsgroup address.</td>
</tr>
<tr>
<td>ParameterList</td>
<td>This class holds MIME parameters (attribute-value pairs).</td>
</tr>
<tr>
<td>PreencodedMimeBodyPart</td>
<td>A MimeBodyPart that handles data that has already been encoded.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddressException</td>
<td>The exception thrown when a wrongly formatted address is encountered.</td>
</tr>
<tr>
<td>ParseException</td>
<td>The exception thrown due to an error in parsing RFC822 or MIME headers.</td>
</tr>
</tbody>
</table>
Classes specific to Internet mail systems. This package supports features that are specific to Internet mail systems based on the MIME standard (RFC 2045, RFC 2046, and RFC 2047). The IMAP, SMTP, and POP3 protocols use MimeMessages.

The JavaMail API specification requires support for the following properties, which must be set in the system properties. The properties are always set as strings; the Type column describes how the string is interpreted. For example, use (in J2SE 1.2 and newer)

```
System.setProperty("mail.mime.address.strict", "false");
```
to set the mail.mime.address.strict property, which is of type boolean.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.mime.address.strict</td>
<td>boolean</td>
<td>The mail.mime.address.strict controls the parsing of address headers. The default, strict parsing of address headers is done. If this property is set to &quot;false&quot;, parsing is not done and many addresses that sometimes occur in real messages are allowed. See the InternetAddress details.</td>
</tr>
<tr>
<td>mail.mime.charset</td>
<td>String</td>
<td>The mail.mime.charset System property can be used to specify the default MIME charset to use for encoded words and text parts that do not otherwise specify a charset. The default MIME charset is derived from the default Java charset, as specified in file.encoding System property. Applications will have no need to specify the default MIME charset. In cases where the default MIME charset to be used for mail messages is different than the default for files stored on the system, this property should be set.</td>
</tr>
</tbody>
</table>
### mail.mime.decodetext.strict

**Type:** boolean  
**Default:** true

The `mail.mime.decodetext.strict` property controls decoding of MIME encoded words. The MIME spec requires that encoded words start at the beginning of a whitespace-separated word. Some mailers may incorrectly include encoded words in the middle of a word. If the `mail.mime.decodetext.strict` property is set to "false", an attempt will be made to decode these illegal encoded words. The default is true.

### mail.mime.encodeeol.strict

**Type:** boolean  
**Default:** false

The `mail.mime.encodeeol.strict` property controls the choice of Content-Transfer-Encoding for MIME parts that are not of type "text". Often such parts will contain textual data for which an encoding that allows normal end-of-line conventions is appropriate, but such a part will appear to contain only textual data, but will require an encoding that preserves CR and LF characters without change. If the `mail.mime.encodeeol.strict` System property is set to "true", the appropriate encoding will be used when necessary. The default is false.

### mail.mime.decodefilename

**Type:** boolean  
**Default:** false

If set to "true", the `getFileName` method of the MimeUtility method decodes any non-ASCII characters in the filename that this decoding violates the MIME specification, but is useful for interoperating with some mail clients that use this convention. The default is false.

### mail.mime.encodefilename

**Type:** boolean  
**Default:** false

If set to "true", the `setFileName` method of the MimeUtility method encodes any non-ASCII characters in the filename that this encoding violates the MIME specification, but is useful for interoperating with some mail clients that use this convention. The default is false.
mail.mime.decodeparameters  boolean  If set to "true", non-ASCII parameters in a ParameterList, e.g., in a Content-Type header, will be encoded as specified by default is false.

mail.mime.encodeparameters  boolean  If set to "true", non-ASCII parameters in a ParameterList, e.g., in a Content-Type header, will be decoded as specified by default is false.

mail.mime.multipart.ignoremissingendboundary  boolean  Normally, when parsing a multipart message, a message that is missing the final end boundary line is not considered an error. The data simply ends at the end boundary line. Note that messages of this form violate the MIME specification. If the property mail.mime.multipart.ignoremissingendboundary is set to false, such messages are considered an error and a MessagingException will be thrown when parsing such a message.

mail.mime.multipart.ignoremissingboundaryparameter  boolean  If the Content-Type header for a multipart content does not have a boundary parameter, the multipart parsing code will find the first line in the content that looks like a boundary line and extract the boundary parameter from the line. If this property is set to true, a MessagingException will be thrown if the Content-Type header doesn't contain a boundary parameter. The default is false.

The following properties are supported by Sun's implementation of JavaMail, but are not currently a required part of the specification. As above, these must be set as System properties. The names, types, defaults, and semantics of these properties may change in future releases.

<table>
<thead>
<tr>
<th>Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.mime.decodetextoptions</td>
<td></td>
<td>If set to &quot;true&quot;, the BASE64 decoder will ignore errors in Base64 encoding</td>
</tr>
<tr>
<td>mail.mime.encoderoptions</td>
<td></td>
<td>If set to &quot;true&quot;, the encoder will not encode non-ASCII characters</td>
</tr>
<tr>
<td>mail.mime.multipart.ignoremissingendboundary</td>
<td></td>
<td>If set to false, multipart messages that are missing the final end boundary line are considered an error</td>
</tr>
<tr>
<td>mail.mime.multipart.ignoremissingboundaryparameter</td>
<td></td>
<td>If set to false, multipart messages that lack a boundary parameter are considered an error</td>
</tr>
<tr>
<td>mail.mime.multipart.ignoremissingtextoptions</td>
<td></td>
<td>If set to false, multipart messages that are missing the final text boundary line are considered an error</td>
</tr>
</tbody>
</table>

The following properties are supported by Sun's implementation of JavaMail, but are not currently a required part of the specification. As above, these must be set as System properties. The names, types, defaults, and semantics of these properties may change in future releases.
<table>
<thead>
<tr>
<th>Property</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>mail.mime.base64.ignoreerrors</td>
<td>boolean</td>
<td>The encoded data, returning EOF. This may be useful when dealing with improperly encoded messages that contain extraneous data at the end of the encoded stream. Note however that errors anywhere in the stream will cause the decoder to stop decoding so this should be used with extreme caution. The default is false.</td>
</tr>
<tr>
<td>mail.mime.foldtext</td>
<td>boolean</td>
<td>If set to &quot;true&quot;, header fields containing just text such as the Subject and Content-Description header fields, and long parameter values in structured headers such as Content-Type will be folded (broken into 76 character lines) when set and unfolded when read. The default is true.</td>
</tr>
<tr>
<td>mail.mime.setcontenttypefilename</td>
<td>boolean</td>
<td>If set to &quot;true&quot;, the setFileName method will also set the name parameter on the Content-Type header to the specified filename. This supports interoperability with some old mail clients. The default is true.</td>
</tr>
<tr>
<td>mail.mime.setdefaulttextcharset</td>
<td>boolean</td>
<td>When updating the headers of a message, a body part with a text content type but no charset parameter will have a charset parameter.</td>
</tr>
<tr>
<td>Property</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>mail.mime.applefilenames</td>
<td>boolean</td>
<td>Apple Mail incorrectly encodes filenames that contain spaces, forgetting to quote the parameter value. If this property is set to &quot;true&quot;, JavaMail will try to detect this situation when parsing parameters and work around it. The default is false.</td>
</tr>
<tr>
<td>mail.alternates</td>
<td>String</td>
<td>A string containing other email addresses that the current user is known by. The MimeMessage reply method will eliminate any of these addresses from the recipient list in the message it constructs, to avoid sending the reply back to the sender.</td>
</tr>
<tr>
<td>mail.replyallcc</td>
<td>boolean</td>
<td>If set to &quot;true&quot;, the MimeMessage reply method will put all recipients except the original sender in the cc list of the newly constructed message. Normally, recipients in the To header of the original message will also appear in the To list of the newly constructed message.</td>
</tr>
</tbody>
</table>
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.mail.internet

Package Hierarchies:

All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.mail.Address` (implements `java.io.Serializable`)
    - `javax.mail.internet.InternetAddress` (implements `java.lang.Cloneable`)
  - `javax.mail.internet.NewsAddress`
  - `javax.mail.BodyPart` (implements `javax.mail.Part`)
    - `javax.mail.internet.MimeBodyPart` (implements `javax.mail.internet.MimePart`)
      - `javax.mail.internet.PreencodedMimeBodyPart`
  - `javax.mail.internet.ContentDisposition`
  - `javax.mail.internet.ContentType`
  - `java.text.Format` (implements `java.lang.Cloneable`, `java.io.Serializable`)
    - `java.text.SimpleDateFormat`
      - `javax.mail.internet.MailDateFormat`
  - `javax.mail.Header`
    - `javax.mail.internet.InternetHeaders.InternetHeader`
  - `javax.mail.internet.HeaderTokenizer`
  - `javax.mail.internet.HeaderTokenizer.Token`
  - `javax.mail.internet.InternetHeaders`
  - `javax.mail.Message` (implements `javax.mail.Part`)
    - `javax.mail.internet.MimeMessage` (implements `javax.mail.internet.MimePart`)
  - `javax.mail.Message.RecipientType` (implements `java.io.Serializable`)
    - `javax.mail.internet.MimeMessage.RecipientType`
  - `javax.mail.internet.MimePartDataSource` (implements `javax.activation.DataSource`, `javax.mail.MessageAware`)
  - `javax.mail.internet.MimeUtility`
  - `javax.mail.Multipart`
    - `javax.mail.internet.MimeMultipart`
  - `javax.mail.internet.ParameterList`
  - `java.lang.Throwable` (implements `java.io.Serializable`)
    - `java.lang.Exception`
○ javax.mail.**MessagingException**
  ○ javax.mail.internet.**ParseException**
  ○ javax.mail.internet.**AddressException**
Interface Hierarchy

- `javax.mail.Part`
  - `javax.mail.internet.MimePart`
  - `javax.mail.internet.SharedInputStream`
## Package javax.mail.search

Message search terms for the JavaMail API.

See:  [Description](#)

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AddressStringTerm</strong></td>
<td>This abstract class implements string comparisons for Message addresses.</td>
</tr>
<tr>
<td><strong>AddressTerm</strong></td>
<td>This class implements Message Address comparisons.</td>
</tr>
<tr>
<td><strong>AndTerm</strong></td>
<td>This class implements the logical AND operator on individual SearchTerms.</td>
</tr>
<tr>
<td><strong>BodyTerm</strong></td>
<td>This class implements searches on a Message Body.</td>
</tr>
<tr>
<td><strong>ComparisonTerm</strong></td>
<td>This class models the comparison operator.</td>
</tr>
<tr>
<td><strong>DateTerm</strong></td>
<td>This class implements comparisons for Dates.</td>
</tr>
<tr>
<td><strong>FlagTerm</strong></td>
<td>This class implements comparisons for Message Flags.</td>
</tr>
<tr>
<td><strong>FromStringTerm</strong></td>
<td>This class implements string comparisons for the From Address header.</td>
</tr>
<tr>
<td><strong>FromTerm</strong></td>
<td>This class implements comparisons for the From Address header.</td>
</tr>
<tr>
<td><strong>HeaderTerm</strong></td>
<td>This class implements comparisons for Message headers.</td>
</tr>
<tr>
<td><strong>IntegerComparisonTerm</strong></td>
<td>This class implements comparisons for integers.</td>
</tr>
<tr>
<td><strong>MessageIDTerm</strong></td>
<td>This term models the RFC822 &quot;MessageId&quot; - a message-id for Internet messages that is supposed to be unique per message.</td>
</tr>
<tr>
<td><strong>MessageNumberTerm</strong></td>
<td>This class implements comparisons for Message numbers.</td>
</tr>
<tr>
<td><strong>NotTerm</strong></td>
<td>This class implements the logical NEGATION</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>NotTerm</td>
<td>This class implements the logical <strong>NOT</strong> operator on individual SearchTerms.</td>
</tr>
<tr>
<td>OrTerm</td>
<td>This class implements the logical <strong>OR</strong> operator on individual SearchTerms.</td>
</tr>
<tr>
<td>ReceivedDateTerm</td>
<td>This class implements comparisons for the Message Received date</td>
</tr>
<tr>
<td>RecipientStringTerm</td>
<td>This class implements string comparisons for the Recipient Address headers.</td>
</tr>
<tr>
<td>RecipientTerm</td>
<td>This class implements comparisons for the Recipient Address headers.</td>
</tr>
<tr>
<td>SearchTerm</td>
<td>Search criteria are expressed as a tree of search-terms, forming a parse-tree for the search expression.</td>
</tr>
<tr>
<td>SentDateTerm</td>
<td>This class implements comparisons for the Message SentDate.</td>
</tr>
<tr>
<td>SizeTerm</td>
<td>This class implements comparisons for Message sizes.</td>
</tr>
<tr>
<td>StringTerm</td>
<td>This class implements the match method for Strings.</td>
</tr>
<tr>
<td>SubjectTerm</td>
<td>This class implements comparisons for the Message Subject header.</td>
</tr>
</tbody>
</table>

**Exception Summary**

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SearchException</td>
<td>The exception thrown when a Search expression could not be handled.</td>
</tr>
</tbody>
</table>
Package javax.mail.search Description

Message search terms for the JavaMail API. This package defines classes that can be used to construct a search expression to search a folder for messages matching the expression; see the search method on javax.mail.Folder. See SearchTerm.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.mail.search

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.mail.search.**SearchTerm** (implements java.io.**Serializable**)  
    - javax.mail.search.**AddressTerm**
    - javax.mail.search.**FromTerm**
    - javax.mail.search.**RecipientTerm**
  - javax.mail.search.**AndTerm**
  - javax.mail.search.**ComparisonTerm**
    - javax.mail.search.**DateTerm**
      - javax.mail.search.**ReceivedDateTerm**
      - javax.mail.search.**SentDateTerm**
    - javax.mail.search.**IntegerComparisonTerm**
    - javax.mail.search.**MessageNumberTerm**
    - javax.mail.search.**SizeTerm**
  - javax.mail.search.**FlagTerm**
  - javax.mail.search.**NotTerm**
  - javax.mail.search.**OrTerm**
  - javax.mail.search.**StringTerm**
    - javax.mail.search.**AddressStringTerm**
      - javax.mail.search.**FromStringTerm**
      - javax.mail.search.**RecipientStringTerm**
    - javax.mail.search.**BodyTerm**
    - javax.mail.search.**HeaderTerm**
    - javax.mail.search.**MessageIDTerm**
    - javax.mail.search.**SubjectTerm**
  - java.lang.**Throwable** (implements java.io.**Serializable**)  
    - java.lang.**Exception**
      - javax.mail.**MessagingException**
        -javax.mail.search.**SearchException**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.faces.application

APIs that are used to link an application's business logic objects to JavaServer Faces, as well as convenient pluggable mechanisms to manage the execution of an application that is based on JavaServer Faces.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Application</strong></td>
</tr>
<tr>
<td><strong>ApplicationFactory</strong></td>
</tr>
<tr>
<td><strong>FacesMessage</strong></td>
</tr>
<tr>
<td><strong>FacesMessage.Severity</strong></td>
</tr>
<tr>
<td><strong>NavigationHandler</strong></td>
</tr>
<tr>
<td><strong>StateManager</strong></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td><strong>StateManagerWrapper</strong></td>
</tr>
<tr>
<td>-------------------------</td>
</tr>
<tr>
<td><strong>ViewHandler</strong></td>
</tr>
<tr>
<td><strong>ViewHandlerWrapper</strong></td>
</tr>
</tbody>
</table>

**Exception Summary**

| **ViewExpiredException** | Implementations must throw this `FacesException` attempting to restore the view `StateManager.restoreView(java.lang.String, java.lang.String)` results in failure on postback. |
APIs that are used to link an application's business logic objects to JavaServer Faces, as well as convenient pluggable mechanisms to manage the execution of an application that is based on JavaServer Faces. The main class in this package is Application.
Hierarchy For Package javax.faces.application

Package Hierarchies:

All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.faces.application.Application`
  - `javax.faces.application.ApplicationFactory`
  - `javax.faces.application.FacesMessage` (implements `java.io.Serializable`)
  - `javax.faces.application.NavigationHandler`
  - `javax.faces.application.StateManager`
    - `javax.faces.application.StateManagerWrapper`
  - `javax.faces.applicationStateManager.SerializedView`
  - `java.lang.Throwable` (implements `java.io.Serializable`)
    - `java.lang.Exception`
      - `javax.faces.FacesException`
        - `javax.faces.application.ViewExpiredException`
      - `javax.faces.application.ViewHandler`
    - `javax.faces.application.ViewHandlerWrapper`
Package javax.interceptor

The javax.interceptor package contains classes and interfaces for use with EJB interceptors.

See:  

### Interface Summary

<table>
<thead>
<tr>
<th>InvocationContext</th>
<th>Context information passed to AroundInvoke and Interceptor-class lifecycle callback methods.</th>
</tr>
</thead>
</table>

### Annotation Types Summary

<table>
<thead>
<tr>
<th>Annotation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AroundInvoke</td>
<td>Used to exclude class-level interceptors for a business method.</td>
</tr>
<tr>
<td>ExcludeClassInterceptors</td>
<td>Used to exclude default interceptors for a bean or a business method.</td>
</tr>
<tr>
<td>ExcludeDefaultInterceptors</td>
<td>Declares an ordered list of interceptors for a class or method.</td>
</tr>
<tr>
<td>Interceptors</td>
<td></td>
</tr>
</tbody>
</table>
The `javax.interceptor` package contains classes and interfaces for use with EJB interceptors.
Hierarchy For Package javax.interceptor

Package Hierarchies:
  All Packages
Interface Hierarchy

- `javax.interceptor.InvocationContext`
Annotation Type Hierarchy

- `javax.interceptor.AroundInvoke` (implements `java.lang.annotation.Annotation`)
- `javax.interceptor.ExcludeClassInterceptors` (implements `java.lang.annotation.Annotation`)
- `javax.interceptor.ExcludeDefaultInterceptors` (implements `java.lang.annotation.Annotation`)
- `javax.interceptor.Interceptors` (implements `java.lang.annotation.Annotation`)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
## Package javax.faces.model

Standard model data beans for JavaServer Faces.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DataModelListener</td>
<td>Represents an event listener that wishes to be notified of DataModelEvents occurring on a particular DataModel instance.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayDataModel</td>
<td>is a convenience implementation of DataModel that wraps an array of Java objects.</td>
</tr>
<tr>
<td>DataModel</td>
<td>is an abstraction around arbitrary data binding technologies that can be used to adapt a variety of data sources for use by JavaServer Faces components that support per-row processing for their child components (such as UIData.</td>
</tr>
<tr>
<td>DataModelEvent</td>
<td>represents an event of interest to registered listeners that occurred on the specified DataModel.</td>
</tr>
<tr>
<td>ListDataModel</td>
<td>is a convenience implementation of DataModel that wraps an List of Java objects.</td>
</tr>
<tr>
<td>ResultDataModel</td>
<td>is a convenience implementation of DataModel that wraps a JSTL Result object, typically representing the results of executing an SQL query via JSTL tags.</td>
</tr>
<tr>
<td>ResultSetDataModel</td>
<td>is a convenience implementation of DataModel that wraps a ResultSet of Java objects.</td>
</tr>
<tr>
<td><strong>ScalarDataModel</strong></td>
<td><strong>ScalarDataModel</strong> is a convenience implementation of DataModel that wraps an individual Java object.</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SelectItem</strong></td>
<td><strong>SelectItem</strong> represents a single item in the list of supported items associated with a UISelectMany or UISelectOne component.</td>
</tr>
<tr>
<td><strong>SelectItemGroup</strong></td>
<td><strong>SelectItemGroup</strong> is a subclass of SelectItem that identifies a set of options that will be made available as a subordinate &quot;submenu&quot; or &quot;options list&quot;, depending upon the requirements of the UISelectMany or UISelectOne renderer that is actually used.</td>
</tr>
</tbody>
</table>
Package javax.faces.model Description

Standard model data beans for JavaServer Faces.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.faces.model

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.model.**DataModel**
    - javax.faces.model.**ArrayDataModel**
    - javax.faces.model.**ListDataModel**
    - javax.faces.model.**ResultDataModel**
    - javax.faces.model.**ResultSetDataModel**
    - javax.faces.model.**ScalarDataModel**
  - java.util.**EventObject** (implements java.io.**Serializable**)  
    - javax.faces.model.**DataModelEvent**
  - javax.faces.model.**SelectItem** (implements java.io.**Serializable**)  
    - javax.faces.model.**SelectItemGroup**
Interface Hierarchy

- java.util.**EventListener**
  - javax.faces.model.**DataModelListener**
Package javax.el

Provides the API for the Unified Expression Language shared by the JSP 2.1 and JSF 1.2 technologies.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ELContextListener</code></td>
<td>The listener interface for receiving notification when an <code>ELContext</code> is created.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ArrayELResolver</code></td>
<td>Defines property resolution behavior on arrays.</td>
</tr>
<tr>
<td><code>BeanELResolver</code></td>
<td>Defines property resolution behavior on objects using the JavaBeans component architecture.</td>
</tr>
<tr>
<td><code>BeanELResolver.BeanProperties</code></td>
<td></td>
</tr>
<tr>
<td><code>BeanELResolver.BeanProperty</code></td>
<td></td>
</tr>
<tr>
<td><code>CompositeELResolver</code></td>
<td>Maintains an ordered composite list of child <code>ELResolver</code> ELResolvers.</td>
</tr>
<tr>
<td><code>ELContext</code></td>
<td>Context information for expression evaluation.</td>
</tr>
<tr>
<td><code>ELContextEvent</code></td>
<td>An event which indicates that an <code>ELContext</code> has been created.</td>
</tr>
<tr>
<td><code>ELResolver</code></td>
<td>Enables customization of variable and property resolution behavior for EL expression evaluation.</td>
</tr>
<tr>
<td><code>Expression</code></td>
<td>Base class for the expression subclasses <code>ValueExpression</code> and <code>MethodExpression</code>, implementing characteristics common to both.</td>
</tr>
<tr>
<td>ExpressionFactory</td>
<td>The interface to a map between EL function names and methods.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>FunctionMapper</td>
<td>Defines property resolution behavior on instances of List.</td>
</tr>
<tr>
<td>ListELResolver</td>
<td>Defines property resolution behavior on instances of Map.</td>
</tr>
<tr>
<td>MapELResolver</td>
<td>Defines property resolution behavior on instances of Map.</td>
</tr>
<tr>
<td>MethodExpression</td>
<td>An Expression that refers to a method on an object.</td>
</tr>
<tr>
<td>MethodInfo</td>
<td>Holds information about a method that a MethodExpression evaluated to.</td>
</tr>
<tr>
<td>ResourceBundleELResolver</td>
<td>Defines property resolution behavior on instances of ResourceBundle.</td>
</tr>
<tr>
<td>ValueExpression</td>
<td>An Expression that can get or set a value.</td>
</tr>
<tr>
<td>VariableMapper</td>
<td>The interface to a map between EL variables and the EL expressions they are associated with.</td>
</tr>
</tbody>
</table>

## Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELException</td>
<td>Represents any of the exception conditions that can arise during expression evaluation.</td>
</tr>
<tr>
<td>MethodNotFoundException</td>
<td>Thrown when a method could not be found while evaluating a MethodExpression.</td>
</tr>
<tr>
<td>PropertyNotFoundException</td>
<td>Thrown when a property could not be found while evaluating a ValueExpression or MethodExpression.</td>
</tr>
<tr>
<td>PropertyNotWritableException</td>
<td>Thrown when a property could not be written to while setting the value on a ValueExpression.</td>
</tr>
</tbody>
</table>
Package javax.el Description

Provides the API for the **Unified Expression Language** shared by the JSP 2.1 and JSF 1.2 technologies.

The Expression Language (EL) is a simple language designed to satisfy the specific needs of web application developers. It is currently defined in its own specification document within the JavaServer Pages (tm) (JSP) 2.1 specification, but does not have any dependencies on any portion of the JSP 2.1 specification. It is intended for general use outside of the JSP and JSF specifications as well.

This package contains the classes and interfaces that describe and define the programmatic access to the Expression Language engine. The API is logically partitioned as follows:

- EL Context
- Expression Objects
- Creation of Expressions
- Resolution of Model Objects and their Properties
- EL Functions
- EL Variables

**EL Context**

An important goal of the EL is to ensure it can be used in a variety of environments. It must therefore provide enough flexibility to adapt to the specific requirements of the environment where it is being used.

Class **ELContext** is what links the EL with the specific environment where it is being used. It provides the mechanism through which all relevant context for creating or evaluating an expression is specified.

Creation of **ELContext** objects is controlled through the underlying technology. For example, in JSP, the `JspContext.getELContext()` factory method is used.

Some technologies provide the ability to add an **ELContextListener** so
that applications and frameworks can ensure their own context objects are attached to any newly created ELContext.

Expression Objects

At the core of the Expression Language is the notion of an expression that gets parsed according to the grammar defined by the Expression Language.

There are two types of expressions defined by the EL: value expressions and method expressions. A ValueExpression such as "${customer.name}" can be used either as an rvalue (return the value associated with property name of the model object customer) or as an lvalue (set the value of the property name of the model object customer).

A MethodExpression such as "${handler.process}" makes it possible to invoke a method (process) on a specific model object (handler).

All expression classes extend the base class Expression, making them serializable and forcing them to implement equals() and hashCode(). Moreover, each method on these expression classes that actually evaluates an expression receives a parameter of class ELContext, which provides the context required to evaluate the expression.

Creation of Expressions

An expression is created through the ExpressionFactory class. The factory provides two creation methods; one for each type of expression supported by the EL.

To create an expression, one must provide an ELContext, a string representing the expression, and the expected type (ValueExpression) or signature (MethodExpression). The ELContext provides the context necessary to parse an expression. Specifically, if the expression uses an EL function (for example ${fn:toUpperCase(customer.name)}) or an EL variable, then FunctionMapper and VariableMapper objects must be available within the ELContext so that EL functions and EL variables are properly mapped.
Resolution of Model Objects and their Properties

Through the `ELResolver` base class, the EL features a pluggable mechanism to resolve model object references as well as properties of these objects.

The EL API provides implementations of `ELResolver` supporting property resolution for common data types which include arrays (`ArrayELResolver`), JavaBeans (`BeanELResolver`), Lists (`ListELResolver`), Maps (`MapELResolver`), and Resource Bundles (`ResourceBundleELResolver`).

Tools can easily obtain more information about resolvable model objects and their resolvable properties by calling method `getFeatureDescriptors` on the `ELResolver`. This method exposes objects of type `java.beans.FeatureDescriptor`, providing all information of interest on top-level model objects as well as their properties.

EL Functions

If an EL expression uses a function (for example `${fn:toUpperCase(customer.name)}`), then a `FunctionMapper` object must also be specified within the ELContext. The `FunctionMapper` is responsible to map `${prefix:name()}` style functions to static methods that can execute the specified functions.

EL Variables

Just like `FunctionMapper` provides a flexible mechanism to add functions to the EL, `VariableMapper` provides a flexible mechanism to support the notion of EL variables.

An EL variable does not directly refer to a model object that can then be resolved by an `ELResolver`. Instead, it refers to an EL expression. The evaluation of that EL expression gives the EL variable its value.

For example, in the following code snippet

```xml
<h:inputText value="#{handler.customer.name}"/>
```
handler refers to a model object that can be resolved by an EL Resolver.

However, in this other example:

```xml
<c:forEach var="item" items="#{model.list}"
    <h:inputText value="#{item.name}"/>
</c:forEach>
```

`item` is an EL variable because it does not refer directly to a model object. Instead, it refers to another EL expression, namely a specific item in the collection referred to by the EL expression `#{model.list}`.

Assuming that there are three elements in `#{model.list}`, this means that for each invocation of `<h:inputText>`, the following information about `item` must be preserved in the `VariableMapper`:

- first invocation: `item` maps to first element in `#{model.list}`
- second invocation: `item` maps to second element in `#{model.list}`
- third invocation: `item` maps to third element in `#{model.list}`

`VariableMapper` provides the mechanisms required to allow the mapping of an EL variable to the EL expression from which it gets its value.
Hierarchy For Package javax.el

Package Hierarchies:
  All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.el.BeanELResolver.BeanProperties`
  - `javax.el.BeanELResolver.BeanProperty`
  - `javax.el.ELContext`
  - `javax.el.ELResolver`
    - `javax.el.ArrayELResolver`
    - `javax.el.BeanELResolver`
    - `javax.el.CompositeELResolver`
    - `javax.el.ListELResolver`
    - `javax.el.MapELResolver`
    - `javax.el.ResourceBundleELResolver`
  - `java.util.EventObject` (implements java.io.Serializable)
    - `javax.el.ELContextEvent`
  - `javax.el.Expression` (implements java.io.Serializable)
    - `javax.el.MethodExpression`
    - `javax.el.ValueExpression`
  - `javax.el.ExpressionFactory`
  - `javax.el.FunctionMapper`
  - `javax.elMethodInfo`
  - `java.lang Throwable` (implements java.io.Serializable)
    - `java.lang.RuntimeException`
      - `javax.el.ELException`
        - `javax.el.MethodNotFoundException`
        - `javax.el.PropertyNotFoundException`
        - `javax.el.PropertyNotWritableException`
  - `javax.el.VariableMapper`
Interface Hierarchy

- java.util.EventListener
- javax.el.ELContextListener
Package javax.xml.registry.infomodel

This package describes the information model for the JAXR API.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Association</td>
<td>A RegistryObject instance may be associated with zero or more RegistryObject instances.</td>
</tr>
<tr>
<td>AuditableEvent</td>
<td>AuditableEvent instances provide a long term record of events that effect a change of state in a RegistryObject.</td>
</tr>
<tr>
<td>Classification</td>
<td>The Classification interface is used to classify RegistryObject instances.</td>
</tr>
<tr>
<td>ClassificationScheme</td>
<td>A ClassificationScheme instance represents a taxonomy that may be used to classify or categorize RegistryObject instances.</td>
</tr>
<tr>
<td>Concept</td>
<td>The Concept interface is used to represent taxonomy elements and their structural relationship with each other in order to describe an internal taxonomy.</td>
</tr>
<tr>
<td>EmailAddress</td>
<td>Represents an email address.</td>
</tr>
<tr>
<td>ExtensibleObject</td>
<td>An ExtensibleObject is one that allows itself to be extended by utilizing dynamically added Slots that add arbitrary attributes to the object on a per instance basis.</td>
</tr>
<tr>
<td>ExternalIdentifier</td>
<td>ExternalIdentifier instances provide the additional identifier information to RegistryObjects such as DUNS number, Social Security Number, or an alias name of the organization.</td>
</tr>
<tr>
<td>ExternalLink</td>
<td>ExternalLink instances model a named URI to content that may reside outside the registry.</td>
</tr>
<tr>
<td>ExtrinsicObjects</td>
<td>ExtrinsicObjects provide metadata that</td>
</tr>
<tr>
<td><strong>ExtrinsicObject</strong></td>
<td>describes submitted content whose type is not intrinsically known to the registry and therefore must be described by means of additional attributes (e.g., mime type).</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>InternationalString</strong></td>
<td>This interface represents a String that has been internationalized into several Locales.</td>
</tr>
<tr>
<td><strong>Key</strong></td>
<td>Represents a unique key that identifies a RegistryObject.</td>
</tr>
<tr>
<td><strong>LocalizedString</strong></td>
<td>This interface is used as a simple wrapper interface that associates a String with its Locale.</td>
</tr>
<tr>
<td><strong>Organization</strong></td>
<td>Organization instances provide information on organizations such as a Submitting Organization.</td>
</tr>
<tr>
<td><strong>PersonName</strong></td>
<td>Represents a person's name.</td>
</tr>
<tr>
<td><strong>PostalAddress</strong></td>
<td>PostalAddress is a simple re-usable entity class that defines attributes of a postal Address.</td>
</tr>
<tr>
<td><strong>RegistryEntry</strong></td>
<td>The RegistryEntry interface is a base interface for interfaces in the model that require additional metadata beyond what is provided by the RegistryObject interface.</td>
</tr>
<tr>
<td><strong>RegistryObject</strong></td>
<td>The RegistryObject class is an abstract base class used by most classes in the model.</td>
</tr>
<tr>
<td><strong>RegistryPackage</strong></td>
<td>RegistryPackage instances are RegistryEntries that group logically related RegistryEntries together.</td>
</tr>
<tr>
<td><strong>Service</strong></td>
<td>Service instances are RegistryObjects that provide information on services (for example, web services) offered by an Organization.</td>
</tr>
<tr>
<td><strong>ServiceBinding</strong></td>
<td>ServiceBinding instances are RegistryObjects that represent technical information on a specific way to access a specific interface offered by a Service instance.</td>
</tr>
<tr>
<td><strong>Slot</strong></td>
<td>Slot instances provide a dynamic way to add arbitrary attributes to RegistryObject instances.</td>
</tr>
<tr>
<td><strong>A SpecificationLink provides the linkage</strong></td>
<td></td>
</tr>
<tr>
<td><strong>SpecificationLink</strong></td>
<td>between a ServiceBinding and one of its technical specifications that describes how to use the service using the ServiceBinding.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>TelephoneNumber</strong></td>
<td>A simple re-usable entity class that defines attributes of a telephone number.</td>
</tr>
<tr>
<td><strong>URIValidator</strong></td>
<td>Defines common behavior expected of any class that validates URIs.</td>
</tr>
<tr>
<td><strong>User</strong></td>
<td>User instances are RegistryObjects that are used to provide information about registered users within the registry.</td>
</tr>
<tr>
<td><strong>Versionable</strong></td>
<td>The Versionable interface defines the behavior common to classes that are capable of creating versions of their instances.</td>
</tr>
</tbody>
</table>
Package javax.xml.registry.infomodel

Description

This package describes the information model for the JAXR API. It is based largely on the ebXML Registry Information Model with enhancements to support other registry specifications such as UDDI.

The class diagram in Figure 1 shows the public view of the information model.

![Figure 1. Information Model Public View]

The class diagram in Figure 2 shows the inheritance view of the information model.

![Figure 2. Information Model Detailed View]

Overview  Class  Tree  Deprecated  Index  Help
Hierarchy For Package
javax.xml.registry.infomodel

Package Hierarchies:
   All Packages
Interface Hierarchy

- javax.xml.registry.infomodel.EmailAddress
- javax.xml.registry.infomodel.ExtensibleObject
  - javax.xml.registry.infomodel.PostalAddress
  - javax.xml.registry.infomodel.RegistryObject
    - javax.xml.registry.infomodel.Association
    - javax.xml.registry.infomodel.AuditableEvent
    - javax.xml.registry.infomodel.Classification
    - javax.xml.registry.infomodel.Concept
    - javax.xml.registry.infomodel.ExternalIdentifier
    - javax.xml.registry.infomodel.ExternalLink (also extends javax.xml.registry.infomodel.URIValidator)
    - javax.xml.registry.infomodel.Organization
    - javax.xml.registry.infomodel.RegistryEntry (also extends javax.xml.registry.infomodel.Versionable)
      - javax.xml.registry.infomodel.ClassificationScheme
      - javax.xml.registry.infomodel.ExtrinsicObject
      - javax.xml.registry.infomodel.RegistryPackage
      - javax.xml.registry.infomodel.Service
    - javax.xml.registry.infomodel.ServiceBinding (also extends javax.xml.registry.infomodel.URIValidator)
    - javax.xml.registry.infomodel.SpecificationLink
    - javax.xml.registry.infomodel.User
- javax.xml.registry.infomodel.InternationalString
- javax.xml.registry.infomodel.Key
- javax.xml.registry.infomodel.LocalizedString
- javax.xml.registry.infomodel.PersonName
- javax.xml.registry.infomodel.Slot
- javax.xml.registry.infomodel.TelephoneNumber
- javax.xml.registry.infomodel.URIValidator
  - javax.xml.registry.infomodel.ExternalLink (also extends javax.xml.registry.infomodel.RegistryObject)
  - javax.xml.registry.infomodel.ServiceBinding (also extends javax.xml.registry.infomodel.RegistryObject)
- javax.xml.registry.infomodel.Versionable
  - javax.xml.registry.infomodel.RegistryEntry (also extends
javax.xml.registry.infomodel.**RegistryObject**
- javax.xml.registry.infomodel.**ClassificationScheme**
- javax.xml.registry.infomodel.**ExtrinsicObject**
- javax.xml.registry.infomodel.**RegistryPackage**
- javax.xml.registry.infomodel.**Service**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.persistence

The javax.persistence package contains the classes and interfaces that define the contracts between a persistence provider and the managed classes and the clients of the Java Persistence API.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntityManager</td>
<td>Interface used to interact with the persistence context.</td>
</tr>
<tr>
<td>EntityManagerFactory</td>
<td>The EntityManagerFactory interface is used by the application to obtain an application-managed entity manager.</td>
</tr>
<tr>
<td>EntityTransaction</td>
<td>The EntityTransaction interface is used to control resource transactions on resource-local entity managers.</td>
</tr>
<tr>
<td>Query</td>
<td>Interface used to control query execution.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Persistence</td>
<td>Bootstrap class that is used to obtain an EntityManagerFactory.</td>
</tr>
</tbody>
</table>

### Enum Summary

<table>
<thead>
<tr>
<th>Enum Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CascadeType</td>
<td>Defines the set of cascadable operations that are propagated to the associated entity.</td>
</tr>
<tr>
<td>DiscriminatorType</td>
<td>Defines supported types of the discriminator column.</td>
</tr>
<tr>
<td>EnumType</td>
<td>Defines mapping for the enumerated types.</td>
</tr>
<tr>
<td></td>
<td>Defines strategies for fetching data from the</td>
</tr>
<tr>
<td><strong>FetchType</strong></td>
<td>database.</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------</td>
</tr>
<tr>
<td><strong>FlushModeType</strong></td>
<td>Flush mode setting.</td>
</tr>
<tr>
<td><strong>GenerationType</strong></td>
<td>Defines the types of primary key generation.</td>
</tr>
<tr>
<td><strong>InheritanceType</strong></td>
<td>Defines inheritance strategy options.</td>
</tr>
<tr>
<td><strong>LockModeType</strong></td>
<td>Lock modes that can be specified by means of the <code>EntityManager.lock()</code> method.</td>
</tr>
<tr>
<td><strong>PersistenceContextType</strong></td>
<td>Specifies whether a transaction-scoped or extended persistence context is to be used in <code>PersistenceContext</code>.</td>
</tr>
<tr>
<td><strong>TemporalType</strong></td>
<td>Type used to indicate a specific mapping of <code>Date</code> or <code>Calendar</code>.</td>
</tr>
</tbody>
</table>

## Exception Summary

<p>| <strong>EntityExistsException</strong> | Thrown by the persistence provider when <code>EntityManager.persist(Object)</code> is called and the entity already exists. |
| <strong>EntityNotFoundException</strong> | Thrown by the persistence provider when an entity reference obtained by <code>EntityManager.getReference(Class, Object)</code> is accessed but the entity does not exist. |
| <strong>NonUniqueResultException</strong> | Thrown by the persistence provider when <code>getSingleResult()</code> is executed on a query and there is more than one result from the query. |
| <strong>NoResultException</strong> | Thrown by the persistence provider when <code>getSingleResult()</code> is executed on a query and there is no result to return. |
| <strong>OptimisticLockException</strong> | Thrown by the persistence provider when an optimistic locking conflict occurs. |
| <strong>PersistenceException</strong> | Thrown by the persistence provider when a problem occurs. |
| <strong>RollbackException</strong> | Thrown by the persistence provider when the <code>EntityTransaction.commit()</code> fails. |</p>
<table>
<thead>
<tr>
<th>Annotation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>TransactionRequiredException</strong></td>
<td>Thrown by the persistence provider when a transaction is required but is not active.</td>
</tr>
<tr>
<td><strong>AssociationOverride</strong></td>
<td>This annotation is used to override a many-to-one or one-to-one mapping of property or field for an entity relationship.</td>
</tr>
<tr>
<td><strong>AssociationOverrides</strong></td>
<td>This annotation is used to override mappings of multiple many-to-one or one-to-one relationship properties or fields.</td>
</tr>
<tr>
<td><strong>AttributeOverride</strong></td>
<td>The <strong>AttributeOverride</strong> annotation is used to override the mapping of a <strong>Basic</strong> (whether explicit or default) property or field or <strong>Id</strong> property or field.</td>
</tr>
<tr>
<td><strong>AttributeOverrides</strong></td>
<td>Is used to override mappings of multiple properties or fields.</td>
</tr>
<tr>
<td><strong>Basic</strong></td>
<td>The <strong>Basic</strong> annotation is the simplest type of mapping to a database column.</td>
</tr>
<tr>
<td><strong>Column</strong></td>
<td>Is used to specify a mapped column for a persistent property or field.</td>
</tr>
<tr>
<td><strong>ColumnResult</strong></td>
<td>References name of a column in the SELECT clause of a SQL query - i.e., column alias, if applicable.</td>
</tr>
<tr>
<td><strong>DiscriminatorColumn</strong></td>
<td>Is used to define the discriminator column for the <strong>SINGLE_TABLE</strong> and <strong>JOINED</strong> inheritance mapping strategies.</td>
</tr>
<tr>
<td><strong>DiscriminatorValue</strong></td>
<td>Is used to specify the value of the discriminator column for entities of the given type.</td>
</tr>
<tr>
<td><strong>Embeddable</strong></td>
<td>Defines a class whose instances are stored as an intrinsic part of an owning entity and share the identity of the entity.</td>
</tr>
<tr>
<td><strong>Embedded</strong></td>
<td>Defines a persistent field or property of an entity whose value is an instance of an embeddable class.</td>
</tr>
<tr>
<td><strong>EmbeddedId</strong></td>
<td>Is applied to a persistent field or property of an entity class or mapped superclass to denote a composite primary key that is an embeddable class.</td>
</tr>
<tr>
<td><strong>Entity</strong></td>
<td>Specifies that the class is an entity.</td>
</tr>
<tr>
<td><strong>EntityListeners</strong></td>
<td>Specifies the callback listener classes to be used for an entity or mapped superclass.</td>
</tr>
<tr>
<td><strong>EntityResult</strong></td>
<td>References an entity in the SELECT clause of a SQL query.</td>
</tr>
<tr>
<td><strong>Enumerated</strong></td>
<td>Specifies that a persistent property or field should be persisted as a enumerated type.</td>
</tr>
<tr>
<td><strong>ExcludeDefaultListeners</strong></td>
<td>Specifies that the invocation of default listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.</td>
</tr>
<tr>
<td><strong>ExcludeSuperclassListeners</strong></td>
<td>Specifies that the invocation of superclass listeners is to be excluded for the entity class (or mapped superclass) and its subclasses.</td>
</tr>
<tr>
<td><strong>FieldResult</strong></td>
<td>Is used to map the columns specified in the SELECT list of the query to the properties or fields of the entity class.</td>
</tr>
<tr>
<td><strong>GeneratedValue</strong></td>
<td>Provides for the specification of generation strategies for the values of primary keys.</td>
</tr>
<tr>
<td><strong>Id</strong></td>
<td>Specifies the primary key property or field of an entity.</td>
</tr>
<tr>
<td><strong>IdClass</strong></td>
<td>Specifies a composite primary key class that is mapped to multiple fields or properties of the entity.</td>
</tr>
<tr>
<td>Annotation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Inheritance</strong></td>
<td>Defines the inheritance strategy to be used for an entity class hierarchy.</td>
</tr>
<tr>
<td><strong>JoinColumn</strong></td>
<td>Is used to specify a mapped column for joining an entity association.</td>
</tr>
<tr>
<td><strong>JoinColumns</strong></td>
<td>Defines mapping for the composite foreign keys.</td>
</tr>
<tr>
<td><strong>JoinTable</strong></td>
<td>This annotation is used in the mapping of associations.</td>
</tr>
<tr>
<td><strong>Lob</strong></td>
<td>Specifies that a persistent property or field should be persisted as a large object to a database-supported large object type.</td>
</tr>
<tr>
<td><strong>ManyToMany</strong></td>
<td>Defines a many-valued association with many-to-many multiplicity.</td>
</tr>
<tr>
<td><strong>ManyToOne</strong></td>
<td>This annotation defines a single-valued association to another entity class that has many-to-one multiplicity.</td>
</tr>
<tr>
<td><strong>MapKey</strong></td>
<td>Is used to specify the map key for associations of type <code>Map</code>.</td>
</tr>
<tr>
<td><strong>MappedSuperclass</strong></td>
<td>Designates a class whose mapping information is applied to the entities that inherit from it.</td>
</tr>
<tr>
<td><strong>NamedNativeQueries</strong></td>
<td>Is used to specify an array of native SQL named queries.</td>
</tr>
<tr>
<td><strong>NamedNativeQuery</strong></td>
<td>Is used to specify a native SQL named query.</td>
</tr>
<tr>
<td><strong>NamedQueries</strong></td>
<td>Specifies an array of named Java Persistence query language queries.</td>
</tr>
<tr>
<td><strong>NamedQuery</strong></td>
<td>Is used to specify a named query in the Java Persistence query language, which is a static query expressed in metadata.</td>
</tr>
<tr>
<td><strong>OneToMany</strong></td>
<td>Defines a many-valued association with one-to-many multiplicity.</td>
</tr>
<tr>
<td><strong>OneToOne</strong></td>
<td>This annotation defines a single-valued association to another entity that has</td>
</tr>
<tr>
<td>Annotation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>one-to-one multiplicity.</td>
<td></td>
</tr>
<tr>
<td><strong>OrderBy</strong></td>
<td>This annotation specifies the ordering of the elements of a collection valued association at the point when the association is retrieved.</td>
</tr>
<tr>
<td><strong>PersistenceContext</strong></td>
<td>Expresses a dependency on an <a href="https://docs.oracle.com/en/middleware/express/ee/12.2.1/guid-6716552.html">EntityManager</a> persistence context.</td>
</tr>
<tr>
<td><strong>PersistenceContexts</strong></td>
<td>Declares one or more <a href="https://docs.oracle.com/en/middleware/express/ee/12.2.1/guid-6716552.html">PersistenceContext</a> annotations.</td>
</tr>
<tr>
<td><strong>PersistenceProperty</strong></td>
<td>Describes a single container or persistence provider property.</td>
</tr>
<tr>
<td><strong>PersistenceUnit</strong></td>
<td>Expresses a dependency on an <a href="https://docs.oracle.com/en/middleware/express/ee/12.2.1/guid-6716552.html">EntityManagerFactory</a>.</td>
</tr>
<tr>
<td><strong>PersistenceUnits</strong></td>
<td>Declares one or more <a href="https://docs.oracle.com/en/middleware/express/ee/12.2.1/guid-6716552.html">PersistenceUnit</a> annotations.</td>
</tr>
<tr>
<td><strong>PostLoad</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PostPersist</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PostRemove</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PostUpdate</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PrePersist</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PreRemove</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PreUpdate</strong></td>
<td>Is used to specify callback methods for the corresponding lifecycle event.</td>
</tr>
<tr>
<td><strong>PrimaryKeyJoinColumn</strong></td>
<td>This annotation specifies a primary key column that is used as a foreign key to join to another table.</td>
</tr>
<tr>
<td><strong>PrimaryKeyJoinColumn</strong></td>
<td>This annotation groups <a href="https://docs.oracle.com/en/middleware/express/ee/12.2.1/guid-6716552.html">PrimaryKeyJoinColumn</a> annotations.</td>
</tr>
<tr>
<td>Annotation</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>QueryHint</strong></td>
<td>An implementation-specific query hint.</td>
</tr>
<tr>
<td><strong>SecondaryTable</strong></td>
<td>This annotation is used to specify a secondary table for the annotated entity class.</td>
</tr>
<tr>
<td><strong>SecondaryTables</strong></td>
<td>This annotation is used to specify multiple secondary tables for an entity.</td>
</tr>
<tr>
<td><strong>SequenceGenerator</strong></td>
<td>This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the <code>GeneratedValue</code> annotation.</td>
</tr>
<tr>
<td><strong>SqlResultSetMapping</strong></td>
<td>This annotation is used to specify the mapping of the result of a native SQL query.</td>
</tr>
<tr>
<td><strong>SqlResultSetMappings</strong></td>
<td>This annotation is used to define one or more <code>SqlResultSetMapping</code>.</td>
</tr>
<tr>
<td><strong>Table</strong></td>
<td>This annotation specifies the primary table for the annotated entity.</td>
</tr>
<tr>
<td><strong>TableGenerator</strong></td>
<td>This annotation defines a primary key generator that may be referenced by name when a generator element is specified for the <code>GeneratedValue</code> annotation.</td>
</tr>
<tr>
<td><strong>Temporal</strong></td>
<td>This annotation must be specified for persistent fields or properties of type <code>Date</code> and <code>Calendar</code>.</td>
</tr>
<tr>
<td><strong>Transient</strong></td>
<td>This annotation specifies that the property or field is not persistent.</td>
</tr>
<tr>
<td><strong>UniqueConstraint</strong></td>
<td>This annotation is used to specify that a unique constraint is to be included in the generated DDL for a primary or secondary table.</td>
</tr>
<tr>
<td><strong>Version</strong></td>
<td>This annotation specifies the version field or property of an entity class that serves as its optimistic lock value.</td>
</tr>
</tbody>
</table>
The `javax.persistence` package contains the classes and interfaces that define the contracts between a persistence provider and the managed classes and the clients of the Java Persistence API.
Hierarchy For Package javax.persistence

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.persistence.**Persistence**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
    - java.lang.**RuntimeException**
      - javax.persistence.**PersistenceException**
        - javax.persistence.**EntityExistsException**
        - javax.persistence.**EntityNotFoundException**
        - javax.persistence.**NonUniqueResultException**
        - javax.persistence.**NoResultException**
        - javax.persistence.**OptimisticLockException**
        - javax.persistence.**RollbackException**
        - javax.persistence.**TransactionRequiredException**
Interface Hierarchy

- javax.persistence.**EntityManager**
- javax.persistence.**EntityManagerFactory**
- javax.persistence.**EntityTransaction**
- javax.persistence.**Query**
Annotation Type Hierarchy

- `javax.persistence.AssociationOverride` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.AssociationOverrides` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.AttributeOverride` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.AttributeOverrides` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Basic` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Column` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.ColumnResult` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.DiscriminatorColumn` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.DiscriminatorValue` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Embeddable` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Embedded` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.EmbeddedId` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Entity` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.EntityListeners` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.EntityResult` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Enumerated` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.ExcludeDefaultListeners` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.ExcludeSuperclassListeners` (implements `java.lang.annotation.Annotation`
java.lang.annotation.Annotation

- javax.persistence.FieldResult (implements java.lang.annotation.Annotation)
- javax.persistence.GeneratedValue (implements java.lang.annotation.Annotation)
- javax.persistence.Id (implements java.lang.annotation.Annotation)
- javax.persistence.IdClass (implements java.lang.annotation.Annotation)
- javax.persistence.Inheritance (implements java.lang.annotation.Annotation)
- javax.persistence.JoinColumn (implements java.lang.annotation.Annotation)
- javax.persistence.JoinColumns (implements java.lang.annotation.Annotation)
- javax.persistence.JoinTable (implements java.lang.annotation.Annotation)
- javax.persistence.Lob (implements java.lang.annotation.Annotation)
- javax.persistence.ManyToMany (implements java.lang.annotation.Annotation)
- javax.persistence.ManyToOne (implements java.lang.annotation.Annotation)
- javax.persistence.MapKey (implements java.lang.annotation.Annotation)
- javax.persistence.MappedSuperclass (implements java.lang.annotation.Annotation)
- javax.persistence.NamedNativeQueries (implements java.lang.annotation.Annotation)
- javax.persistence.NamedNativeQuery (implements java.lang.annotation.Annotation)
- javax.persistence.NamedQueries (implements java.lang.annotation.Annotation)
- javax.persistence.NamedQuery (implements java.lang.annotation.Annotation)
- javax.persistence.OneToMany (implements java.lang.annotation.Annotation)
- javax.persistence.OneToOne (implements java.lang.annotation.Annotation)
- javax.persistence.OrderBy (implements java.lang.annotation.Annotation)
- `javax.persistence.PersistenceContext` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceContexts` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceProperty` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceUnit` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceUnits` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PostLoad` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PostPersist` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PostRemove` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PostUpdate` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PrePersist` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PreRemove` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PreUpdate` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PrimaryKeyJoinColumn` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PrimaryKeyJoinColumns` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.QueryHint` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.SecondaryTable` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.SecondaryTables` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.SequenceGenerator` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.SqlResultSetMapping` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.SqlResultSetMappings` (implements `java.lang.annotation.Annotation`
java.lang.annotation.Annotation

- javax.persistence.Table (implements java.lang.annotation.Annotation)
- javax.persistence.TableGenerator (implements java.lang.annotation.Annotation)
- javax.persistence.Temporal (implements java.lang.annotation.Annotation)
- javax.persistence.Transient (implements java.lang.annotation.Annotation)
- javax.persistence.UniqueConstraint (implements java.lang.annotation.Annotation)
- javax.persistence.Version (implements java.lang.annotation.Annotation)
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
    - javax.persistence.**CascadeType**
    - javax.persistence.**DiscriminatorType**
    - javax.persistence.**EnumType**
    - javax.persistence.**FetchType**
    - javax.persistence.**FlushModeType**
    - javax.persistence.**GenerationType**
    - javax.persistence.**InheritanceType**
    - javax.persistence.**LockModeType**
    - javax.persistence.**PersistenceContextType**
    - javax.persistence.**TemporalType**
Package javax.xml.ws

This package contains the core JAX-WS APIs.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AsyncHandler&lt;T&gt;</strong></td>
<td>The <code>AsyncHandler</code> interface is implemented by clients that wish to receive callback notification of the completion of service endpoint operations invoked asynchronously.</td>
</tr>
<tr>
<td><strong>Binding</strong></td>
<td>The <code>Binding</code> interface is the base interface for JAX-WS protocol bindings.</td>
</tr>
<tr>
<td><strong>BindingProvider</strong></td>
<td>The <code>BindingProvider</code> interface provides access to the protocol binding and associated context objects for request and response message processing.</td>
</tr>
<tr>
<td><strong>Dispatch&lt;T&gt;</strong></td>
<td>The <code>Dispatch</code> interface provides support for the dynamic invocation of a service endpoint operations.</td>
</tr>
<tr>
<td><strong>LogicalMessage</strong></td>
<td>The <code>LogicalMessage</code> interface represents a protocol agnostic XML message and contains methods that provide access to the payload of the message.</td>
</tr>
<tr>
<td><strong>Provider&lt;T&gt;</strong></td>
<td>Service endpoints may implement the <code>Provider</code> interface as a dynamic alternative to an SEI.</td>
</tr>
<tr>
<td><strong>Response&lt;T&gt;</strong></td>
<td>The <code>Response</code> interface provides methods used to obtain the payload and context of a message sent in response to an operation invocation.</td>
</tr>
<tr>
<td><strong>WebServiceContext</strong></td>
<td>A <code>WebServiceContext</code> makes it possible for a web service endpoint implementation class to access message context and security information relative to a request being served.</td>
</tr>
</tbody>
</table>
Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Endpoint</td>
<td>A Web service endpoint.</td>
</tr>
<tr>
<td>Holder&lt;T&gt;</td>
<td>Holds a value of type T.</td>
</tr>
<tr>
<td>Service</td>
<td>Service objects provide the client view of a Web service.</td>
</tr>
<tr>
<td>WebServicePermission</td>
<td>This class defines web service permissions.</td>
</tr>
</tbody>
</table>

Enum Summary

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Service.Mode</td>
<td>The orientation of a dynamic client or service.</td>
</tr>
</tbody>
</table>

Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ProtocolException</td>
<td>The ProtocolException class is a base class for exceptions related to a specific protocol binding.</td>
</tr>
<tr>
<td>WebServiceException</td>
<td>The WebServiceException class is the base exception class for all JAX-WS API runtime exceptions.</td>
</tr>
</tbody>
</table>

Annotation Types Summary

<table>
<thead>
<tr>
<th>AnnotationType</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BindingType</td>
<td>The BindingType annotation is used to specify the binding to use for a web service endpoint implementation class.</td>
</tr>
<tr>
<td>RequestWrapper</td>
<td>Used to annotate methods in the Service Endpoint Interface with the request wrapper bean to be used at runtime.</td>
</tr>
<tr>
<td>ResponseWrapper</td>
<td>Used to annotate methods in the Service Endpoint Interface with the response wrapper bean to be used at runtime.</td>
</tr>
<tr>
<td>Provider</td>
<td>Used to indicate whether a Provider is a JAX-WS provider.</td>
</tr>
<tr>
<td><strong>ServiceMode</strong></td>
<td>Implementation wishes to work with entire protocol messages or just with protocol message payloads.</td>
</tr>
<tr>
<td>-----------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>WebEndpoint</strong></td>
<td>Used to annotate the <code>getPortName()</code> methods of a generated service interface.</td>
</tr>
<tr>
<td><strong>WebFault</strong></td>
<td>Used to annotate service specific exception classes to customize to the local and namespace name of the fault element and the name of the fault bean.</td>
</tr>
<tr>
<td><strong>WebServiceClient</strong></td>
<td>Used to annotate a generated service interface.</td>
</tr>
<tr>
<td><strong>WebServiceProvider</strong></td>
<td>Used to annotate a Provider implementation class.</td>
</tr>
<tr>
<td><strong>WebServiceRef</strong></td>
<td>The <code>WebServiceRef</code> annotation is used to define a reference to a web service and (optionally) an injection target for it.</td>
</tr>
<tr>
<td><strong>WebServiceRefs</strong></td>
<td>The <code>WebServiceRefs</code> annotation allows multiple web service references to be declared at the class level.</td>
</tr>
</tbody>
</table>
Package javax.xml.ws Description

This package contains the core JAX-WS APIs.
Hierarchy For Package javax.xml.ws

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.ws.**Endpoint**
  - javax.xml.ws.**Holder**<T>
  - java.security.**Permission** (implements java.security.**Guard**, java.io.**Serializable**)
    - java.security.**BasicPermission** (implements java.io.**Serializable**)
    - javax.xml.ws.**WebServicePermission**
  - javax.xml.ws.**Service**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - java.lang.**RuntimeException**
        - javax.xml.ws.**WebServiceException**
        - javax.xml.ws.**ProtocolException**
Interface Hierarchy

- `javax.xml.ws.AsyncHandler<T>`
- `javax.xml.ws.Binding`
- `javax.xml.ws.BindingProvider`  
  - `javax.xml.ws.Dispatch<T>`
- `java.util.concurrent.Future<V>`  
  - `javax.xml.ws.Response<T>`
- `javax.xml.ws.LogicalMessage`
- `javax.xml.ws.Provider<T>`
- `javax.xml.ws.WebServiceContext`
Annotation Type Hierarchy

- `javax.xml.ws.BindingType` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.RequestWrapper` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.ResponseWrapper` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.ServiceMode` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebEndpoint` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebFault` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebServiceClient` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebServiceProvider` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebServiceRef` (implements `java.lang.annotation.Annotation`)
- `javax.xml.ws.WebServiceRefs` (implements `java.lang.annotation.Annotation`
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
  - javax.xml.ws.**Service.Mode**
Package `javax.xml.bind.attachment`

This package is implemented by a MIME-based package processor that enables the interpretation and creation of optimized binary data within an MIME-based package format.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AttachmentMarshaller</td>
<td>Enable JAXB marshalling to optimize storage of binary data.</td>
</tr>
<tr>
<td>AttachmentUnmarshaller</td>
<td>Enables JAXB unmarshalling of a root document containing optimized binary data formats.</td>
</tr>
</tbody>
</table>
Package javax.xml.bind.attachment Description

This package is implemented by a MIME-based package processor that enables the interpretation and creation of optimized binary data within an MIME-based package format.

Soap MTOM[1], XOP([2][3]) and WS-I AP[4] standardize approaches to optimized transmission of binary datatypes as an attachment. To optimally support these standards within a message passing environment, this package enables an integrated solution between a MIME-based package processor and JAXB unmarshall/marshal processes.
Package Specification

- JAXB Specification
Related Standards

- [1]SOAP Message Transmission Optimization Mechanism
- [2]XML-binary Optimized Packaging
- [4]Describing Media Content of Binary Data in XML

Since:
   JAXB 2.0

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.xml.bind.attachment

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.bind.attachment.**AttachmentMarshaller**
  - javax.xml.bind.attachment.**AttachmentUnmarshaller**
Package javax.xml.soap

Provides the API for creating and building SOAP messages.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Detail</strong></td>
</tr>
<tr>
<td><strong>DetailEntry</strong></td>
</tr>
<tr>
<td><strong>Name</strong></td>
</tr>
<tr>
<td><strong>Node</strong></td>
</tr>
<tr>
<td><strong>SOAPBody</strong></td>
</tr>
<tr>
<td><strong>SOAPBodyElement</strong></td>
</tr>
<tr>
<td><strong>SOAPConstants</strong></td>
</tr>
<tr>
<td><strong>SOAPElement</strong></td>
</tr>
<tr>
<td><strong>SOAPEnvelope</strong></td>
</tr>
<tr>
<td><strong>SOAPFault</strong></td>
</tr>
<tr>
<td><strong>SOAPFaultElement</strong></td>
</tr>
<tr>
<td><strong>SOAPHeader</strong></td>
</tr>
<tr>
<td><strong>SOAPHeaderElement</strong></td>
</tr>
<tr>
<td><strong>Text</strong></td>
</tr>
</tbody>
</table>
### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AttachmentPart</strong></td>
<td>A single attachment to a SOAPMessage object.</td>
</tr>
<tr>
<td><strong>MessageFactory</strong></td>
<td>A factory for creating SOAPMessage objects.</td>
</tr>
<tr>
<td><strong>MimeHeader</strong></td>
<td>An object that stores a MIME header name and its value.</td>
</tr>
<tr>
<td><strong>MimeHeaders</strong></td>
<td>A container for MimeHeader objects, which represent the MIME headers present in a MIME part of a message.</td>
</tr>
<tr>
<td><strong>SAAJMetaFactory</strong></td>
<td>The access point for the implementation classes of the factories defined in the SAAJ API.</td>
</tr>
<tr>
<td><strong>SAAJResult</strong></td>
<td>Acts as a holder for the results of a JAXP transformation or a JAXB marshalling, in the form of a SAAJ tree.</td>
</tr>
<tr>
<td><strong>SOAPConnection</strong></td>
<td>A point-to-point connection that a client can use for sending messages directly to a remote party (represented by a URL, for instance).</td>
</tr>
<tr>
<td><strong>SOAPConnectionFactory</strong></td>
<td>A factory for creating SOAPConnection objects.</td>
</tr>
<tr>
<td><strong>SOAPElementFactory</strong></td>
<td>Deprecated. - Use javax.xml.soap.SOAPFactory for creating SOAPElements.</td>
</tr>
<tr>
<td><strong>SOAPFactory</strong></td>
<td>SOAPFactory is a factory for creating various objects that exist in the SOAP XML tree.</td>
</tr>
<tr>
<td><strong>SOAPMessage</strong></td>
<td>The root class for all SOAP messages.</td>
</tr>
<tr>
<td><strong>SOAPPart</strong></td>
<td>The container for the SOAP-specific portion of a SOAPMessage object.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOAPException</strong></td>
<td>An exception that signals that a SOAP exception has occurred.</td>
</tr>
</tbody>
</table>
Package javax.xml.soap Description

Provides the API for creating and building SOAP messages. This package is defined in the SOAP with Attachments API for Java™ (SAAJ) 1.3 specification.

The API in the javax.xml.soap package allows you to do the following:

- create a point-to-point connection to a specified endpoint
- create a SOAP message
- create an XML fragment
- add content to the header of a SOAP message
- add content to the body of a SOAP message
- create attachment parts and add content to them
- access/add/modify parts of a SOAP message
- create/add/modify SOAP fault information
- extract content from a SOAP message
- send a SOAP request-response message

In addition the APIs in the javax.xml.soap package extend their counterparts in the org.w3c.dom package. This means that the SOAPPart of a SOAPMessage is also a DOM Level 2 Document, and can be manipulated as such by applications, tools and libraries that use DOM (see http://www.w3.org/DOM/ for more information). It is important to note that, while it is possible to use DOM APIs to add ordinary DOM nodes to a SAAJ tree, the SAAJ APIs are still required to return SAAJ types when examining or manipulating the tree. In order to accomplish this the SAAJ APIs (specifically SOAPElement.getChildElements()) are allowed to silently replace objects that are incorrectly typed relative to SAAJ requirements with equivalent objects of the required type. These replacements must never cause the logical structure of the tree to change, so from the perspective of the DOM APIs the tree will remain unchanged. However, the physical composition of the tree will have changed so that references to the nodes that were replaced will refer to nodes that are no longer a part of the tree. The SAAJ APIs are not allowed to make these replacements if they are not required so the replacement objects will never subsequently be silently replaced by future calls to the SAAJ API.
What this means in practical terms is that an application that starts to use SAAJ APIs on a tree after manipulating it using DOM APIs must assume that the tree has been translated into an all SAAJ tree and that any references to objects within the tree that were obtained using DOM APIs are no longer valid. Switching from SAAJ APIs to DOM APIs is not allowed to cause invalid references and neither is using SAAJ APIs exclusively. It is only switching from using DOM APIs on a particular SAAJ tree to using SAAJ APIs that causes the risk of invalid references.
Hierarchy For Package javax.xml.soap

Package Hierarchies:

All Packages

________________________________________
Class Hierarchy

- java.lang.**Object**
  - javax.xml.soap.**AttachmentPart**
  - javax.xml.transform.dom.**DOMResult** (implements javax.xml.transform.**Result**)
    - javax.xml.soap.**SAAJResult**
  - javax.xml.soap.**MessageFactory**
  - javax.xml.soap.**MimeHeader**
  - javax.xml.soap.**MimeHeaders**
  - javax.xml.soap.**SAAJMetaFactory**
  - javax.xml.soap.**SOAPConnection**
  - javax.xml.soap.**SOAPConnectionFactory**
  - javax.xml.soap.**SOAPElementFactory**
  - javax.xml.soap.**SOAPFactory**
  - javax.xml.soap.**SOAPMessage**
  - javax.xml.soap.**SOAPPart** (implements org.w3c.dom.**Document**,javax.xml.soap.**Node**)
- java.lang.**Throwable** (implements java.io.**Serializable**)
  - java.lang.**Exception**
    - javax.xml.soap.**SOAPException**
Interface Hierarchy

- javax.xml.soap.Name
- org.w3c.dom.Node
  - org.w3c.dom.CharacterData
    - org.w3c.dom.Text
      - javax.xml.soap.Text (also extends javax.xml.soap.Node)
  - org.w3c.dom.Element
    - javax.xml.soap.SOAPElement (also extends javax.xml.soap.Node)
      - javax.xml.soap.DetailEntry
      - javax.xml.soap.SOAPBody
    - javax.xml.soap.SOAPBodyElement
      - javax.xml.soap.SOAPFault
    - javax.xml.soap.SOAPEnvelope
    - javax.xml.soap.SOAPFaultElement
      - javax.xml.soap.Detail
    - javax.xml.soap.SOAPHeader
    - javax.xml.soap.SOAPHeaderElement
  - javax.xml.soap.Node
    - javax.xml.soap.SOAPElement (also extends org.w3c.dom.Element)
      - javax.xml.soap.DetailEntry
      - javax.xml.soap.SOAPBody
    - javax.xml.soap.SOAPBodyElement
      - javax.xml.soap.SOAPFault
    - javax.xml.soap.SOAPEnvelope
    - javax.xml.soap.SOAPFaultElement
      - javax.xml.soap.Detail
    - javax.xml.soap.SOAPHeader
    - javax.xml.soap.SOAPHeaderElement
  - javax.xml.soap.Text (also extends org.w3c.dom.Text)
- javax.xml.soap.SOAPConstants
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
# Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Attribute</strong></td>
<td>An interface that contains information about an attribute.</td>
</tr>
<tr>
<td><strong>Characters</strong></td>
<td>This describes the interface to Characters events.</td>
</tr>
<tr>
<td><strong>Comment</strong></td>
<td>An interface for comment events</td>
</tr>
<tr>
<td><strong>DTD</strong></td>
<td>This is the top level interface for events dealing with DTDs</td>
</tr>
<tr>
<td><strong>EndDocument</strong></td>
<td>A marker interface for the end of the document</td>
</tr>
<tr>
<td><strong>EndElement</strong></td>
<td>An interface for the end element event</td>
</tr>
<tr>
<td><strong>EntityDeclaration</strong></td>
<td>An interface for handling Entity Declarations</td>
</tr>
<tr>
<td></td>
<td>This interface is used to record and report unparsed entity declarations.</td>
</tr>
<tr>
<td><strong>EntityReference</strong></td>
<td>An interface for handling Entity events</td>
</tr>
<tr>
<td><strong>Namespace</strong></td>
<td>An interface that contains information about a namespace.</td>
</tr>
<tr>
<td><strong>NotationDeclaration</strong></td>
<td>An interface for handling Notation Declarations</td>
</tr>
<tr>
<td></td>
<td>Receive notification of a notation declaration event.</td>
</tr>
<tr>
<td><strong>ProcessingInstruction</strong></td>
<td>An interface that describes the data found in processing instructions</td>
</tr>
<tr>
<td><strong>StartDocument</strong></td>
<td>An interface for the start document event</td>
</tr>
<tr>
<td><strong>StartElement</strong></td>
<td>The StartElement interface provides access to information about start elements.</td>
</tr>
<tr>
<td><strong>XMLEvent</strong></td>
<td>This is the base event interface for handling markup events.</td>
</tr>
</tbody>
</table>
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.stream.events

Package Hierarchies:

   All Packages
Interface Hierarchy

- javax.xml.stream.XMLStreamConstants
  - javax.xml.stream.events.XMLEvent
    - javax.xml.stream.events.Attribute
    - javax.xml.stream.events.Namespace
  - javax.xml.stream.events.Characters
  - javax.xml.stream.events.Comment
  - javax.xml.stream.events.DTD
  - javax.xml.stream.events.EndDocument
  - javax.xml.stream.events.EndElement
  - javax.xml.stream.events.EntityDeclaration
  - javax.xml.stream.events.EntityReference
  - javax.xml.stream.events.NotationDeclaration
  - javax.xml.stream.events.ProcessingInstruction
  - javax.xml.stream.events.StartDocument
  - javax.xml.stream.events.StartElement
Package javax.faces.webapp

Classes required for integration of JavaServer Faces into web applications, including a standard servlet, base classes for JSP custom component tags, and concrete tag implementations for core tags.

See: [Description](#)

<table>
<thead>
<tr>
<th>Class Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>AttributeTag</strong></td>
<td><strong>Deprecated. The Faces implementation must now provide the implementation for this class.</strong></td>
</tr>
<tr>
<td><strong>ConverterELTag</strong></td>
<td><strong>ConverterELTag</strong> is a base class for all JSP custom actions that create and register a Converter instance on the ValueHolder associated with our most immediate surrounding instance of a tag whose implementation class is a subclass of <a href="#">UIComponentClassicTagBase</a>.</td>
</tr>
<tr>
<td><strong>ConverterTag</strong></td>
<td><strong>Deprecated. This has been partially replaced by ConverterELTag.</strong></td>
</tr>
<tr>
<td><strong>FacesServlet</strong></td>
<td><strong>FacesServlet</strong> is a servlet that manages the request processing lifecycle for web applications that are utilizing JavaServer Faces to construct the user interface.</td>
</tr>
<tr>
<td><strong>FacetTag</strong></td>
<td><strong>FacetTag</strong> is the JSP mechanism for denoting a <a href="#">UIComponent</a> is to be added as a facet to the component associated with its parent.</td>
</tr>
<tr>
<td><strong>UIComponentBodyTag</strong></td>
<td><strong>Deprecated. All component tags now implement BodyTag.</strong></td>
</tr>
<tr>
<td><strong>UIComponentClassicTagBase</strong></td>
<td><strong>UIComponentTagBase</strong> is the base class for all JSP tags that use the &quot;classic&quot; JSP tag interface that correspond to a</td>
</tr>
<tr>
<td>Class</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>UIComponentELTag</strong></td>
<td><code>UIComponentELTag</code> specializes its superclass to allow for properties that take their values from EL API expressions.</td>
</tr>
<tr>
<td><strong>UIComponentTag</strong></td>
<td><strong>Deprecated.</strong> Use of this class has been replaced with <code>UIComponentELTag</code>, which extends <code>UIComponentClassicTagBase</code> to add properties that use the EL API introduced as part of JSP 2.1.</td>
</tr>
<tr>
<td><strong>UIComponentTagBase</strong></td>
<td><code>UIComponentTagBase</code> is the base class for all JSP tags that correspond to a <code>UIComponent</code> instance in the view.</td>
</tr>
<tr>
<td><strong>ValidatorELTag</strong></td>
<td><code>ValidatorELTag</code> is a base class for all JSP custom actions that create and register a Validator instance on the <code>EditableValueHolder</code> associated with our most immediate surrounding instance of a tag whose implementation class is a subclass of <code>UIComponentTag</code>.</td>
</tr>
<tr>
<td><strong>ValidatorTag</strong></td>
<td><strong>Deprecated.</strong> This has been partially replaced by <code>ValidatorELTag</code>.</td>
</tr>
</tbody>
</table>
Package javax.faces.webapp Description

Classes required for integration of JavaServer Faces into web applications, including a standard servlet, base classes for JSP custom component tags, and concrete tag implementations for core tags.
Hierarchy For Package javax.faces.webapp

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.webapp.**FacesServlet** (implements javax.servlet.**Servlet**)
  - javax.servlet.jsp.tagext.**TagSupport** (implements javax.servlet.jsp.tagext.**IterationTag**, java.io.**Serializable**)
    - javax.faces.webapp.**AttributeTag**
    - javax.faces.webapp.**ConverterELTag**
    - javax.faces.webapp.**ConverterTag**
    - javax.faces.webapp.**FacetTag**
    - javax.faces.webapp.**ValidatorELTag**
    - javax.faces.webapp.**ValidatorTag**
  - javax.faces.webapp.**UIComponentTagBase** (implements javax.servlet.jsp.tagext.**JspTag**)
    - javax.faces.webapp.**UIComponentClassicTagBase** (implements javax.servlet.jsp.tagext.**BodyTag**, javax.servlet.jsp.tagext.**JspIdConsumer**)
      - javax.faces.webapp.**UIComponentELTag** (implements javax.servlet.jsp.tagext.**Tag**)
      - javax.faces.webapp.**UIComponentTag** (implements javax.servlet.jsp.tagext.**Tag**)
        - javax.faces.webapp.**UIComponentBodyTag**
Constant Field Values

Contents

- `javax.el.*`
- `javax.faces.*`
- `javax.jms.*`
- `javax.mail.*`
- `javax.persistence.*`
- `javax.resource.*`
- `javax.servlet.*`
- `javax.transaction.*`
- `javax.xml.*`

### `javax.el.*`

<table>
<thead>
<tr>
<th>ELResolver</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String RESOLVABLE_AT_DESIGN_TIME</td>
</tr>
<tr>
<td></td>
<td>&quot;resolvableAtDesignTime&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String TYPE</td>
</tr>
<tr>
<td></td>
<td>&quot;type&quot;</td>
</tr>
</tbody>
</table>

### `javax.faces.*`

<table>
<thead>
<tr>
<th>FactoryFinder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String APPLICATION_FACTORY</td>
</tr>
<tr>
<td></td>
<td>&quot;javax.faces.application.ApplicationFactory&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String FACES_CONTEXT_FACTORY</td>
</tr>
<tr>
<td></td>
<td>&quot;javax.faces.context.FacesContextFactory&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String LIFECYCLE_FACTORY</td>
</tr>
<tr>
<td></td>
<td>&quot;javax.faces.lifecycle.LifecycleFactory&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String RENDER_KIT_FACTORY</td>
</tr>
<tr>
<td></td>
<td>&quot;javax.faces.render.RenderKitFactory&quot;</td>
</tr>
</tbody>
</table>

### `javax.faces.application.FacesMessage`

<table>
<thead>
<tr>
<th>FacesMessage</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String FACES_MESSAGES</td>
</tr>
<tr>
<td></td>
<td>&quot;javax.faces.Messages&quot;</td>
</tr>
</tbody>
</table>

### `javax.faces.application.StateManager`

<table>
<thead>
<tr>
<th>StateManager</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String STATE_SAVING_METHOD_CLIENT</td>
</tr>
</tbody>
</table>
public static final String STATE_SAVING_METHOD_PARAM_NAME = "javax.faces.STATE_SAVING_METHOD";

javax.faces.application.ViewHandler
public static final String CHARACTER_ENCODING_KEY = "javax.faces.request.encoding";
public static final String DEFAULT_SUFFIX = "javax.faces.DEFAULT_SUFFIX";

javax.faces.component.NamingContainer
public static final char SEPARATOR_CHAR = 58;

javax.faces.component.UIColumn
public static final String COMPONENT_FAMILY = "javax.faces.Column";
public static final String COMPONENT_TYPE = "javax.faces.Column";

javax.faces.component.UIColumn
public static final String COMPONENT_FAMILY = "javax.faces.Command";
public static final String COMPONENT_TYPE = "javax.faces.Command";

javax.faces.component.UICommand
public static final String COMPONENT_FAMILY = "javax.faces.Data";
public static final String COMPONENT_TYPE = "javax.faces.Data";

javax.faces.component.UIForm
public static final String COMPONENT_FAMILY = "javax.faces.Graphic";
public static final String COMPONENT_TYPE = "javax.faces.Graphic";

javax.faces.component.UIInput
public static final String COMPONENT_FAMILY = "javax.faces.Input";
public static final String REQUIRED_MESSAGE_ID = "javax.faces.component.UIInput.REQUIRED";
public static final String CONVERSION_MESSAGE_ID = "javax.faces.component.UIInput.CONVERSION";

javax.faces.component.UIGraphic
public static final String COMPONENT_FAMILY = "javax.faces.component.UIInput";
public static final String COMPONENT_TYPE = "javax.faces.component.UIInput";

javax.faces.component.UIInput

<table>
<thead>
<tr>
<th>Java Class</th>
<th>Component Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.faces.component.UIMessage</td>
<td>String</td>
<td>COMPONENT_FAMILY: javax.faces.Message</td>
</tr>
<tr>
<td>javax.faces.component.UIMessages</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.Messages</td>
</tr>
<tr>
<td>javax.faces.component.UINamingContainer</td>
<td>String</td>
<td>COMPONENT_FAMILY: javax.faces.NamingContainer</td>
</tr>
<tr>
<td>javax.faces.component.UIOutput</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.Output</td>
</tr>
<tr>
<td>javax.faces.component.UIPanel</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.Panel</td>
</tr>
<tr>
<td>javax.faces.component.UIParameter</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.Parameter</td>
</tr>
<tr>
<td>javax.faces.component.UISelectBoolean</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.SelectBoolean</td>
</tr>
<tr>
<td>javax.faces.component.UISelectItem</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.SelectItem</td>
</tr>
<tr>
<td>javax.faces.component.UISelectItems</td>
<td>String</td>
<td>COMPONENT_TYPE: javax.faces.SelectItems</td>
</tr>
<tr>
<td>javax.faces.component.UISelectMany</td>
<td>javax.faces.component.UISelectMany</td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------</td>
<td></td>
</tr>
<tr>
<td>public static final String COMPONENT_FAMILY</td>
<td>&quot;javax.faces.SelectItems&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.SelectItems&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String INVALID_MESSAGE_ID</td>
<td>&quot;javax.faces.component.UISelectMany.INVALID&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.UISelectOne</th>
<th>javax.faces.component.UISelectOne</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_FAMILY</td>
<td>&quot;javax.faces.SelectMany&quot;</td>
</tr>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.SelectMany&quot;</td>
</tr>
<tr>
<td>public static final String INVALID_MESSAGE_ID</td>
<td>&quot;javax.faces.component.UISelectOne.INVALID&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component UIViewRoot</th>
<th>javax.faces.component UIViewRoot</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_FAMILY</td>
<td>&quot;javax.faces.ViewRoot&quot;</td>
</tr>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.ViewRoot&quot;</td>
</tr>
<tr>
<td>public static final String UNIQUE_ID_PREFIX</td>
<td>&quot;j_id&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.html.HtmlColumn</th>
<th>javax.faces.component.html.HtmlColumn</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlColumn&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlCommandButton&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.html.HtmlCommandLink</th>
<th>javax.faces.component.html.HtmlCommandLink</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlCommandLink&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.html.HtmlDataTable</th>
<th>javax.faces.component.html.HtmlDataTable</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlDataTable&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.html.HtmlForm</th>
<th>javax.faces.component.html.HtmlForm</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlForm&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.component.html.HtmlGraphicImage</th>
<th>javax.faces.component.html.HtmlGraphicImage</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String COMPONENT_TYPE</td>
<td>&quot;javax.faces.HtmlGraphicImage&quot;</td>
</tr>
</tbody>
</table>
javax.faces.component.html.**HtmlInputHidden**
public static final String COMPONENT_TYPE "javax.faces.HtmlInputHidden"

javax.faces.component.html.**HtmlInputSecret**
public static final String COMPONENT_TYPE "javax.faces.HtmlInputSecret"

javax.faces.component.html.**HtmlInputText**
public static final String COMPONENT_TYPE "javax.faces.HtmlInputText"

javax.faces.component.html.**HtmlInputTextarea**
public static final String COMPONENT_TYPE "javax.faces.HtmlInputTextarea"

javax.faces.component.html.**HtmlMessage**
public static final String COMPONENT_TYPE "javax.faces.HtmlMessage"

javax.faces.component.html.**HtmlMessages**
public static final String COMPONENT_TYPE "javax.faces.HtmlMessages"

javax.faces.component.html.**HtmlOutputFormat**
public static final String COMPONENT_TYPE "javax.faces.HtmlOutputFormat"

javax.faces.component.html.**HtmlOutputLabel**
public static final String COMPONENT_TYPE "javax.faces.HtmlOutputLabel"

javax.faces.component.html.**HtmlOutputLink**
public static final String COMPONENT_TYPE "javax.faces.HtmlOutputLink"

javax.faces.component.html.**HtmlOutputText**
public static final String COMPONENT_TYPE "javax.faces.HtmlOutputText"

javax.faces.component.html.**HtmlPanelGrid**
public static final String COMPONENT_TYPE "javax.faces.HtmlPanelGrid"

javax.faces.component.html.**HtmlPanelGroup**
public static final String COMPONENT_TYPE "javax.faces.HtmlPanelGroup"
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Method Name</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.faces.component.html.HtmlSelectBooleanCheckbox</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectBooleanCheckbox&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectManyCheckbox</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectManyCheckbox&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectManyListbox</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectManyListbox&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectManyMenu</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectManyMenu&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectOneListbox</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectOneListbox&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectOneMenu</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectOneMenu&quot;</td>
</tr>
<tr>
<td>javax.faces.component.html.HtmlSelectOneRadio</td>
<td>COMPONENT_TYPE</td>
<td>String</td>
<td>&quot;javax.faces.HtmlSelectOneRadio&quot;</td>
</tr>
<tr>
<td>javax.faces.context.ExternalContext</td>
<td>BASIC_AUTH</td>
<td>String</td>
<td>&quot;BASIC&quot;</td>
</tr>
<tr>
<td>javax.faces.context.ExternalContext</td>
<td>CLIENT_CERT_AUTH</td>
<td>String</td>
<td>&quot;CLIENT_CERT&quot;</td>
</tr>
<tr>
<td>javax.faces.context.ExternalContext</td>
<td>DIGEST_AUTH</td>
<td>String</td>
<td>&quot;DIGEST&quot;</td>
</tr>
<tr>
<td>javax.faces.context.ExternalContext</td>
<td>FORM_AUTH</td>
<td>String</td>
<td>&quot;FORM&quot;</td>
</tr>
<tr>
<td>javax.faces.convert.BigDecimalConverter</td>
<td>CONVERTER_ID</td>
<td>String</td>
<td>&quot;javax.faces.BigDecimal&quot;</td>
</tr>
<tr>
<td>javax.faces.convert.BigDecimalConverter</td>
<td>DECIMAL_ID</td>
<td>String</td>
<td>&quot;javax.faces.converter.BigDecimalConverter.DECIMAL&quot;</td>
</tr>
<tr>
<td>javax.faces.convert.BigDecimalConverter</td>
<td>STRING_ID</td>
<td>String</td>
<td>&quot;javax.faces.converter.STRING&quot;</td>
</tr>
<tr>
<td>javax.faces.convert.BigIntegerConverter</td>
<td>BIGINTEGER_ID</td>
<td>String</td>
<td>&quot;javax.faces.converter.BigIntegerConverter.BIGINTEGER&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>CONVERTER_ID</td>
<td>&quot;javax.faces.&quot;</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
<td>---------------</td>
<td>----------------</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.BooleanConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>BOOLEAN_ID</th>
<th>&quot;javax.faces.converter.BooleanConverter.BOOLEAN&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>CONVERTER_ID</td>
<td>&quot;javax.faces.&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.converter.&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.ByteConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>BYTE_ID</th>
<th>&quot;javax.faces.converter.ByteConverter.BYTE&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>CONVERTER_ID</td>
<td>&quot;javax.faces.&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.converter.&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.CharacterConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>CHARACTER_ID</th>
<th>&quot;javax.faces.converter.CharacterConverter.CHARACTER&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>CONVERTER_ID</td>
<td>&quot;javax.faces.&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.converter.&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.DateTimeConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>CONVERTER_ID</th>
<th>&quot;javax.faces.&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>DATE_ID</td>
<td>&quot;javax.faces.converter.DateTimeConverter.DATE&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>DATETIME_ID</td>
<td>&quot;javax.faces.converter.DateTimeConverter.DATETIME&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.converter.&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>TIME_ID</td>
<td>&quot;javax.faces.converter.DateTimeConverter.TIME&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.DoubleConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>CONVERTER_ID</th>
<th>&quot;javax.faces.&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>DOUBLE_ID</td>
<td>&quot;javax.faces.converter.DoubleConverter.DOUBLE&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>STRING_ID</td>
<td>&quot;javax.faces.converter.&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.EnumConverter**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>CONVERTER_ID</th>
<th>&quot;javax.faces.&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>ENUM_ID</td>
<td>&quot;javax.faces.converter.EnumConverter.ENUM&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>ENUM_NO_CLASS_ID</td>
<td>&quot;javax.faces.converter.EnumConverter.ENUM_NO_CLASS&quot;</td>
</tr>
</tbody>
</table>

**javax.faces.convert.FloatConverter**

<p>| public static final | String | CONVERTER_ID | &quot;javax.faces.&quot; |</p>
<table>
<thead>
<tr>
<th>Class</th>
<th>Field</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>javax.faces.convert.IntegerConverter</code></td>
<td><code>CONVERTER_ID</code></td>
<td><code>javax.faces.converter.IntegerConverter</code></td>
</tr>
<tr>
<td><code>javax.faces.convert.LongConverter</code></td>
<td><code>CONVERTER_ID</code></td>
<td><code>javax.faces.converter.LongConverter</code></td>
</tr>
<tr>
<td><code>javax.faces.convert.NumberConverter</code></td>
<td><code>CONVERTER_ID</code></td>
<td><code>javax.faces.converter.NumberConverter</code></td>
</tr>
<tr>
<td><code>javax.faces.convert.ShortConverter</code></td>
<td><code>CONVERTER_ID</code></td>
<td><code>javax.faces.converter.ShortConverter</code></td>
</tr>
<tr>
<td><code>javax.faces.lifecycle.LifecycleFactory</code></td>
<td><code>DEFAULT_LIFECYCLE</code></td>
<td>&quot;DEFAULT&quot;</td>
</tr>
<tr>
<td><code>javax.faces.render.RenderKitFactory</code></td>
<td><code>HTML_BASIC_RENDER_KIT</code></td>
<td>&quot;HTML_BASIC&quot;</td>
</tr>
<tr>
<td><code>javax.faces.render.ResponseStateManager</code></td>
<td><code>RENDER_KIT_ID_PARAM</code></td>
<td>&quot;javax.faces.RenderKitId&quot;</td>
</tr>
<tr>
<td><code>javax.faces.render.ResponseStateManager</code></td>
<td><code>VIEW_STATE_PARAM</code></td>
<td>&quot;javax.faces ViewState&quot;</td>
</tr>
<tr>
<td>javax.faces.validator</td>
<td>DoubleRangeValidator</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>public static final String MAXIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.DoubleRangeValidator.MAXIMUM&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String MINIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.DoubleRangeValidator.MINIMUM&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String NOT_IN_RANGE_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.DoubleRangeValidator.NOT_IN_RANGE&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String TYPE_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.DoubleRangeValidator.TYPE&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String VALIDATOR_ID</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.validator</th>
<th>LengthValidator</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String MAXIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LengthValidator.MAXIMUM&quot;</td>
</tr>
<tr>
<td>public static final String MINIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LengthValidator.MINIMUM&quot;</td>
</tr>
<tr>
<td>public static final String VALIDATOR_ID</td>
<td>&quot;javax.faces.Length&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.validator</th>
<th>LongRangeValidator</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String MAXIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LongRangeValidator.MAXIMUM&quot;</td>
</tr>
<tr>
<td>public static final String MINIMUM_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LongRangeValidator.MINIMUM&quot;</td>
</tr>
<tr>
<td>public static final String NOT_IN_RANGE_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LongRangeValidator.NOT_IN_RANGE&quot;</td>
</tr>
<tr>
<td>public static final String TYPE_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.LongRangeValidator.TYPE&quot;</td>
</tr>
<tr>
<td>public static final String VALIDATOR_ID</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.validator</th>
<th>Validator</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String NOT_IN_RANGE_MESSAGE_ID</td>
<td>&quot;javax.faces.validator.NOT_IN_RANGE&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.webapp</th>
<th>FacesServlet</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String CONFIG_FILES_ATTR</td>
<td>&quot;javax.faces.CONFIG_FILES&quot;</td>
</tr>
<tr>
<td>public static final String LIFECYCLE_ID_ATTR</td>
<td>&quot;javax.faces.LIFECYCLE_ID&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.faces.webapp</th>
<th>UIComponentClassicTagBase</th>
</tr>
</thead>
<tbody>
<tr>
<td>protected static final String UNIQUE_ID_PREFIX</td>
<td>&quot;j_id_&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.jms.*</th>
<th>DeliveryMode</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int NON_PERSISTENT</td>
<td>1</td>
</tr>
<tr>
<td>public static final int PERSISTENT</td>
<td>2</td>
</tr>
</tbody>
</table>
javax.jms.

**Message**

- public static final int DEFAULT_DELIVERY_MODE = 2
- public static final int DEFAULT_PRIORITY = 4
- public static final long DEFAULT_TIME_TO_LIVE = 0L

javax.jms.

**Session**

- public static final int AUTO_ACKNOWLEDGE = 1
- public static final int CLIENT_ACKNOWLEDGE = 2
- public static final int DUPS_OK_ACKNOWLEDGE = 3
- public static final int SESSION_TRANSACTED = 0

javax.mail.*

javax.mail.

**Folder**

- public static final int HOLDS_FOLDERS = 2
- public static final int HOLDS_MESSAGES = 1
- public static final int READ_ONLY = 1
- public static final int READ_WRITE = 2

javax.mail.

**Part**

- public static final String ATTACHMENT = "attachment"
- public static final String INLINE = "inline"

javax.mail.

**UIDFolder**

- public static final long LASTUID = -1L

javax.mail.event.

**ConnectionEvent**

- public static final int CLOSED = 3
- public static final int DISCONNECTED = 2
- public static final int OPENED = 1

javax.mail.event.

**FolderEvent**

- public static final int CREATED = 1
- public static final int DELETED = 2
- public static final int RENAMED = 3
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int ENVELOPE_CHANGED 2</td>
<td>public static final int ADDED 1</td>
<td>public static final int ALERT 1</td>
<td>public static final int MESSAGE_DELIVERED 1</td>
<td>public static final String MIME &quot;()&lt;&gt;@;:&quot;:&quot;\t []/?=&quot;</td>
<td>public static final int ATOM -1</td>
<td>public static final int ALL -1</td>
<td>public static final int EQ 3</td>
</tr>
<tr>
<td>public static final int FLAGS_CHANGED 1</td>
<td>public static final int REMOVED 2</td>
<td>public static final int NOTICE 2</td>
<td>public static final int MESSAGE_NOT_DELIVERED 2</td>
<td>public static final String RFC822 &quot;()&lt;&gt;@;:&quot;:&quot;\t .[]&quot;</td>
<td>public static final int COMMENT -3</td>
<td></td>
<td>public static final int GE 6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>public static final int MESSAGE_PARTIALLY_DELIVERED 3</td>
<td></td>
<td>public static final int EOF -4</td>
<td></td>
<td>public static final int GT 5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>public static final int QUOTEDSTRING -2</td>
<td></td>
<td>public static final int LE 1</td>
</tr>
<tr>
<td>javax.persistence.*</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>public static final int <strong>LT</strong></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>public static final int <strong>NE</strong></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.persistence.Persistence</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String <strong>PERSISTENCE_PROVIDER</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.resource.*</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.resource.cci.InteractionSpec</td>
</tr>
<tr>
<td>public static final int <strong>SYNC_RECEIVE</strong></td>
</tr>
<tr>
<td>public static final int <strong>SYNC_SEND</strong></td>
</tr>
<tr>
<td>public static final int <strong>SYNC_SEND_RECEIVE</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.resource.spi.ConnectionEvent</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int <strong>CONNECTION_CLOSED</strong></td>
</tr>
<tr>
<td>public static final int <strong>CONNECTION_ERROR_OCCURRED</strong></td>
</tr>
<tr>
<td>public static final int <strong>LOCAL_TRANSACTION_COMMITTED</strong></td>
</tr>
<tr>
<td>public static final int <strong>LOCAL_TRANSACTION_ROLLED_BACK</strong></td>
</tr>
<tr>
<td>public static final int <strong>LOCAL_TRANSACTION_STARTED</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.resource.spi.work.WorkEvent</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int <strong>WORK_ACCEPTED</strong></td>
</tr>
<tr>
<td>public static final int <strong>WORK_COMPLETED</strong></td>
</tr>
<tr>
<td>public static final int <strong>WORK_REJECTED</strong></td>
</tr>
<tr>
<td>public static final int <strong>WORK_STARTED</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.resource.spi.work.WorkException</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String <strong>INTERNAL</strong></td>
</tr>
<tr>
<td>public static final String <strong>START_TIMED_OUT</strong></td>
</tr>
<tr>
<td>public static final String <strong>TX_CONCURRENT_WORK_DISALLOWED</strong></td>
</tr>
<tr>
<td>public static final String <strong>TX_RECREATE_FAILED</strong></td>
</tr>
<tr>
<td>public static final String <strong>UNDEFINED</strong></td>
</tr>
</tbody>
</table>
javax.resource.spi.work

<table>
<thead>
<tr>
<th>WorkManager</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IMMEDIATE</td>
<td>0L</td>
</tr>
<tr>
<td>INDEFINITE</td>
<td>9223372036854775807L</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>-1L</td>
</tr>
</tbody>
</table>

javax.servlet.*

<table>
<thead>
<tr>
<th>HttpServletRequest</th>
</tr>
</thead>
<tbody>
<tr>
<td>BASIC_AUTH</td>
</tr>
<tr>
<td>CLIENT_CERT_AUTH</td>
</tr>
<tr>
<td>DIGEST_AUTH</td>
</tr>
<tr>
<td>FORM_AUTH</td>
</tr>
</tbody>
</table>

javax.servlet.http

<table>
<thead>
<tr>
<th>HttpServletResponse</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC_ACCEPTED</td>
</tr>
<tr>
<td>SC_BAD_GATEWAY</td>
</tr>
<tr>
<td>SC_BAD_REQUEST</td>
</tr>
<tr>
<td>SC_CONFLICT</td>
</tr>
<tr>
<td>SC_CONTINUE</td>
</tr>
<tr>
<td>SC_CREATED</td>
</tr>
<tr>
<td>SC_EXPECTATION_FAILED</td>
</tr>
<tr>
<td>SC_FORBIDDEN</td>
</tr>
<tr>
<td>SC_FOUND</td>
</tr>
<tr>
<td>SC_GATEWAY_TIMEOUT</td>
</tr>
<tr>
<td>SC_GONE</td>
</tr>
<tr>
<td>SC_HTTP_VERSION_NOT_SUPPORTED</td>
</tr>
<tr>
<td>SC_INTERNAL_SERVER_ERROR</td>
</tr>
<tr>
<td>SC_LENGTH_REQUIRED</td>
</tr>
<tr>
<td>SC_METHOD_NOT_ALLOWED</td>
</tr>
<tr>
<td>SC_MOVED_PERMANENTLY</td>
</tr>
<tr>
<td>SC_MOVED_TEMPORARILY</td>
</tr>
<tr>
<td>SC_MULTIPLE_CHOICES</td>
</tr>
<tr>
<td>SC_NO_CONTENT</td>
</tr>
<tr>
<td>SC_NON_AUTHORITATIVE_INFORMATION</td>
</tr>
<tr>
<td>SC_NOT_ACCEPTABLE</td>
</tr>
<tr>
<td>SC_NOT_FOUND</td>
</tr>
<tr>
<td>Status Code</td>
</tr>
<tr>
<td>-------------</td>
</tr>
<tr>
<td>501</td>
</tr>
<tr>
<td>304</td>
</tr>
<tr>
<td>200</td>
</tr>
<tr>
<td>206</td>
</tr>
<tr>
<td>402</td>
</tr>
<tr>
<td>412</td>
</tr>
<tr>
<td>407</td>
</tr>
<tr>
<td>413</td>
</tr>
<tr>
<td>408</td>
</tr>
<tr>
<td>414</td>
</tr>
<tr>
<td>416</td>
</tr>
<tr>
<td>205</td>
</tr>
<tr>
<td>303</td>
</tr>
<tr>
<td>503</td>
</tr>
<tr>
<td>101</td>
</tr>
<tr>
<td>307</td>
</tr>
<tr>
<td>401</td>
</tr>
<tr>
<td>415</td>
</tr>
<tr>
<td>305</td>
</tr>
</tbody>
</table>

### javax.servlet.jsp.JspWriter

<table>
<thead>
<tr>
<th>Buffer Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_BUFFER</td>
<td>-1</td>
</tr>
<tr>
<td>NO_BUFFER</td>
<td>0</td>
</tr>
<tr>
<td>UNBOUNDED_BUFFER</td>
<td>-2</td>
</tr>
</tbody>
</table>

### javax.servlet.jsp.PageContext

<table>
<thead>
<tr>
<th>Scope Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>APPLICATION</td>
<td>javax.servlet.jsp.jspApplicationContext</td>
</tr>
<tr>
<td>CONFIG</td>
<td>javax.servlet.jsp.jspConfig</td>
</tr>
<tr>
<td>EXCEPTION</td>
<td>javax.servlet.jsp.jspException</td>
</tr>
<tr>
<td>OUT</td>
<td>javax.servlet.jsp.jspOut</td>
</tr>
<tr>
<td>PAGE</td>
<td>javax.servlet.jsp.jspPage</td>
</tr>
<tr>
<td>REQUEST</td>
<td>javax.servlet.jsp.jspRequest</td>
</tr>
<tr>
<td>RESPONSE</td>
<td>javax.servlet.jsp.jspResponse</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.BodyTag</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int EVAL_BODY_BUFFERED 2</td>
</tr>
<tr>
<td>public static final int EVAL_BODY_TAG 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.IterationTag</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int EVAL_BODY_AGAIN 2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int EVAL_BODY_INCLUDE 1</td>
</tr>
<tr>
<td>public static final int EVAL_PAGE 6</td>
</tr>
<tr>
<td>public static final int SKIP_BODY 0</td>
</tr>
<tr>
<td>public static final int SKIP_PAGE 5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.TagAttributeInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String ID &quot;id&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.TagInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String BODY_CONTENT_EMPTY &quot;empty&quot;</td>
</tr>
<tr>
<td>public static final String BODY_CONTENT_JSP &quot;JSP&quot;</td>
</tr>
<tr>
<td>public static final String BODY_CONTENT_SCRIPTLESS &quot;scriptless&quot;</td>
</tr>
<tr>
<td>public static final String BODY_CONTENT_TAG_DEPENDENT &quot;tagdependent&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.servlet.jsp.tagext.VariableInfo</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int AT_BEGIN 1</td>
</tr>
<tr>
<td>public static final int AT_END 2</td>
</tr>
<tr>
<td>public static final int NESTED 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.transaction.*</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>javax.transaction.Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int STATUS_ACTIVE 0</td>
</tr>
<tr>
<td>public static final int STATUS_COMMITTED 3</td>
</tr>
<tr>
<td>Constant</td>
</tr>
<tr>
<td>-------------------------------</td>
</tr>
<tr>
<td>STATUS_COMMITTING</td>
</tr>
<tr>
<td>STATUS_MARKED_ROLLBACK</td>
</tr>
<tr>
<td>STATUS_NO_TRANSACTION</td>
</tr>
<tr>
<td>STATUS_PREPARED</td>
</tr>
<tr>
<td>STATUS_PREPARING</td>
</tr>
<tr>
<td>STATUS_ROLLED_BACK</td>
</tr>
<tr>
<td>STATUS_ROLLING_BACK</td>
</tr>
<tr>
<td>STATUS_UNKNOWN</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class</th>
<th>Constant</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.transaction.xa</td>
<td>XAException</td>
<td></td>
</tr>
<tr>
<td>XA_HEURCOM</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>XA_HEURHAZ</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>XA_HEURMIX</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>XA_HEURRB</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>XA_NOMIGRATE</td>
<td>9</td>
<td></td>
</tr>
<tr>
<td>XA_RBBASE</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>XA_RBCOMMFAIL</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>XA_RBDEADLOCK</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>XA_RBEND</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>XA_RBINTEGRITY</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>XA_RBOTHER</td>
<td>104</td>
<td></td>
</tr>
<tr>
<td>XA_RBPOTO</td>
<td>105</td>
<td></td>
</tr>
<tr>
<td>XA_RBROLLBACK</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>XA_RBTIMEOUT</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>XA_RTRANSIENT</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>XA_RDONLY</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>XA_RETRY</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>XAER_ASYNC</td>
<td>-2</td>
<td></td>
</tr>
<tr>
<td>XAER_DUPID</td>
<td>-8</td>
<td></td>
</tr>
<tr>
<td>XAER_INVAL</td>
<td>-5</td>
<td></td>
</tr>
<tr>
<td>XAER_NOTA</td>
<td>-4</td>
<td></td>
</tr>
<tr>
<td>XAER_OUTSIDE</td>
<td>-9</td>
<td></td>
</tr>
<tr>
<td>XAER_PROTO</td>
<td>-6</td>
<td></td>
</tr>
<tr>
<td>XAER_RMERR</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>XAER_RMFAL</td>
<td>-7</td>
<td></td>
</tr>
<tr>
<td>javax.transaction.xa.Xid</td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------------------------</td>
<td>-----------------</td>
<td>---</td>
</tr>
<tr>
<td>public static final int MAXBOQUALSIZE</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>public static final int MAXGTRIDSIZE</td>
<td>64</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.xml.*</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>javax.xml.bind.JAXBContext</td>
<td></td>
<td></td>
</tr>
<tr>
<td>public static final String JAXB_CONTEXT_FACTORY</td>
<td>&quot;javax.xml.bind.context.factory&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.xml.bind.Marshaller</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String JAXB_ENCODING</td>
<td></td>
<td></td>
</tr>
<tr>
<td>public static final String JAXB_FORMATTED_OUTPUT</td>
<td>&quot;jaxb.formatted.output&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String JAXB_FRAGMENT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>public static final String JAXB_NO_NAMESPACE_SCHEMA_LOCATION</td>
<td>&quot;jaxb.noNamespaceSchemaLocation&quot;</td>
<td></td>
</tr>
<tr>
<td>public static final String JAXB_SCHEMA_LOCATION</td>
<td>&quot;jaxb.schemaLocation&quot;</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.xml.bind.ValidationEvent</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final int ERROR</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>public static final int FATAL_ERROR</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>public static final int WARNING</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>javax.xml.registry.FindQualifier</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String AND_ALL_KEYS</td>
<td>&quot;andAllKeys&quot;</td>
<td></td>
</tr>
</tbody>
</table>
### javax.xml.registry.JAXRResponse

| public static final int STATUS_FAILURE | 2 |
| public static final int STATUS_SUCCESS | 0 |
| public static final int STATUS_UNAVAILABLE | 3 |
| public static final int STATUS_WARNING | 1 |

### javax.xml.registry.LifeCycleManager

<p>| public static final String ASSOCIATION | &quot;Association&quot; |
| public static final String AUDITABLE_EVENT | &quot;AuditableEvent&quot; |
| public static final String CLASSIFICATION | &quot;Classification&quot; |
| public static final String CLASSIFICATION_SCHEME | &quot;ClassificationScheme&quot; |
| public static final String CONCEPT | &quot;Concept&quot; |
| public static final String EMAIL_ADDRESS | &quot;EmailAddress&quot; |
| public static final String EXTERNAL_IDENTIFIER | &quot;ExternalIdentifier&quot; |
| public static final String EXTERNAL_LINK | &quot;ExternalLink&quot; |
| public static final String EXTRINSIC_OBJECT | &quot;ExtrinsicObject&quot; |
| public static final String INTERNATIONAL_STRING | &quot;InternationalString&quot; |
| public static final String KEY | &quot;Key&quot; |
| public static final String LOCALIZED_STRING | &quot;LocalizedString&quot; |
| public static final String ORGANIZATION | &quot;Organization&quot; |
| public static final String PERSON_NAME | &quot;PersonName&quot; |
| public static final String POSTAL_ADDRESS | &quot;PostalAddress&quot; |
| public static final String REGISTRY_ENTRY | &quot;RegistryEntry&quot; |
| public static final String REGISTRY_PACKAGE | &quot;RegistryPackage&quot; |
| public static final String SERVICE | &quot;Service&quot; |</p>
<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>SERVICE_BINDING</th>
<th>&quot;ServiceBinding&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>String</td>
<td>SLOT</td>
<td>&quot;Slot&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>SPECIFICATION_LINK</td>
<td>&quot;SpecificationLink&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>TELEPHONE_NUMBER</td>
<td>&quot;TelephoneNumber&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>USER</td>
<td>&quot;User&quot;</td>
</tr>
<tr>
<td>public static final</td>
<td>String</td>
<td>VERSIONABLE</td>
<td>&quot;Versionable&quot;</td>
</tr>
</tbody>
</table>

**javax.xml.registry.Query**

<table>
<thead>
<tr>
<th>public static final</th>
<th>int</th>
<th>QUERY_TYPE_EBXML_FILTER_QUERY</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>int</td>
<td>QUERY_TYPE_SQL</td>
<td>0</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>QUERY_TYPE_XQUERY</td>
<td>1</td>
</tr>
</tbody>
</table>

**javax.xml.registry.infomodel.AuditableEvent**

<table>
<thead>
<tr>
<th>public static final</th>
<th>int</th>
<th>EVENT_TYPE_CREATED</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>int</td>
<td>EVENT_TYPE_DELETED</td>
<td>1</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>EVENT_TYPE_DEPRECATED</td>
<td>2</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>EVENT_TYPE_UNDEPRECATED</td>
<td>5</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>EVENT_TYPE_UPDATED</td>
<td>3</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>EVENT_TYPE_VERSIONED</td>
<td>4</td>
</tr>
</tbody>
</table>

**javax.xml.registry.infomodel.ClassificationScheme**

<table>
<thead>
<tr>
<th>public static final</th>
<th>int</th>
<th>VALUE_TYPE_EMBEDDED_PATH</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>int</td>
<td>VALUE_TYPE_NON_UNIQUE</td>
<td>2</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>VALUE_TYPE_UNIQUE</td>
<td>0</td>
</tr>
</tbody>
</table>

**javax.xml.registry.infomodel.LocalizedString**

<table>
<thead>
<tr>
<th>public static final</th>
<th>String</th>
<th>DEFAULT_CHARSET_NAME</th>
<th>&quot;UTF-8&quot;</th>
</tr>
</thead>
</table>

**javax.xml.registry.infomodel.RegistryEntry**

<table>
<thead>
<tr>
<th>public static final</th>
<th>int</th>
<th>STABILITY_DYNAMIC</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final</td>
<td>int</td>
<td>STABILITY_DYNAMIC_COMPATIBLE</td>
<td>1</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>STABILITY_STATIC</td>
<td>2</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>STATUS_APPROVED</td>
<td>1</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>STATUS_DEPRECATED</td>
<td>2</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>STATUS_SUBMITTED</td>
<td>0</td>
</tr>
<tr>
<td>public static final</td>
<td>int</td>
<td>STATUS_WITHDRAWN</td>
<td>3</td>
</tr>
</tbody>
</table>
### javax.xml.registry.infomodel

<table>
<thead>
<tr>
<th>Slot</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADDRESS_LINES_SLOT</td>
<td>&quot;addressLines&quot;</td>
</tr>
<tr>
<td>AUTHORIZED_NAME_SLOT</td>
<td>&quot;authorizedName&quot;</td>
</tr>
<tr>
<td>OPERATOR_SLOT</td>
<td>&quot;operator&quot;</td>
</tr>
<tr>
<td>SORT_CODE_SLOT</td>
<td>&quot;sortCode&quot;</td>
</tr>
</tbody>
</table>

### javax.xml.rpc

<table>
<thead>
<tr>
<th>Call</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENCODINGSTYLE_URI_PROPERTY</td>
<td>&quot;javax.xml.rpc.encodingstyle.namespace.uri&quot;</td>
</tr>
<tr>
<td>OPERATION_STYLE_PROPERTY</td>
<td>&quot;javax.xml.rpc.soap.operation.style&quot;</td>
</tr>
<tr>
<td>PASSWORD_PROPERTY</td>
<td>&quot;javax.xml.rpc.security.auth.password&quot;</td>
</tr>
<tr>
<td>SESSION_MAINTAIN_PROPERTY</td>
<td>&quot;javax.xml.rpc.session.maintain&quot;</td>
</tr>
<tr>
<td>SOAPACTION_URI_PROPERTY</td>
<td>&quot;javax.xml.rpc.soap.http.soapaction.uri&quot;</td>
</tr>
<tr>
<td>SOAPACTION_USE_PROPERTY</td>
<td>&quot;javax.xml.rpc.soap.http.soapaction.use&quot;</td>
</tr>
<tr>
<td>USERNAME_PROPERTY</td>
<td>&quot;javax.xml.rpc.security.auth.username&quot;</td>
</tr>
</tbody>
</table>

### javax.xml.rpc.NAMESPACECONSTANTS

<table>
<thead>
<tr>
<th>NamespaceConstants</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSPREFIX_SCHEMA_XSD</td>
<td>&quot;<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>&quot;</td>
</tr>
<tr>
<td>NSPREFIX_SCHEMA_XSI</td>
<td>&quot;<a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a>&quot;</td>
</tr>
<tr>
<td>NSPREFIX_SOAP_ENCODING</td>
<td>&quot;<a href="http://schemas.xmlsoap.org/soap/encoding/">http://schemas.xmlsoap.org/soap/encoding/</a>&quot;</td>
</tr>
<tr>
<td>NSPREFIX_SOAP_ENVELOPE</td>
<td>&quot;<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>&quot;</td>
</tr>
<tr>
<td>SERVICEFACTORY_PROPERTY</td>
<td>&quot;javax.xml.rpc.ServiceFactory&quot;</td>
</tr>
</tbody>
</table>

### javax.xml.rpc.STUB

<table>
<thead>
<tr>
<th>Stub</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDPOINT_ADDRESS_PROPERTY</td>
<td>&quot;javax.xml.rpc.service.endpoint.address&quot;</td>
</tr>
<tr>
<td>PASSWORD_PROPERTY</td>
<td>&quot;javax.xml.rpc.security.auth.password&quot;</td>
</tr>
<tr>
<td>SESSION_MAINTAIN_PROPERTY</td>
<td>&quot;javax.xml.rpc.session.maintain&quot;</td>
</tr>
<tr>
<td>USERNAME_PROPERTY</td>
<td>&quot;javax.xml.rpc.security.auth.username&quot;</td>
</tr>
</tbody>
</table>

### javax.xml.soap.SOAPCONSTANTS

<table>
<thead>
<tr>
<th>SOAPConstants</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>String</td>
<td>DEFAULT_SOAP_PROTOCOL</td>
</tr>
<tr>
<td>String</td>
<td>DYNAMIC_SOAP_PROTOCOL</td>
</tr>
<tr>
<td>String</td>
<td>SOAP_1_1_CONTENT_TYPE</td>
</tr>
<tr>
<td>String</td>
<td>SOAP_1_1_PROTOCOL</td>
</tr>
<tr>
<td>String</td>
<td>SOAP_1_2_CONTENT_TYPE</td>
</tr>
<tr>
<td>String</td>
<td>SOAP_1_2_PROTOCOL</td>
</tr>
<tr>
<td>String</td>
<td>SOAP_ENV_PREFIX</td>
</tr>
<tr>
<td>String</td>
<td>URI_NS_SOAP_1_1_ENVELOPE</td>
</tr>
<tr>
<td>String</td>
<td>URI_NS_SOAP_1_2_ENCODING</td>
</tr>
<tr>
<td>String</td>
<td>URI_NS_SOAP_1_2_ENVELOPE</td>
</tr>
<tr>
<td>String</td>
<td>URI_NS_SOAP_ENCODING</td>
</tr>
<tr>
<td>String</td>
<td>URI_NS_SOAP_ENVELOPE</td>
</tr>
<tr>
<td>String</td>
<td>URI_SOAP_1_2_ROLE_NEXT</td>
</tr>
<tr>
<td>String</td>
<td>URI_SOAP_1_2_ROLE_NONE</td>
</tr>
<tr>
<td>String</td>
<td>URI_SOAP_1_2_ROLE_ULTIMATE_RECEIVER</td>
</tr>
<tr>
<td>String</td>
<td>URI_SOAP.Actor_NEXT</td>
</tr>
</tbody>
</table>

### javax.xml.soap.SOAPMessage

- **public static final String CHARACTER_SET_ENCODING**: "javax.xml.soap.character-set-encoding"
- **public static final String WRITE_XML_DECLARATION**: "javax.xml.soap.write-xml-declaration"

### javax.xml.stream.XMLInputFactory

- **public static final String ALLOCATOR**: "javax.xml.stream.allocators.allocators"
- **public static final String IS_COALEScing**: "javax.xml.stream.isCoalescing"
- **public static final String IS_NAMESPACE_AWARE**: "javax.xml.stream.isNamespaceAware"
- **public static final String IS_REPLACING_ENTITY_REFERENCES**: "javax.xml.stream.isReplacingEntityReferences"
- **public static final String IS_SUPPORTING_EXTERNAL_ENTITIES**: "javax.xml.stream.isSupportingExternalEntities"
- **public static final String IS_VALIDATING**: "javax.xml.stream.isValidating"
- **public static final String REPORTER**: "javax.xml.stream.reporter"
- **public static final String RESOLVER**: "javax.xml.stream.resolver"
- **public static final String SUPPORT_DTD**: "javax.xml.stream.supportSupportDTD"
public static final String IS_REPAIRING_NAMESPACES = "javax.xml.stream.isRepairingNamespaces"

javax.xml.stream.XMLStreamConstants

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIBUTE</td>
<td>10</td>
</tr>
<tr>
<td>CDATA</td>
<td>12</td>
</tr>
<tr>
<td>CHARACTERS</td>
<td>4</td>
</tr>
<tr>
<td>COMMENT</td>
<td>5</td>
</tr>
<tr>
<td>DTD</td>
<td>11</td>
</tr>
<tr>
<td>END_DOCUMENT</td>
<td>8</td>
</tr>
<tr>
<td>END_ELEMENT</td>
<td>2</td>
</tr>
<tr>
<td>ENTITY_DECLARATION</td>
<td>15</td>
</tr>
<tr>
<td>ENTITY_REFERENCE</td>
<td>9</td>
</tr>
<tr>
<td>NAMESPACE</td>
<td>13</td>
</tr>
<tr>
<td>NOTATION_DECLARATION</td>
<td>14</td>
</tr>
<tr>
<td>PROCESSING_INSTRUCTION</td>
<td>3</td>
</tr>
<tr>
<td>SPACE</td>
<td>6</td>
</tr>
<tr>
<td>START_DOCUMENT</td>
<td>7</td>
</tr>
<tr>
<td>START_ELEMENT</td>
<td>1</td>
</tr>
</tbody>
</table>

javax.xml.ws.BindingProvider

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENDPOINT_ADDRESS_PROPERTY</td>
<td>&quot;javax.xml.ws.service.endpoint.address&quot;</td>
</tr>
<tr>
<td>PASSWORDPROPERTY</td>
<td>&quot;javax.xml.ws.security.auth.password&quot;</td>
</tr>
<tr>
<td>SESSION_MAINTAIN_PROPERTY</td>
<td>&quot;javax.xml.ws.session.maintain&quot;</td>
</tr>
<tr>
<td>SOAPACTION_URIPROPERTY</td>
<td>&quot;javax.xml.ws.soap.http.soapaction.uri&quot;</td>
</tr>
<tr>
<td>SOAPACTION_USEPROPERTY</td>
<td>&quot;javax.xml.ws.soap.http.soapaction.use&quot;</td>
</tr>
<tr>
<td>USERNAMEPROPERTY</td>
<td>&quot;javax.xml.ws.security.auth.username&quot;</td>
</tr>
</tbody>
</table>

javax.xml.ws.Endpoint

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL_PORT</td>
<td>&quot;javax.xml.ws.wsdl.port&quot;</td>
</tr>
<tr>
<td>WSDL_SERVICE</td>
<td>&quot;javax.xml.ws.wsdl.service&quot;</td>
</tr>
</tbody>
</table>

javax.xml.ws.handler.MessageContext

<table>
<thead>
<tr>
<th>Constant Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP_REQUEST_HEADERS</td>
<td>&quot;javax.xml.ws.http.request.headers&quot;</td>
</tr>
<tr>
<td>HTTP_REQUEST_METHOD</td>
<td>&quot;javax.xml.ws.http.request.method&quot;</td>
</tr>
<tr>
<td>HTTP_RESPONSE_CODE</td>
<td>&quot;javax.xml.ws.http.response.code&quot;</td>
</tr>
<tr>
<td>HTTP_RESPONSE_HEADERS</td>
<td>&quot;javax.xml.ws.http.response.headers&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>javax.xml.ws.http.HTTPBinding</strong></td>
<td></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>public static final String HTTP_BINDING</td>
<td>&quot;<a href="http://www.w3.org/2004/08/wsd/htp">http://www.w3.org/2004/08/wsd/htp</a>&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>javax.xml.ws.soap.SOAPBinding</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String SOAP11HTTP_BINDING</td>
<td>&quot;<a href="http://schemas.xmlsoap.org/soap/http">http://schemas.xmlsoap.org/soap/http</a>&quot;</td>
</tr>
<tr>
<td>public static final String SOAP11HTTP_MTOM_BINDING</td>
<td>&quot;<a href="http://schemas.xmlsoap.org/soap/http">http://schemas.xmlsoap.org/soap/http</a>?&quot;</td>
</tr>
<tr>
<td>public static final String SOAP12HTTP_BINDING</td>
<td>&quot;<a href="http://www.w3.org/2003/05/soap/bindings/HTTP/">http://www.w3.org/2003/05/soap/bindings/HTTP/</a>&quot;</td>
</tr>
<tr>
<td>public static final String SOAP12HTTP_MTOM_BINDING</td>
<td>&quot;<a href="http://www.w3.org/2003/05/soap/bindings/HTTP/">http://www.w3.org/2003/05/soap/bindings/HTTP/</a>?&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>javax.xml.ws.spi.Provider</strong></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>public static final String JAXWSPROVIDER_PROPERTY</td>
<td>&quot;javax.xml.ws.spi.Provider&quot;</td>
</tr>
</tbody>
</table>
**Package javax.enterprise.deploy.spi.exceptions**

Provides J2EE Product Vendor deployment exception implementation classes.

See: [Description](#)

### Exception Summary

<table>
<thead>
<tr>
<th>Exception Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BeanNotFoundException</strong></td>
<td>This exception is to report that the bean is not a child of the parent bean.</td>
</tr>
<tr>
<td><strong>ClientExecuteException</strong></td>
<td>This exception reports errors in setting up an application client for execution.</td>
</tr>
<tr>
<td><strong>ConfigurationException</strong></td>
<td>This exception reports errors in generating a configuration bean.</td>
</tr>
<tr>
<td><strong>DConfigBeanVersionUnsupportedException</strong></td>
<td>This exception is to report that there is no support for the DConfigBean version requested.</td>
</tr>
<tr>
<td><strong>DeploymentManagerCreationException</strong></td>
<td>This exception is to report problems in returning a DeploymentManager object cause by such things as server down, unable to authenticate and the like.</td>
</tr>
<tr>
<td><strong>InvalidModuleException</strong></td>
<td>This exception is to report an invalid J2EE deployment module</td>
</tr>
<tr>
<td>Exception</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>OperationUnsupportedException</td>
<td>This exception is to report that the method called is not supported by this implementation.</td>
</tr>
<tr>
<td>TargetException</td>
<td>This exception is to report bad target designators.</td>
</tr>
</tbody>
</table>
Package javax.enterprise.deploy.spi.exceptions
Description
Provides J2EE Product Vendor deployment exception implementation classes.
Package Specification

- [JSR 88, J2EE Application Deployment](#)
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- [J2EE Tools](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package
javax.enterprise.deploy.spi.exceptions

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.enterprise.deploy.spi.exceptions.**BeanNotFoundException**
      - javax.enterprise.deploy.spi.exceptions.**ClientExecuteException**
      - javax.enterprise.deploy.spi.exceptions.**ConfigurationException**
      - javax.enterprise.deploy.spi.exceptions.**DConfigBeanVersionUnsupportedException**
      - javax.enterprise.deploy.spi.exceptions.**DeploymentManagerCreationException**
      - javax.enterprise.deploy.spi.exceptions.**InvalidModuleException**
      - javax.enterprise.deploy.spi.exceptions.**OperationUnsupportedException**
      - javax.enterprise.deploy.spi.exceptions.**TargetException**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.faces.convert`

Contains classes and interfaces defining converters.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Converter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Converter</strong></td>
<td>Converter is an interface describing a Java class that can perform Object-to-String and String-to-Object conversions between model data objects and a String representation of those objects that is suitable for rendering.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BigDecimalConverter</strong></td>
<td>Converter implementation for <code>java.math.BigDecimal</code> values.</td>
</tr>
<tr>
<td><strong>BigIntegerConverter</strong></td>
<td>Converter implementation for <code>java.math.BigInteger</code> values.</td>
</tr>
<tr>
<td><strong>BooleanConverter</strong></td>
<td>Converter implementation for <code>java.lang.Boolean</code> (and boolean primitive) values.</td>
</tr>
<tr>
<td><strong>ByteConverter</strong></td>
<td>Converter implementation for <code>java.lang.Byte</code> (and byte primitive) values.</td>
</tr>
<tr>
<td><strong>CharacterConverter</strong></td>
<td>Converter implementation for <code>java.lang.Character</code> (and char primitive) values.</td>
</tr>
<tr>
<td><strong>DateTimeConverter</strong></td>
<td>Converter implementation for <code>java.util.Date</code> values.</td>
</tr>
<tr>
<td><strong>DoubleConverter</strong></td>
<td>Converter implementation for <code>java.lang.Double</code> (and double primitive) values.</td>
</tr>
<tr>
<td><strong>EnumConverter</strong></td>
<td>Converter implementation for <code>java.lang.Enum</code> (and enum primitive) values.</td>
</tr>
<tr>
<td><strong>FloatConverter</strong></td>
<td>Converter implementation for <code>java.lang.Float</code> (and float primitive) values.</td>
</tr>
<tr>
<td>Converter</td>
<td>Implementation for Java values.</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>IntegerConverter</td>
<td>Converter for <code>java.lang.Integer</code> (and int primitive) values.</td>
</tr>
<tr>
<td>LongConverter</td>
<td>Converter for <code>java.lang.Long</code> (and long primitive) values.</td>
</tr>
<tr>
<td>NumberConverter</td>
<td>Converter for <code>java.lang.Number</code> values.</td>
</tr>
<tr>
<td>ShortConverter</td>
<td>Converter for <code>java.lang.Short</code> (and short primitive) values.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConverterException</td>
<td><code>ConverterException</code> is an exception thrown by the <code>getAsObject()</code> or <code>getAsText()</code> method of a <code>Converter</code>, to indicate that the requested conversion cannot be performed.</td>
</tr>
</tbody>
</table>
Package javax.faces.convert Description

Contains classes and interfaces defining converters. The main class in this package is Converter.
Hierarchy For Package javax.faces.convert

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.convert.**BigDecimalConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**BigIntegerConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**BooleanConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**ByteConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**CharacterConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**DateTimeConverter** (implements javax.faces.convert.**Converter**, javax.faces.component.**StateHolder**)
  - javax.faces.convert.**DoubleConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**EnumConverter** (implements javax.faces.convert.**Converter**, javax.faces.component.**StateHolder**)
  - javax.faces.convert.**FloatConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**IntegerConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**LongConverter** (implements javax.faces.convert.**Converter**)
  - javax.faces.convert.**NumberConverter** (implements javax.faces.convert.**Converter**, javax.faces.component.**StateHolder**)
  - javax.faces.convert.**ShortConverter** (implements javax.faces.convert.**Converter**)
- java.lang.**Throwable** (implements java.io.**Serializable**)
  - java.lang.**Exception**
    - java.lang.**RuntimeException**
      - javax.faces.**FacesException**
      - javax.faces.convert.**ConverterException**
Interface Hierarchy

- javax.faces.convert.**Converter**
Package javax.xml.rpc.holders

This package contains the standard Java Holder classes.

See: Description

### Interface Summary

| Holder | The java.xml.rpc.holders.Holder interface represents the base interface for both standard and generated Holder classes. |

### Class Summary

<table>
<thead>
<tr>
<th>Class Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>BigDecimalHolder</td>
</tr>
<tr>
<td>BigIntegerHolder</td>
</tr>
<tr>
<td>BooleanHolder</td>
</tr>
<tr>
<td>BooleanWrapperHolder</td>
</tr>
<tr>
<td>ByteArrayHolder</td>
</tr>
<tr>
<td>ByteHolder</td>
</tr>
<tr>
<td>ByteWrapperHolder</td>
</tr>
<tr>
<td>CalendarHolder</td>
</tr>
<tr>
<td>DoubleHolder</td>
</tr>
<tr>
<td>DoubleWrapperHolder</td>
</tr>
<tr>
<td>FloatHolder</td>
</tr>
<tr>
<td>FloatWrapperHolder</td>
</tr>
<tr>
<td>IntegerWrapperHolder</td>
</tr>
<tr>
<td>IntHolder</td>
</tr>
<tr>
<td>LongHolder</td>
</tr>
<tr>
<td>LongWrapperHolder</td>
</tr>
<tr>
<td>ObjectHolder</td>
</tr>
<tr>
<td>QNameHolder</td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.holders Description

This package contains the standard Java Holder classes.
Hierarchy For Package javax.xml.rpc.holders

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.rpc.holders.**BigDecimalHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**BigIntegerHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**BooleanHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**BooleanWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**ByteArrayHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**ByteHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**ByteWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**CalendarHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**DoubleHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**DoubleWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**FloatHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**FloatWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**IntegerWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**IntHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**LongHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**LongWrapperHolder** (implements javax.xml.rpc.holders.**Holder**)
  - javax.xml.rpc.holders.**ObjectHolder** (implements javax.xml.rpc.holders.**Holder**)

 class hierarchy
- javax.xml.rpc.holders.QNameHolder (implements javax.xml.rpc.holders.Holder)
- javax.xml.rpc.holders.ShortHolder (implements javax.xml.rpc.holders.Holder)
- javax.xml.rpc.holders.ShortWrapperHolder (implements javax.xml.rpc.holders.Holder)
- javax.xml.rpc.holders.StringHolder (implements javax.xml.rpc.holders.Holder)
## Interface Hierarchy

- [javax.xml.rpc.holders.Holder](#)

### Table

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Class</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

[Submit a bug or feature](#)

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Package javax.xml.bind

Provides a runtime binding framework for client applications including unmarshalling, marshalling, and validation capabilities.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DatatypeConverterInterface</td>
<td>The DatatypeConverterInterface is for JAXB provider use only.</td>
</tr>
<tr>
<td>Element</td>
<td>This is an element marker interface.</td>
</tr>
<tr>
<td>Marshaller</td>
<td>The Marshaller class is responsible for governing the process of serializing Java content trees back into XML data.</td>
</tr>
<tr>
<td>NotIdentifiableEvent</td>
<td>This event indicates that a problem was encountered resolving an ID/IDREF.</td>
</tr>
<tr>
<td>ParseConversionEvent</td>
<td>This event indicates that a problem was encountered while converting a string from the XML data into a value of the target Java data type.</td>
</tr>
<tr>
<td>PrintConversionEvent</td>
<td>This event indicates that a problem was encountered while converting data from the Java content tree into its lexical representation.</td>
</tr>
<tr>
<td>Unmarshaller</td>
<td>The Unmarshaller class governs the process of deserializing XML data into newly created Java content trees, optionally validating the XML data as it is unmarshalled.</td>
</tr>
<tr>
<td>UnmarshallerHandler</td>
<td>Unmarshaller implemented as SAX ContentHandler.</td>
</tr>
<tr>
<td></td>
<td>This event indicates that a problem was encountered while validating the incoming XML data during an unmarshal operation,</td>
</tr>
</tbody>
</table>
### ValidationEvent
while performing on-demand validation of the Java content tree, or while marshalling the Java content tree back to XML data.

### ValidationEventHandler
A basic event handler interface for validation errors.

### ValidationEventLocator
Encapsulate the location of a ValidationEvent.

### Validator
**Deprecated. since JAXB 2.0**

---

## Class Summary

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Binder&lt;XmlNode&gt;</strong></td>
<td>Enable synchronization between XML infoset nodes and JAXB objects representing same XML document.</td>
</tr>
<tr>
<td><strong>DatatypeConverter</strong></td>
<td>The javaType binding declaration can be used to customize the binding of an XML schema datatype to a Java datatype.</td>
</tr>
<tr>
<td><strong>JAXBContext</strong></td>
<td>The JAXBContext class provides the client's entry point to the JAXB API.</td>
</tr>
<tr>
<td><strong>JAXBElement&lt;T&gt;</strong></td>
<td>JAXB representation of an Xml Element.</td>
</tr>
<tr>
<td><strong>JAXBElement.GlobalScope</strong></td>
<td>Designates global scope for an xml element.</td>
</tr>
<tr>
<td><strong>JAXBIntrospector</strong></td>
<td>Provide access to JAXB xml binding data for a JAXB object.</td>
</tr>
<tr>
<td><strong>Marshaller.Listener</strong></td>
<td>Register an instance of an implementation of this class with a Marshaller to externally listen for marshal events.</td>
</tr>
<tr>
<td><strong>SchemaOutputResolver</strong></td>
<td>Controls where a JAXB implementation puts the generates schema files.</td>
</tr>
<tr>
<td><strong>Unmarshaller.Listener</strong></td>
<td>Register an instance of an implementation of this class with Unmarshaller to</td>
</tr>
</tbody>
</table>
**Exception Summary**

<table>
<thead>
<tr>
<th>Exception Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXBException</td>
<td>This is the root exception class for all JAXB exceptions.</td>
</tr>
<tr>
<td>MarshalException</td>
<td>This exception indicates that an error has occurred while performing a marshal operation that the provider is unable to recover from.</td>
</tr>
<tr>
<td>PropertyException</td>
<td>This exception indicates that an error was encountered while getting or setting a property.</td>
</tr>
<tr>
<td>TypeConstraintException</td>
<td>This exception indicates that a violation of a dynamically checked type constraint was detected.</td>
</tr>
<tr>
<td>UnmarshalException</td>
<td>This exception indicates that an error has occurred while performing an unmarshal operation that prevents the JAXB Provider from completing the operation.</td>
</tr>
<tr>
<td>ValidationException</td>
<td>This exception indicates that an error has occurred while performing a validate operation.</td>
</tr>
</tbody>
</table>
Package javax.xml.bind Description

Provides a runtime binding framework for client applications including unmarshalling, marshalling, and validation capabilities.

JAXBContext is the client-entry point to the runtime binding framework.
Package Specification

- JAXB Specification
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- The [JAXB Website](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package javax.xml.bind

Package Hierarchies:
  All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.xml.bind.Binder<XmlNode>`
  - `javax.xml.bind.DatatypeConverter`
  - `javax.xml.bind.JAXBContext`
  - `javax.xml.bind.JAXBElement<T>` (implements `java.io.Serializable`)
  - `javax.xml.bind.JAXBElemnet.GlobalScope`
  - `javax.xml.bind.JAXBIntrospector`
  - `javax.xml.bind.Marshaller.Listener`
  - `javax.xml.bind.SchemaOutputResolver`
  - `java.lang.Throwable` (implements `java.io.Serializable`)
    - `java.lang.Exception`
      - `javax.xml.bind.JAXBException`
        - `javax.xml.bind.MarshalException`
        - `javax.xml.bind.PropertyException`
        - `javax.xml.bind.UnmarshalException`
        - `javax.xml.bind.ValidationException`
    - `java.lang.RuntimeException`
      - `javax.xml.bind.TypeConstraintException`
  - `javax.xml.bind.Unmarshaller.Listener`
Interface Hierarchy

- org.xml.sax.ContentHandler
  - javax.xml.bind.UnmarshallerHandler
- javax.xml.bind.DatatypeConverterInterface
- javax.xml.bind.Element
- javax.xml.bind.Marshaller
- javax.xml.bind.Unmarshaller
  - javax.xml.bind.NotIdentifiableEvent
  - javax.xml.bind.ParseConversionEvent
  - javax.xml.bind.PrintConversionEvent
- javax.xml.bind.ValidationEventHandler
- javax.xml.bind.ValidationEventLocator
- javax.xml.bind.Validator

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.servlet.jsp.tagext`

Classes and interfaces for the definition of JavaServer Pages Tag Libraries.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BodyTag</strong></td>
<td>The BodyTag interface extends IterationTag by defining additional methods that let a tag handler manipulate the content of evaluating its body.</td>
</tr>
<tr>
<td><strong>DynamicAttributes</strong></td>
<td>For a tag to declare that it accepts dynamic attributes, it must implement this interface.</td>
</tr>
<tr>
<td><strong>IterationTag</strong></td>
<td>The IterationTag interface extends Tag by defining one additional method that controls the reevaluation of its body.</td>
</tr>
<tr>
<td><strong>JspIdConsumer</strong></td>
<td>This interface indicates to the container that a tag handler wishes to be provided with a compiler generated ID.</td>
</tr>
<tr>
<td><strong>JspTag</strong></td>
<td>Serves as a base class for Tag and SimpleTag.</td>
</tr>
<tr>
<td><strong>SimpleTag</strong></td>
<td>Interface for defining Simple Tag Handlers.</td>
</tr>
<tr>
<td><strong>Tag</strong></td>
<td>The interface of a classic tag handler that does not want to manipulate its body.</td>
</tr>
<tr>
<td><strong>TryCatchFinally</strong></td>
<td>The auxiliary interface of a Tag, IterationTag or BodyTag tag handler that wants additional hooks for managing resources.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BodyContent</strong></td>
<td>An encapsulation of the evaluation of the body of an action so it is available to a tag handler.</td>
</tr>
<tr>
<td><strong>BodyTagSupport</strong></td>
<td>A base class for defining tag handlers implementing BodyTag.</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FunctionInfo</td>
<td>Information for a function in a Tag Library.</td>
</tr>
<tr>
<td>JspFragment</td>
<td>Encapsulates a portion of JSP code in an object that can be invoked as many times as needed.</td>
</tr>
<tr>
<td>PageData</td>
<td>Translation-time information on a JSP page.</td>
</tr>
<tr>
<td>SimpleTagSupport</td>
<td>A base class for defining tag handlers implementing SimpleTag.</td>
</tr>
<tr>
<td>TagAdapter</td>
<td>Wraps any SimpleTag and exposes it using a Tag interface.</td>
</tr>
<tr>
<td>TagAttributeInfo</td>
<td>Information on the attributes of a Tag, available at translation time.</td>
</tr>
<tr>
<td>TagData</td>
<td>The (translation-time only) attribute/value information for a tag instance.</td>
</tr>
<tr>
<td>TagExtraInfo</td>
<td>Optional class provided by the tag library author to describe additional translation-time information not described in the TLD.</td>
</tr>
<tr>
<td>TagFileInfo</td>
<td>Tag information for a tag file in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.</td>
</tr>
<tr>
<td>TagInfo</td>
<td>Tag information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.</td>
</tr>
<tr>
<td>TagLibraryInfo</td>
<td>Translation-time information associated with a taglib directive, and its underlying TLD file.</td>
</tr>
<tr>
<td>TagLibraryValidator</td>
<td>Translation-time validator class for a JSP page.</td>
</tr>
<tr>
<td>TagSupport</td>
<td>A base class for defining new tag handlers implementing Tag.</td>
</tr>
<tr>
<td>TagVariableInfo</td>
<td>Variable information for a tag in a Tag Library; This class is instantiated from the Tag Library Descriptor file (TLD) and is available only at translation time.</td>
</tr>
<tr>
<td>ValidationMessage</td>
<td>A validation message from either TagLibraryValidator or TagExtraInfo.</td>
</tr>
<tr>
<td></td>
<td>Information on the scripting variables that are</td>
</tr>
<tr>
<td>VariableInfo</td>
<td>created/modified by a tag (at run-time).</td>
</tr>
</tbody>
</table>
Package javax.servlet.jsp.tagext Description

Classes and interfaces for the definition of JavaServer Pages Tag Libraries.

Custom actions can be used by JSP authors and authoring tools to simplify writing JSP pages. A custom action can be either an empty or a non-empty action.

An empty tag has no body. There are two equivalent syntaxes, one with separate start and end tags, and one where the start and end tags are combined. The two following examples are identical:

```xml
<x:foo att="myObject"></foo>
<x:foo att="myObject"/>
```

A non-empty tag has a start tag, a body, and an end tag. A prototypical example is of the form:

```xml
<x:foo att="myObject">
    BODY
</x:foo/>
```

The JavaServer Pages(tm) (JSP) specification provides a portable mechanism for the description of tag libraries.

A JSP tag library contains

- A Tag Library Descriptor
- A number of Tag Files or Tag handler classes defining request-time behavior
- Additional classes and resources used at runtime
- Possibly some additional classes to provide extra translation information

This API is described in the following sections:

1. Classic Tag Handlers
2. Tag Handlers that want to access their Body Content
3. Dynamic Attributes
4. Annotated Tag Handler Management Example
5. Cooperating Actions
6. Simple Tag Handlers
7. JSP Fragments
8. Example Simple Tag Handler Scenario
9. Translation-time Classes
1. Classic Tag Handlers

This section introduces the notion of a tag handler and describes the classic types of tag handler.

JSP 2.0 introduces a new type of Tag Handler called a Simple Tag Handler, which is described in a later section. The protocol for Simple Tag handlers is much more straightforward.

Tag Handler

A tag handler is a run-time, container-managed object that evaluates custom actions during the execution of a JSP page. A tag handler supports a protocol that allows the JSP container to provide good integration of the server-side actions within a JSP page.

A tag handler is created initially using a zero argument constructor on its corresponding class; the method `java.beans.Beans.instantiate()` is not used.

A tag handler has some properties that are exposed to the page as attributes on an action; these properties are managed by the JSP container (via generated code). The setter methods used to set the properties are discovered using the JavaBeans introspector machinery.

The protocol supported by a tag handler provides for passing of parameters, the evaluation and reevaluation of the body of the action, and for getting access to objects and other tag handlers in the JSP page.

A tag handler instance is responsible for processing one request at a time. It is the responsibility of the JSP container to enforce this.

Additional translation time information associated with the action indicates the name of any scripting variables it may introduce, their types and their scope. At specific moments, the JSP container will automatically synchronize the `PageContext` information with variables in the scripting language so they can be made available directly through the scripting elements.
Properties

A tag handler has some properties. All tag handlers have a `pageContext` property for the JSP page where the tag is located, and a `parent` property for the tag handler to the closest enclosing action. Specific tag handler classes may have additional properties.

All attributes of a custom action must be JavaBeans component properties, although some properties may not be exposed as attributes. The attributes that are visible to the JSP translator are exactly those listed in the Tag Library Descriptor (TLD).

All properties of a tag handler instance exposed as attributes will be initialized by the container using the appropriate setter methods before the instance can be used to perform the action methods. It is the responsibility of the JSP container to invoke the appropriate setter methods to initialize these properties. It is the responsibility of user code, be it scriptlets, JavaBeans code, or code inside custom tags, to not invoke these setter methods, as doing otherwise would interfere with the container knowledge.

The setter methods that should be used when assigning a value to an attribute of a custom action are determined by using the JavaBeans introspector on the tag handler class, then use the setter method associated with the property that has the same name as the attribute in question. An implication (unclear in the JavaBeans specification) is that there is only one setter per property.

Unspecified attributes/properties should not be set (using a setter method).

Once properly set, all properties are expected to be persistent, so that if the JSP container ascertains that a property has already been set on a given tag handler instance, it must not set it again.

The JSP container may reuse classic tag handler instances for multiple occurrences of the corresponding custom action, in the same page or in different pages, but only if the same set of attributes are used for all occurrences. If a tag handler is used for more than one occurrence, the
container must reset all attributes where the values differ between the custom action occurrences. Attributes with the same value in all occurrences must not be reset. If an attribute value is set as a request-time attribute value (using a scripting or an EL expression), the container must reset the attribute between all reuses of the tag handler instance.

User code can access property information and access and modify tag handler internal state starting with the first action method (doStartTag) up until the last action method (doEndTag or doFinally for tag handlers implementing TryCatchFinally).

**Tag Handler as a Container-Managed Object**

Since a tag handler is a container managed object, the container needs to maintain its references; specifically, user code should not keep references to a tag handler except between the start of the first action method (doStartTag()) and the end of the last action method (doEndTag() or doFinally() for those tags that implement TryCatchFinally).

The restrictions on references to tag handler objects and on modifying attribute properties gives the JSP container substantial freedom in effectively managing tag handler objects to achieve different goals. For example, a container may implementing different pooling strategies to minimize creation cost, or may hoist setting of properties to reduce cost when a tag handler is inside another iterative tag.

**Conversions**

A tag handler implements an action; the JSP container must follow the type conversions described in Section 2.13.2 when assigning values to the attributes of an action.

**Empty and Non-Empty Actions**

An empty action has no body; it may use one of two syntaxes: either <foo/> or <foo/></foo>. Since empty actions have no body the methods related to body manipulation are not invoked. There is a mechanism in the Tag Library Descriptor to indicate that a tag can only be used to write empty actions; when used, non-empty actions using that tag will produce
a translation error.

A non-empty action has a body.

**The Tag Interface**

A Tag handler that does not want to process its body can implement just the Tag interface. A tag handler may not want to process its body because it is an empty tag or because the body is just to be "passed through".

The Tag interface includes methods to provide page context information to the Tag Handler instance, methods to handle the life-cycle of tag handlers, and two main methods for performing actions on a tag: `doStartTag()` and `doEndTag()`. The method `doStartTag()` is invoked when encountering the start tag and its return value indicates whether the body (if there is any) should be skipped, or evaluated and passed through to the current response stream. The method `doEndTag()` is invoked when encountering the end tag; its return value indicates whether the rest of the page should continue to be evaluated or not.

If an exception is encountered during the evaluation of the body of a tag, its `doEndTag` method will not be evaluated. See the TryCatchFinally tag for methods that are guaranteed to be evaluated.

**The IterationTag Interface**

The IterationTag interface is used to repeatedly reevaluate the body of a custom action. The interface has one method: `doAfterBody()` which is invoked after each evaluation of the body to determine whether to reevaluate or not.

Reevaluation is requested with the value 2, which in JSP 1.1 is defined to be `BodyTag.EVAL_BODY_TAG`. That constant value is still kept in JSP 1.2 (for full backwards compatibility) but, to improve clarity, a new name is also available: `IterationTag.EVAL_BODY_AGAIN`. To stop iterating, the returned value should be 0, which is `Tag.SKIP_BODY`.

**The JspIdConsumer Interface**
This interface indicates to the container that a tag handler wishes to be provided with a compiler generated ID that is unique within the page.

The **TagSupport Base Class**

The TagSupport class is a base class that can be used when implementing the Tag or IterationTag interfaces.
2. Tag Handlers that want Access to their Body Content

The evaluation of a body is delivered into a BodyContent object. This is then made available to tag handlers that implement the BodyTag interface. The BodyTagSupport class provides a useful base class to simplify writing these handlers.

If a Tag handler wants to have access to the content of its body then it must implement the BodyTag interface. This interface extends IterationTag, provides two additional methods setBodyContent(BodyContent) and doInitBody() and refers to an object of type BodyContent.

A BodyContent is a subclass of JspWriter that has a few additional methods to convert its contents into a String, insert the contents into another JspWriter, to get a Reader into its contents, and to clear the contents. Its semantics also assure that buffer size will never be exceeded.

The JSP page implementation will create a BodyContent if the doStartTag() method returns a EVAL_BODY_BUFFERED. This object will be passed to doInitBody(); then the body of the tag will be evaluated, and during that evaluation out will be bound to the BodyContent just passed to the BodyTag handler. Then doAfterBody() will be evaluated. If that method returns SKIP_BODY, no more evaluations of the body will be done; if the method returns EVAL_BODY_AGAIN, then the body will be evaluated, and doAfterBody() will be invoked again.

The content of a BodyContent instance remains available until after the invocation of its associated doEndTag() method.

A common use of the BodyContent is to extract its contents into a String and then use the String as a value for some operation. Another common use is to take its contents and push it into the out Stream that was valid when the start tag was encountered (that is available from the PageContext object passed to the handler in setPageContext).
3. Dynamic Attributes

Any tag handler can optionally extend the DynamicAttributes interface to indicate that it supports dynamic attributes. In addition to implementing the DynamicAttributes interface, tag handlers that support dynamic attributes must declare that they do so in the Tag Library Descriptor.

The TLD is what ultimately determines whether a tag handler accepts dynamic attributes or not. If a tag handler declares that it supports dynamic attributes in the TLD but it does not implement the DynamicAttributes interface, the tag handler must be considered invalid by the container.

If the dynamic-attributes element for a tag being invoked contains the value "true", the following requirements apply:

- For each attribute specified in the tag invocation that does not have a corresponding attribute element in the TLD for this tag, a call must be made to setDynamicAttribute(), passing in the namespace of the attribute (or null if the attribute does not have a namespace or prefix), the name of the attribute without the namespace prefix, and the final value of the attribute.
- Dynamic attributes must be considered to accept request-time expression values as well as deferred expressions.
- Dynamic attributes must be treated as though they were of type java.lang.Object. If a ValueExpression is passed as a dynamic attribute, the default value for the expected return type is assumed to be java.lang.Object. If a MethodExpression is passed as a dynamic attribute, the default method signature is assumed to be void method().
- Note that passing a String literal as a dynamic attribute will never be considered as a deferred expression.
- The JSP container must recognize dynamic attributes that are passed to the tag handler using the <jsp:attribute> standard action.
- If the setDynamicAttribute() method throws JspException, the doStartTag() or doTag() method is not invoked for this tag, and the exception must be treated in the same manner as if it came from a regular attribute setter method.
For a JSP document in either standard or XML syntax, if a dynamic attribute has a prefix that doesn't map to a namespace, a translation error must occur. In standard syntax, only namespaces defined using taglib directives are recognized.

In the following example, assume attributes a and b are declared using the attribute element in the TLD, attributes d1 and d2 are not declared, and the dynamic-attributes element is set to "true". The attributes are set using the calls:

- setA( "1" ),
- setDynamicAttribute( null, "d1", "2" ),
- setB( "4" ),
- setDynamicAttribute( null, "d3", "5" ), and

<jsp:root xmlns:mytag="http://www.foo.com/jsp/taglib/mytag.tld" version="2.0">
  <mytag:invokeDynamic a="1" d1="2" mytag:d2="3">
    <jsp:attribute name="b">4</jsp:attribute>
    <jsp:attribute name="d3">5</jsp:attribute>
    <jsp:attribute name="mytag:d4">6</jsp:attribute>
  </mytag:invokeDynamic>
</jsp:root>
4. Annotated Tag Handler Management Example

Below is a somewhat complete example of the way one JSP container could choose to do some tag handler management. There are many other strategies that could be followed, with different pay offs.

In this example, we are assuming that x:iterate is an iterative tag, while x:doit and x:foobar are simple tag. We will also assume that x:iterate and x:foobar implement the TryCatchFinally interface, while x:doit does not.

```xml
<x:iterate src="foo">
  <x:doit att1="one" att2="<%= 1 + 1 %>">
  </x:doit>
  <x:foobar />
  <x:doit att1="one" att2="<%= 2 + 2 %>">
  </x:doit>
</x:iterate>
<x:doit att1="one" att2="<%= 3 + 3 %>">
</x:doit>
```

The particular code shown below assumes there is some pool of tag handlers that are managed (details not described, although pool managing is simpler when there are no optional attributes), and attempts to reuse tag handlers if possible. The code also "hoists" setting of properties to reduce the cost when appropriate, e.g. inside an iteration.

```java
boolean b1, b2;
IterationTag i; // for x:iterate
Tag d; // for x:doit
Tag d; // for x:foobar

page: // label to end of page...

// initialize iteration tag
i = get tag from pool or new();

i.setPageContext(pc);
i.setParent(null);
i.setSrc("foo");

// x:iterate implements TryCatchFinally
try {
  if ((b1 = i.doStartTag()) == EVAL_BODY_INCLUDE) {
    // initialize doit tag
```
// code has been moved out of the loop for show
d = get tag from pool or new();

d.setPageContext(pc);
d.setParent(i);
d.setAtt1("one");

loop:
  while (1) do {
    // I'm ignoring newlines...

    // two invocations, fused together

    // first invocation of x:doit
d.setAtt2(1+1);
    if ((b2 = d.doStartTag()) == EVAL_BODY_INCLUDE) {
      // nothing
    } else if (b2 != SKIP_BODY) {
      // Q? protocol error ...
    }
  if ((b2 = d.doEndTag()) == SKIP_PAGE) {
      break page; // be done with it.
    } else if (b2 != EVAL_PAGE) {
      // Q? protocol error
    }

    // x:foobar invocation
f = get tag from pool or new();
f.setPageContext(pc);
f.setParent(i);

    // x:foobar implements TryCatchFinally
try {
  if ((b2 = f.doStartTag()) == EVAL_BODY_INCLUDE) {
    // nothing
  } else if (b2 != SKIP_BODY) {
    // Q? protocol error
  }
  if ((b2 = f.doEndTag()) == SKIP_PAGE) {
    break page; // be done with it.
  } else if (b2 != EVAL_PAGE) {
    // Q? protocol error
  }
} catch (Throwable t) {
  f.doCatch(t); // caught, may been rethrown!
} finally {
  f.doFinally();
}
// put f back to pool

// second invocation of x:doit
d.setAtt2(2+2);
if ((b2 = d.doStartTag()) == EVAL_BODY_INCLUDE) {
    // nothing
} else if (b2 != SKIP_BODY) {
    // Q? protocol error
}
if ((b2 = d.doEndTag()) == SKIP_PAGE) {
    break page; // be done with it.
} else if (b2 != EVAL_PAGE) {
    // Q? protocol error
}

if ((b2 = i.doAfterBody()) == EVAL_BODY_AGAIN) {
    break loop;
} else if (b2 != SKIP_BODY) {
    // Q? protocol error
}

// loop
}

} else if (b1 != SKIP_BODY) {
    // Q? protocol error
}

// tail end of the IteratorTag ...

if ((b1 = i.doEndTag()) == SKIP_PAGE) {
    break page; // be done with it.
} else if (b1 != EVAL_PAGE) {
    // Q? protocol error
}

// third invocation
// this tag handler could be reused from the previous ones.
d = get tag from pool or new();
d setPageContext(pc);
d.setParent(null);
d.setAtt1("one");
d.setAtt2(3+3);
if ((b1 = d.doStartTag()) == EVAL_BODY_INCLUDE) {
    // nothing
} else if (b1 != SKIP_BODY) {
    // Q? protocol error
if ((b1 = d.doEndTag()) == SKIP_PAGE) {
    break page; // be done with it.
} else if (b1 != EVAL_PAGE) {
    // Q? protocol error
}

catch (Throwable t) {
    i.doCatch(t); // caught, may been rethrown!
} finally {
    i.doFinally();
}
5. Cooperating Actions

Actions can cooperate with other actions and with scripting code in a number of ways.

PageContext

Often two actions in a JSP page will want to cooperate, perhaps by one action creating some server-side object that needs to be accessed by another. One mechanism for doing this is by giving the object a name within the JSP page; the first action will create the object and associate the name to it while the second action will use the name to retrieve the object.

For example, in the following JSP segment the foo action might create a server-side object and give it the name "myObject". Then the bar action might access that server-side object and take some action.

```xml
<x:foo id="myObject" />
<x:bar ref="myObject" />
```

In a JSP implementation, the mapping "name"->value is kept by the implicit object `pageContext`. This object is passed around through the Tag handler instances so it can be used to communicate information: all it is needed is to know the name under which the information is stored into the `pageContext`.

The Runtime Stack

An alternative to explicit communication of information through a named object is implicit coordination based on syntactic scoping.

For example, in the following JSP segment the foo action might create a server-side object; later the nested bar action might access that server-side object. The object is not named within the `pageContext`: it is found because the specific foo element is the closest enclosing instance of a known element type.
This functionality is supported through the
TagSupport.findAncestorWithClass(Tag, Class), which uses a reference
to parent tag kept by each Tag instance, which effectively provides a run-
time execution stack.
6. Simple Tag Handlers

This section presents the API to implement Simple Tag Handlers. Simple Tag Handlers present a much simpler invocation protocol than do Classic Tag Handlers.

The Tag Library Descriptor maps tag library declarations to their physical underlying implementations. A Simple Tag Handler is represented in Java by a class which implements the SimpleTag interface.

Unlike classic tag handlers, the SimpleTag interface does not extend Tag. Instead of supporting doStartTag() and doEndTag(), the SimpleTag interface provides a simple doTag() method, which is called once and only once for any given tag invocation. All tag logic, iteration, body evaluations, etc. are to be performed in this single method. Thus, simple tag handlers have the equivalent power of BodyTag, but with a much simpler lifecycle and interface.

To support body content, the setJspBody() method is provided. The container invokes the setJspBody() method with a JspFragment object encapsulating the body of the tag. The tag handler implementation can call invoke() on that fragment to evaluate the body. The SimpleTagSupport convenience class provides getJspBody() and other useful methods to make this even easier.

Lifecycle of Simple Tag Handlers

This section describes the lifecycle of simple tag handlers, from creation to invocation. For all semantics left unspecified by this section, the semantics default to that of a classic tag handler.

When a simple tag handler is invoked, the following steps occur (in order):

1. Simple tag handlers are created initially using a zero argument constructor on the corresponding implementation class. Unlike classic tag handlers, this instance must never be pooled by the container. A new instance must be created for each tag invocation.
2. The setJspContext() and setParent() methods are invoked on the tag handler. The setParent() method need not be called if the value being passed in is null. In the case of tag files, a JspContext wrapper is created so that the tag file can appear to have its own page scope. Calling getJspContext() must return the wrapped JspContext.

3. The attributes specified as XML element attributes (if any) are evaluated next, in the order in which they are declared, according to the following rules (referred to as "evaluating an XML element attribute" below). The appropriate bean property setter is invoked for each. If no setter is defined for the specified attribute but the tag accepts dynamic attributes, the setDynamicAttribute() method is invoked as the setter.
   - If the attribute is a scripting expression (e.g. "<%= 1+1 %>" in JSP syntax, or "%= 1+1 %" in XML syntax), the expression is evaluated, and the result is converted as per the rules in "Type Conversions", and passed to the setter.
   - Otherwise, if the attribute contains any Expression Language expressions (e.g. "Hello ${name}"), the expression is evaluated, and the result is converted and passed to the setter.
   - Otherwise, the attribute value is taken verbatim, converted, and passed to the setter.

4. The value for each <jsp:attribute> element is evaluated, and the corresponding bean property setter methods are invoked for each, in the order in which they appear in the body of the tag. If no setter is defined for the specified attribute but the tag accepts dynamic attributes, the setDynamicAttribute() method is invoked as the setter.
   - Otherwise, if the attribute is not of type JspFragment, the container evaluates the body of the <jsp:attribute> element. This evaluation can be done in a container-specific manner. Container implementors should note that in the process of evaluating this body, other custom actions may be invoked.
   - Otherwise, if the attribute is of type JspFragment, an instance of a JspFragment object is created and passed in.

5. The value for the body of the tag is determined, and if a body exists the setJspBody() method is called on the tag handler.
   - If the tag is declared to have a body-content of "empty" or no body or an empty body is passed for this invocation, then
setJspBody() is not called.

- Otherwise, the body of the tag is either the body of the 
  &lt;jsp:body&gt; element, or the body of the custom action invocation 
  if no &lt;jsp:body&gt; or &lt;jsp:attribute&gt; elements are present. In this 
  case, an instance of a JspFragment object is created as per the 
  lifecycle described in the JSP Fragments section and it is 
  passed to the setter. If the tag is declared to have a body- 
  content of "tagdependent" the JspFragment must echo the body's 
  contents verbatim. Otherwise, if the tag is declared to have a 
  body-content of type "scriptless", the JspFragment must 
  evaluate the body's contents as a JSP scriptless body.

6. The doTag() method is invoked.
7. The implementation of doTag() performs its function, potentially 
   calling other tag handlers (if the tag handler is implemented as a tag 
   file) and invoking fragments.
8. The doTag() method returns, and the tag handler instance is 
   discarded. If SkipPageException is thrown, the rest of the page is not 
   evaluated and the request is completed. If this request was 
   forwarded or included from another page (or Servlet), only the 
   current page evaluation stops.
9. For each tag scripting variable declared with scopes AT_BEGIN or 
   AT_END, the appropriate scripting variables and scoped attributes are 
   declared, as with classic tag handlers.
7. JSP Fragments

JSP Fragments are represented in Java by an instance of the
javax.servlet.jsp.tagext.JspFragment abstract class. Pieces of JSP
code are translated into JSP fragments in the context of a tag invocation.
JSP Fragments are created when providing the body of a <jsp:attribute>
standard action for an attribute that is defined as a fragment or of type
JspFragment, or when providing the body of a tag invocation handled by
a Simple Tag Handler.

Before being passed to a tag handler, the JspFragment instance is
associated with the JspContext of the surrounding page in an
implementation-dependent manner. In addition, it is associated with the
parent Tag or SimpleTag instance for collaboration purposes, so that when
a custom action is invoked from within the fragment, setParent() can be
called with the appropriate value. The fragment implementation must
keep these associations for the duration of the tag invocation in which it
is used.

The invoke() method executes the body and directs all output to either
the passed in java.io.Writer or the JspWriter returned by the getOut()
method of the JspContext associated with the fragment.

The implementation of each method can optionally throw a JspException,
which must be handled by the invoker. Note that tag library developers
and page authors should not generate JspFragment implementations
manually.

The following sections specify the creation and invocation lifecycles of a
JSP Fragment in detail, from the JSP Container's perspective.

Creation of a JSP Fragment

When a JSP fragment is created, the following steps occur (in order):

1. An instance of a class implementing the JspFragment abstract class
   is obtained (may either be created or can optionally be cached) each
time the tag is invoked. This instance must be configured to produce
the contents of the body of the fragment when invoked. If the fragment is defining the body of a `<jsp:attribute>`, the fragment must evaluate the body each time it is invoked. Otherwise, if the fragment is defining the body of a simple tag, the behavior of the fragment when invoked varies depending on the `body-content` declared for the tag:

- If the `body-content` is "tagdependent", then the fragment must echo the contents of the body verbatim when invoked.
- If the `body-content` is "scriptless", then the fragment must evaluate the body each time it is invoked.

2. The `JspFragment` instance is passed a reference to the current `JspContext`. Whenever the fragment invokes a tag handler, it must use this value when calling `setJspContext()`.

3. The `JspFragment` instance is associated with an instance of the tag handler of the nearest enclosing tag invocation, or with `null` if there is no enclosing tag. Whenever the fragment invokes a tag handler, the fragment must use this value when calling `setParent()`.

**Invocation of a JSP Fragment**

After a JSP fragment is created, it is passed to a tag handler for later invocation. JSP fragments can be invoked either programmatically from a tag handler written in Java, or from a tag file using the `<jsp:invoke>` or `<jsp:doBody>` standard action.

JSP fragments are passed to tag handlers using a bean property of type `JspFragment`. These fragments can be invoked by calling the `invoke()` method in the `JspFragment` abstract class. Note that it is legal (and possible) for a fragment to recursively invoke itself, indirectly.

The following steps are followed when invoking a JSP fragment:

1. The tag handler invoking the fragment is responsible for setting the values of all declared `AT_BEGIN` and `NESTED` variables in the `JspContext` of the calling page/tag, before invoking the fragment. Note that this is not always the same as the `JspContext` of the fragment being invoked, as fragments can be passed from one tag to another. In the case of tag files, for each variable declared in scope `AT_BEGIN` or `NESTED`, if a page scoped attribute exists with the
provided name in the tag file, the JSP container must generate code to create/update the page scoped attribute of the provided name in the calling page/tag. If a page scoped attribute with the provided name does not exist in the calling page, and a page scoped attribute of the provided name is present in the tag file, the scoped attribute is removed from the tag file's page scope. See the chapter on Tag Files for details.

2. If `<jsp:invoke>` or `<jsp:doBody>` is being used to invoke a fragment, if the var attribute is specified, a custom java.io.Writer is created that can expose the result of the invocation as a java.lang.String object. If the varReader attribute is specified, a custom java.io.Writer object is created that can expose the resulting invocation as a java.io.Reader object.

3. The `invoke()` method of the fragment is invoked, passing in an optional Writer.

4. Before executing the body of the fragment, if a non-null value is provided for the writer parameter, then the value of JspContext.getOut() and the implicit "out" object must be updated to send output to that writer. To accomplish this, the container must call pushBody( writer ) on the current JspContext, where writer is the instance of java.io.Writer passed to the fragment upon invocation.

5. The body of the fragment is then evaluated by executing the generated code. The body of the fragment may execute other standard or custom actions. If a classic Custom Tag Handler is invoked and returns SKIP_PAGE, or if a Simple Tag Handler is invoked and throws SkipPageException, the JspFragment must throw SkipPageException to signal that the calling page is to be skipped.

6. Once the fragment has completed its evaluation, even if an exception is thrown, the value of JspContext.getOut() must be restored via a call to popBody() on the current JspContext.

7. The fragment returns from invoke()

8. If `<jsp:invoke>` or `<jsp:doBody>` is being used to invoke a fragment, if the var or varReader attribute is specified, a scoped variable with a name equal to the value of the var or varReader attribute is created (or modified) in the page scope, and the value is set to a java.lang.String or java.io.Reader respectively that can produce the results of the fragment invocation.

9. The `invoke()` method can be called again, zero or more times. When the tag invocation defining the fragment is complete, the tag must
discard the fragment instance since it might be reused by the container.
8. Example Simple Tag Handler Scenario

The following non-normative example is intended to help solidify some of the concepts relating to Tag Files, JSP Fragments and Simple Tag Handlers. In the first section, two sample input files are presented, a JSP (my.jsp), and a simple tag handler implemented using a tag file (simpletag.tag). One possible output of the translation process is presented in the second section.

Although short, the example shows all concepts, including the variable directive. In practice most uses of tags will be much simpler, but probably longer.

The sample generated code is annotated with comments that point to lifecycle steps presented in various sections. The notation is as follows:

- "Step T.x" = Annotated step x from "Lifecycle of Simple Tag Handlers" earlier in this Chapter.
- "Step C.x" = Annotated step x from "Creation of a JSP Fragment" earlier in this Chapter.
- "Step F.x" = Annotated step x from "Invocation of a JSP Fragment" earlier in this Chapter.

Sample Source Files

This section presents the sample source files in this scenario, from which the output files are generated.

Original JSP (my.jsp)

```jsp
<%@ taglib prefix="my" tagdir="/WEB-INF/tags" %>

<my:simpleTag x="10">
    <jsp:attribute name="y">20</jsp:attribute>
    <jsp:attribute name="nonfragment">
        Nonfragment Template Text
    </jsp:attribute>
    <jsp:attribute name="frag">
        Fragment Template Text ${var1}
    </jsp:attribute>
</my:simpleTag>
```
Original Tag File (/WEB-INF/tags/simpletag.tag)

<!-- /WEB-INF/tags/simpletag.tag -->
<%@ attribute name="x" %>
<%@ attribute name="y" %>
<%@ attribute name="nonfragment" %>
<%@ attribute name="frag" fragment="true" %>
<%@ variable name-given="var1" scope="AT_BEGIN" %>
<%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>

Some template text.
<c:set var="var1" value="${x+y}"/>
<jsp:invoke fragment="frag" varReader="var1"/>

Invoke the body:
<jsp:doBody/>

Sample Generated Files

This section presents sample output files that might be generated by a JSP compiler, from the source files presented in the previous section.

Helper class for JspFragment (JspFragmentBase.java)

```
public abstract class JspFragmentBase
    implements javax.servlet.jsp.tagext.JspFragment
{
    protected javax.servlet.jsp.JspContext jspContext;
    protected javax.servlet.jsp.tagext.JspTag parentTag;
    public void JspFragmentBase(
        javax.servlet.jsp.JspContext jspContext,
        javax.servlet.jsp.tagext.JspTag parentTag )
    {
        this.jspContext = jspContext;
        this.parentTag = parentTag;
    }
}
```
Relevant Portion of JSP Service Method

// Step T.1 - Initial creation
MySimpleTag _jsp_mySimpleTag = new MySimpleTag();
// Step T.2 - Set page context and parent (since parent is null,
// no need to call setParent() in this case)
_jsp_mySimpleTag.setJspContext( jspContext );
// Step T.3 - XML element attributes evaluated and set
_jsp.mySimpleTag.setX( "10" );
// Step T.4 - <jsp:attribute> elements evaluated and set
// - parameter y
// (using PageContext.pushBody() is one possible implementation -
// one limitation is that this code will only work for Servlet-based
out = ((PageContext)jspContext).pushBody();
out.write( "20" );
_jsp_mySimpleTag.setY(
    ((javax.servlet.jsp.tagext.BodyContent)out).getString() );
out = jspContext.popBody();
// - parameter nonfragment
// (using PageContext.pushBody() is one possible implementation -
// one limitation is that this code will only work for Servlet-based
// Note that trim is enabled by default, else we would have "\n   N
out = ((PageContext)jspContext).pushBody();
out.write( "Nonfragment Template Text" );
_jsp_mySimpleTag.setNonfragment(
    ((javax.servlet.jsp.tagext.BodyContent)out).getString() );
out = jspContext.popBody();
// - parameter frag
_jsp_mySimpleTag.setFrag( 1
// Step C.1 - New instance of fragment created
// Step C.2 - Store jspContext
// Step C.3 - Association with nearest enclosing Tag instance
new JspFragmentBase( jspContext, _jsp_mySimpleTag ) {  
    public void invoke( java.io.Writer writer ) {  
        javax.servlet.jsp.JspWriter out;  
        // Step F.1-F.3 done in tag file (see following example)
        // Step F.4 - If writer provided, push body:
        if( out == null ) {  
            out = this.jspContext.getOut();  
        } else {  
            out = this.jspContext.pushBody( writer );  
        }  
        // Step F.5 - Evaluate body of fragment:
        try {  
            out.write( "Fragment Template Text ");  
            out.write( jspContext.getExpressionEvaluator().evaluate(  
"${var1}",
                java.lang.String.class,
vResolver, fMapper, "my"));

finally {
    // Step F.6 - Restore value of JspContext.getOut()
    if (writer != null) {
        this.jspContext.popBody();
    }
}

    // Step F.7-F.9 done in tag file (see following example)
}

// Step T.5 - Determine and set body of the tag
// - body of tag
_jsp_mySimpleTag.setJspBody(
    // Step C.1 - New instance of fragment created
    // Step C.2 - Store jspContext
    // Step C.3 - Association with nearest enclosing Tag instance
    new JspFragmentBase(jspContext, _jsp_mySimpleTag) {
        public void invoke(java.io.Writer writer) {
            javax.servlet.jsp.JspWriter out;
            // Step F.1-F.3 done in tag file (see following example)
            // Step F.4 - If writer provided, push body:
            if (writer == null) {
                out = this.jspContext.getOut();
            } else {
                out = this.jspContext.pushBody(writer);
            }
            // Step F.5 - Evaluate body of fragment:
            try {
                out.write("Body of tag that defines an AT_BEGIN\n" +
                    " scripting variable "");
                out.write(jspContext.getExpressionEvaluator().evaluate("${var1}",
                    java.lang.String.class,
                    vResolver, fMapper, "my" ));
                out.write(" .\n");
            }
            finally {
                // Step F.6 - Restore value of JspContext.getOut()
                if (writer != null) {
                    this.jspContext.popBody();
                }
            }
            // Step F.7-F.9 done in tag file (see following example)
        }
    });
}
// Step T.6 - Invoke doTag
// Step T.7 occurs in the tag file (see following example)
// Step T.8 - doTag returns - page will catch SkipPageException.
_jsp_mySimpleTag.doTag();
// Step T.9 - Declare AT_BEGIN and AT_END scripting variables
String var1 = (String)jspContext.findAttribute( "var1" );

Generated Simple Tag Handler (MySimpleTag.java)

public class MySimpleTag
extends javax.servlet.jsp.tagext.SimpleTagSupport
{
    // Attributes:
    private String x;
    private String y;
    private String nonfragment;
    private javax.servlet.jsp.tagext.JspFragment frag;
    // Setters and getters for attributes:
    public void setX( String x ) {
        this.x = x;
    }
    public String getX() {
        return this.x;
    }
    public void setY( String y ) {
        this.y = y;
    }
    public String getY() {
        return this.y;
    }
    public void setNonfragment( String nonfragment ) {
        this.nonfragment = nonfragment;
    }
    public String getNonfragment() {
        return this.nonfragment;
    }
    public void setFrag(javax.servlet.jsp.tagext.JspFragment frag) {
        this.frag = frag;
    }
    public javax.servlet.jsp.tagext.JspFragment getFrag() {
        return this.frag;
    }

    protected JspContext jspContext;
    public void setJspContext( JspContext ctx ) {
        super.setJspContext( ctx );
        // Step T.2 - A JspContext wrapper is created.
        // (Implementation of wrapper not shown).
this.jspContext = new utils.JspContextWrapper( ctx );
}
public JspContext getJspContext() {
    // Step T.2 - Calling getJspContext() must return the
    // wrapped JspContext.
    return this.jspContext;
}
public void doTag() throws JspException {
    java.lang.Object jspValue;
    JspContext jspContext = getJspContext();
    JspContext _jsp_parentContext =
        SimpleTagSupport.this.getJspContext();
    try {
        javax.servlet.jsp.JspWriter out = jspContext.getOut();

        // Create page-scope attributes for each tag attribute:
        this.jspContext.setAttribute( "x", getX() );
        this.jspContext.setAttribute( "y", getY() );
        this.jspContext.setAttribute( "nonfragment", getNonfragm 
            this.jspContext.setAttribute( "frag", getFrag() ) ;

        // Synchronize AT_BEGIN variables from calling page
        if( (jspValue = _jsp_parentContext.getAttribute(
            "var1" )) != null )
        {
            jspContext.setAttribute( "var1", value );
        }
        else {
            jspContext.removeAttribute( "var1",
                JspContext.PAGE_SCOPE );
        }

        // Tag template text:
        out.write( "\n\n\n\n\n\n\n\n\n\nSome template text.\n" );

        // Invoke c:set - recognized tag handler from JSTL:
        jspContext.setAttribute( "var1",
            jspContext.getExpressionEvaluator().evaluate( 
            "${x+y}"
            java.lang.String.class,
            jspContext,
            prefixMap, functionMap, "my" ) );

        // Invoke the "frag" fragment:
        // Step F.1 - Set values of AT_BEGIN and NESTED variable
        // in calling page context.
        if( (jspValue = jspContext.getAttribute( "var1" )) != nu 
            _jsp_parentContext.setAttribute( "var1", value );
    }
else {
    _jsp_parentContext.removeAttribute( "var1",
        JspContext.PAGE_SCOPE );
}

// Step F.2 - varReader is specified, generate a writer.
java.io.Writer _jsp_sout = new java.io.StringWriter();

// Step F.3 - Invoke fragment with writer
getFrag().invoke( _jsp_sout );

// Step F.4 - F.6 occur in the fragment (see above)
// Step F.7 - fragment returns

// Step F.8 - varReader specified, so save to var
_jspContext.setAttribute( "var1", new StringReader( _jsp_sout.toString() ) );

// Step F.9 - Done!

out.write( "\n\nInvoke the body:\n" );

// Invoke the body of the tag:
// Step F.1 - Set values of AT_BEGIN and NESTED variable
// in calling page context.
if( (jspValue = jspContext.getAttribute( "var1" )) != null ) {
    _jsp_parentContext.setAttribute( "var1", value );
} else {
    _jsp_parentContext.removeAttribute( "var1",
        JspContext.PAGE_SCOPE );
}

// Step F.2 - varReader is not specified - does not apply

try {
    // Step F.3 - Invoke body, passing optional writer
    getJspBody().invoke( null );
} finally {
    // Steps F.4 - F.6 occur in the fragment (see above)
    // Step F.7 - fragment returns
}

// Step F.8 does not apply.
// Step F.9 - Done!

finally {
// Tag handlers generate code to synchronize AT-BEGIN with calling page, regardless of whether an error occurs.
if (jspValue = jspContext.getAttribute("var1")) != null {
    _jsp_parentContext.setAttribute("var1", value);
} else {
    _jsp_parentContext.removeAttribute("var1",
        JspContext.PAGE_SCOPE);
}
9. Translation-time Classes

The following classes are used at translation time.

**Tag mapping, Tag name**

A taglib directive introduces a tag library and associates a prefix to it. The TLD associated with the library associates Tag handler classes (plus other information) with tag names. This information is used to associate a Tag class, a prefix, and a name with each custom action element appearing in a JSP page.

At execution time the implementation of a JSP page will use an available Tag instance with the appropriate property settings and then follow the protocol described by the interfaces Tag, IterationTag, BodyTag, SimpleTag, and TryCatchFinally. The implementation guarantees that all tag handler instances are initialized and all are released, but the implementation can assume that previous settings are preserved by a tag handler, to reduce run-time costs.

**Scripting Variables**

JSP supports scripting variables that can be declared within a scriptlet and can be used in another. JSP actions also can be used to define scripting variables so they can used in scripting elements, or in other actions. This is very useful in some cases; for example, the `jsp:useBean` standard action may define an object which can later be used through a scripting variable.

In some cases the information on scripting variables can be described directly into the TLD using elements. A special case is typical interpretation of the "id" attribute. In other cases the logic that decides whether an action instance will define a scripting variable may be quite complex and the name of a `TagExtraInfo` class is instead given in the TLD. The `getVariableInfo` method of this class is used at translation time to obtain information on each variable that will be created at request time when this action is executed. The method is passed a `TagData` instance
that contains the translation-time attribute values.

Validation

The TLD file contains several pieces of information that is used to do syntactic validation at translation-time. It also contains two extensible validation mechanisms: a TagLibraryValidator class can be used to validate a complete JSP page, and a TagExtraInfo class can be used to validate a specific action. In some cases, additional request-time validation will be done dynamically within the methods in the Tag instance. If an error is discovered, an instance of JspTagException can be thrown. If uncaught, this object will invoke the errorpage mechanism of JSP.

The TagLibraryValidator is an addition to the JSP 1.2 specification and is very open ended, being strictly more powerful than the TagExtraInfo mechanism. A JSP page is presented via the PageData object, which abstracts the XML view of the JSP page.

A PageData instance will provides an InputStream (read-only) on the page. Later specifications may add other views on the page (DOM, SAX, JDOM are all candidates), for now these views can be generated from the InputStream and perhaps can be cached for improved performance (recall the view of the page is just read-only).

As of JSP 2.0, the JSP container must support a jsp:id attribute to provide higher quality validation errors. The container will track the JSP pages as passed to the container, and will assign to each element a unique "id", which is passed as the value of the jsp:id attribute. Each XML element in the XML view will be extended with this attribute. The TagLibraryValidator can use the attribute in one or more ValidationMessage objects. The container then, in turn, can use these values to provide more precise information on the location of an error.

The prefix for the id attribute need not be "jsp" but it must map to the namespace http://java.sun.com/JSP/Page. In the case where the user has redefined the jsp prefix, an alternative prefix must be used by the container.
Validation Details

In detail, validation is done as follows:

**First**, the JSP page is parsed using the information in the TLD. At this stage valid mandatory and optional attributes are checked.

**Second**, for each unique tag library in the page as determined by the tag library URI, and in the lexical order in which they appear, their associated validator class (if any) is invoked. This involves several substeps.

The first substep is to obtain an initialized validator instance by either:

- construct a new instance and invoke setInitParameters() on it, or
- obtain an existing instance that is not being used, invoke release() on it, and then invoke setInitParameters() on it, or
- locate an existing instance that is not being used on which the desired setInitParameters() has already been invoked

The class name is as indicated in the `<validator-class>` element, and the Map passed through setInitParameters() is as described in the `<init-params>` element. All TagLibraryValidator classes are supposed to keep their initParameters until new ones are set, or until release() is invoked on them.

The second substep is to perform the actual validation. This is done by invoking the validate() method with a prefix, uri, and PageData that correspond to the taglib directive instance being validated and the PageData representing the page. In the case where a single URI is mapped to more than one prefix, the prefix of the first URI must be used.

The last substep is to invoke the release() method on the validator tag when it is no longer needed. This method releases all resources.

**Finally**, after checking all the tag library validator classes, the TagExtraInfo classes for all tags will be consulted by invoking their validate method. The order of invocation of this methods is undefined.
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.servlet.jsp.tagext

Package Hierarchies:
  All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.servlet.jsp.tagext.FunctionInfo`
  - `javax.servlet.jsp.tagext.JspFragment`
  - `javax.servlet.jsp.tagext.PageData`
  - `javax.servlet.jsp.tagext.SimpleTagSupport` (implements `javax.servlet.jsp.tagext.SimpleTag`)
  - `javax.servlet.jsp.tagext.TagAdapter` (implements `javax.servlet.jsp.tagext.Tag`)
  - `javax.servlet.jsp.tagext.TagAttributeInfo`
  - `javax.servlet.jsp.tagext.TagData` (implements `java.lang.Cloneable`)
  - `javax.servlet.jsp.tagext.TagExtraInfo`
  - `javax.servlet.jsp.tagext.TagFileInfo`
  - `javax.servlet.jsp.tagext.TagInfo`
  - `javax.servlet.jsp.tagext.TagLibraryInfo`
  - `javax.servlet.jsp.tagext.TagLibraryValidator`
  - `javax.servlet.jsp.tagext.TagSupport` (implements `javax.servlet.jsp.tagext.IterationTag`, `java.io.Serializable`)
    - `javax.servlet.jsp.tagext.TagVariableInfo`
    - `javax.servlet.jsp.tagext.ValidationMessage`
    - `javax.servlet.jsp.tagext.VariableInfo`
  - `javax.servlet.jsp.JspWriter`
    - `javax.servlet.jsp.tagext.BodyContent`
Interface Hierarchy

- `javax.servlet.jsp.tagext.DynamicAttributes`
- `javax.servlet.jsp.tagext.JspIdConsumer`
- `javax.servlet.jsp.tagext.JspTag`
  - `javax.servlet.jsp.tagext.SimpleTag`
  - `javax.servlet.jsp.tagext.Tag`
    - `javax.servlet.jsp.tagext.IterationTag`
    - `javax.servlet.jsp.tagext.BodyTag`
- `javax.servlet.jsp.tagext.TryCatchFinally`
Package `javax.management.j2ee.statistics`

Provides the standard interfaces for accessing performance data from J2EE managed objects Package Specification [JSR 77, J2EE Management](https://xmlns.jcp.org/j2ee/7/management/) Related Documentation For overviews, tutorials, examples, guides, and tool documentation, please see: [J2EE Tools](https://java.sun.com/products/j2ee) See:

**Description**

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>BoundaryStatistic</strong></td>
<td>Specifies standard measurements of the upper and lower limits of the value of an attribute.</td>
</tr>
<tr>
<td><strong>BoundedRangeStatistic</strong></td>
<td>The BoundedRangeStatistic model aggregates the attributes of RangeStatistic and BoundaryStatistic and provides standard measurements of a range that has fixed limits.</td>
</tr>
<tr>
<td><strong>CountStatistic</strong></td>
<td>Specifies standard count measurements.</td>
</tr>
<tr>
<td><strong>EJBStats</strong></td>
<td>Specifies statistics provided by all EJB component types.</td>
</tr>
<tr>
<td><strong>EntityBeanStats</strong></td>
<td>Specifies statistics provided by entity beans.</td>
</tr>
<tr>
<td><strong>JavaMailStats</strong></td>
<td>Specifies the statistics provided by a JavaMail resource.</td>
</tr>
<tr>
<td><strong>JCAConnectionPoolStats</strong></td>
<td>Specifies the statistics provided by a JCA Connection Pool</td>
</tr>
<tr>
<td><strong>JCAConnectionStats</strong></td>
<td>Specifies the statistics provided by a JCA connection</td>
</tr>
<tr>
<td><strong>JCAStats</strong></td>
<td>Specifies statistics provided by a JCA resource</td>
</tr>
<tr>
<td><strong>JDBCCConnectionPoolStats</strong></td>
<td>Specifies the statistics provided by a JDBC connection pool.</td>
</tr>
<tr>
<td><strong>SAXConnectionStats</strong></td>
<td>Specifies the statistics provided by all SAX connections.</td>
</tr>
<tr>
<td><strong>JDBCConnectionStats</strong></td>
<td>(pooled and non-pooled) JDBC connections.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td><strong>JDBCStats</strong></td>
<td>Statistics provided by a JDBC resource</td>
</tr>
<tr>
<td><strong>JMSConnectionStats</strong></td>
<td>Specifies the statistics provided by a JMS connection</td>
</tr>
<tr>
<td><strong>JMSConsumerStats</strong></td>
<td>Specifies the statistics provided by a JMS message consumer</td>
</tr>
<tr>
<td><strong>JMSEndpointStats</strong></td>
<td>Specifies the statistics provided by a JMS message producer or a JMS message consumer.</td>
</tr>
<tr>
<td><strong>JMSProducerStats</strong></td>
<td>Specifies the statistics provided by a JMS message producer</td>
</tr>
<tr>
<td><strong>JMSSessionStats</strong></td>
<td>Specifies the statistics provided by a JMS session.</td>
</tr>
<tr>
<td><strong>JMSStats</strong></td>
<td>Specifies the statistics provided by a JMS Resource</td>
</tr>
<tr>
<td><strong>JTAStats</strong></td>
<td>Specifies the statistics provided by a JTA resource.</td>
</tr>
<tr>
<td><strong>JVMStats</strong></td>
<td>Specifies the statistics provided by a Java VM.</td>
</tr>
<tr>
<td><strong>MessageDrivenBeanStats</strong></td>
<td>Specifies the statistics provided by a message driven bean.</td>
</tr>
<tr>
<td><strong>RangeStatistic</strong></td>
<td>Specifies standard measurements of the lowest and highest values an attribute has held as well as its current value.</td>
</tr>
<tr>
<td><strong>ServletStats</strong></td>
<td>Specifies the statistics provided by a Servlet.</td>
</tr>
<tr>
<td><strong>SessionBeanStats</strong></td>
<td>Specifies the statistics provided by session beans of both stateful and stateless types.</td>
</tr>
<tr>
<td><strong>StatefulSessionBeanStats</strong></td>
<td>Specifies the statistics provided by a stateful session bean.</td>
</tr>
<tr>
<td><strong>StatelessSessionBeanStats</strong></td>
<td>Specifies the statistics provided by a stateless session bean.</td>
</tr>
<tr>
<td><strong>Statistic</strong></td>
<td>The Statistic model and its sub-models specify the data models which are required to be used to provide the performance data described by the specific attributes in the Stats models.</td>
</tr>
<tr>
<td><strong>Stats</strong></td>
<td>The Stats model and its submodels specify performance data attributes for each of the specific managed object types.</td>
</tr>
<tr>
<td><strong>TimeStatistic</strong></td>
<td>Specifies standard timing measurements.</td>
</tr>
<tr>
<td><strong>URLStats</strong></td>
<td>Specifies the statistics provided by a URL resource.</td>
</tr>
</tbody>
</table>
Package javax.management.j2ee.statistics

Description

Provides the standard interfaces for accessing performance data from J2EE managed objects
Package Specification

- [JSR 77, J2EE Management]
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools
Hierarchy For Package
javax.management.j2ee.statistics

Package Hierarchies:
All Packages
Interface Hierarchy

- javax.management.j2ee.statistics.JavaMailStats
- javax.management.j2ee.statistics.Statistic
  - javax.management.j2ee.statistics.BoundaryStatistic
  - javax.management.j2ee.statistics.BoundedRangeStatistic
    (also extends
     javax.management.j2ee.statistics.RangeStatistic)
  - javax.management.j2ee.statistics.CountStatistic
- javax.management.j2ee.statistics.RangeStatistic
  - javax.management.j2ee.statistics.BoundedRangeStatistic
    (also extends
     javax.management.j2ee.statistics.BoundaryStatistic)
  - javax.management.j2ee.statistics.TimeStatistic
- javax.management.j2ee.statistics.Stats
  - javax.management.j2ee.statistics.EJBStats
    - javax.management.j2ee.statistics.EntityBeanStats
    - javax.management.j2ee.statistics.MessageDrivenBeanStat
    - javax.management.j2ee.statistics.SessionBeanStats
      - javax.management.j2ee.statistics.StatefulSessionBe
      - javax.management.j2ee.statistics.StatelessSessionBe
  - javax.management.j2ee.statistics.JCAConnectionStats
    - javax.management.j2ee.statistics.JCAConnectionPoolStat
  - javax.management.j2ee.statistics.JCASTats
  - javax.management.j2ee.statistics.JDBCConnectionStats
    - javax.management.j2ee.statistics.JDBCConnectionPoolStat
  - javax.management.j2ee.statistics.JDBCStats
  - javax.management.j2ee.statistics.JMSSessionStats
  - javax.management.j2ee.statistics.JMSStats
  - javax.management.j2ee.statistics.JMSEndpointStats
    - javax.management.j2ee.statistics.JMSConsumerStats
    - javax.management.j2ee.statistics.JMSPproducerStats
  - javax.management.j2ee.statistics.JMSSessionStats
  - javax.management.j2ee.statistics.JMSSStats
  - javax.management.j2ee.statistics.JTATats
  - javax.management.j2ee.statistics.JVMStats
  - javax.management.j2ee.statistics.ServletStats
  - javax.management.j2ee.statistics.URLStats
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.xml.registry`

This package and its sub-packages describe the API classes and interfaces for the JAXR API.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>BulkResponse</code></td>
<td>Contains the response of a method in the API that performs a bulk operation and returns a bulk response.</td>
</tr>
<tr>
<td><code>BusinessLifeCycleManager</code></td>
<td>The <code>BusinessLifeCycleManager</code> interface, which is exposed by the Registry Service, implements the life cycle management functionality of the Registry as part of a business level API.</td>
</tr>
<tr>
<td><code>BusinessQueryManager</code></td>
<td>The <code>BusinessQueryManager</code> interface, which is exposed by the Registry Service, implements the business style query interface.</td>
</tr>
<tr>
<td><code>CapabilityProfile</code></td>
<td>Provides information about the capabilities of a JAXR provider.</td>
</tr>
<tr>
<td><code>Connection</code></td>
<td>This class represents a connection between a JAXR client and a JAXR provider.</td>
</tr>
<tr>
<td><code>DeclarativeQueryManager</code></td>
<td>This interface provides the ability to execute declarative queries (e.g.</td>
</tr>
<tr>
<td><code>FederatedConnection</code></td>
<td>Represents a single logical connection to a federation or group of registry providers.</td>
</tr>
<tr>
<td><code>FindQualifier</code></td>
<td>FindQualifier provides various constants that identify options that effect find method behavior.</td>
</tr>
<tr>
<td><code>JAXRResponse</code></td>
<td>A JAXR requests' response.</td>
</tr>
</tbody>
</table>

The `LifeCycleManager` interface is the
<table>
<thead>
<tr>
<th><strong>LifeCycleManager</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>main interface in the API for managing life cycle operations on objects defined by the information model.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>Query</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>The Query interface encapsulates a query in a declarative query language.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>QueryManager</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the common base interface for all QueryManagers in the API.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>RegistryService</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the principal interface implemented by a JAXR provider.</td>
</tr>
</tbody>
</table>

---

### Class Summary

<table>
<thead>
<tr>
<th><strong>ConnectionFactory</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the abstract base class for factory classes for creating a JAXR connection.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th><strong>DeleteException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A RegistryException that occurs during a delete action.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>FindException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A RegistryException that occurs during a find action.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>InvalidRequestException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This exception is thrown when a JAXR client attempts to invoke an API method that is not valid for some reason.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>JAXRException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Signals that a JAXR exception has occurred.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>RegistryException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>This is the common base class for all Exceptions that are detected on the registry provider side rather than the JAXR client side.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>SaveException</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>A RegistryException that occurs during a save action.</td>
</tr>
</tbody>
</table>

---
<table>
<thead>
<tr>
<th><strong>UnexpectedObjectException</strong></th>
<th>JAXR provider finds a Object that is out-of-place or of the wrong type within the context of a user request.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UnsupportedCapabilityException</strong></td>
<td>This exception must be thrown when a JAXR client attempts to invoke an API method that is not supported by the capability profile that is supported by the JAXR provider.</td>
</tr>
</tbody>
</table>
Package `javax.xml.registry` Description

This package and its sub-packages describe the API classes and interfaces for the JAXR API. In this package are the high-level interfaces and classes that a client uses to interact with a registry via a JAXR provider.

Figure 1 shows the primary interfaces in the JAXR API.

![Figure 1. Primary Interfaces in the JAXR API](image)

Figure 2 shows the response and exception classes in the JAXR API.

![Figure 2. Response and Exception Classes in the JAXR API](image)
Hierarchy For Package javax.xml.registry

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.registry.**ConnectionFactory**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.xml.registry.**JAXRException** (implements javax.xml.registry.**JAXRResponse**)
        - javax.xml.registry.**InvalidRequestException**
        - javax.xml.registry.**RegistryException**
          - javax.xml.registry.**DeleteException**
          - javax.xml.registry.**FindException**
          - javax.xml.registry.**SaveException**
        - javax.xml.registry.**UnexpectedObjectException**
        - javax.xml.registry.**UnsupportedCapabilityException**
Interface Hierarchy

- javax.xml.registry.CapabilityProfile
- javax.xml.registry.Connection
  - javax.xml.registry.FederatedConnection
- javax.xml.registry.FindQualifier
- javax.xml.registry.JAXRResponse
  - javax.xml.registry.BulkResponse
- javax.xml.registry.LifeCycleManager
  - javax.xml.registry.BusinessLifeCycleManager
- javax.xml.registry.Query
- javax.xml.registry.QueryManager
  - javax.xml.registry.BusinessQueryManager
  - javax.xml.registry.DeclarativeQueryManager
- javax.xml.registry.RegistryService
Package javax.mail.util

Utility classes.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ByteArrayDataSource</strong></td>
</tr>
<tr>
<td><strong>SharedByteArrayInputStream</strong></td>
</tr>
<tr>
<td><strong>SharedFileInputStream</strong></td>
</tr>
</tbody>
</table>
Package javax.mail.util Description

Utility classes. This package specifies utility classes that are useful with other JavaMail APIs.
Hierarchy For Package javax.mail.util

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.mail.util.**ByteArrayDataSource** (implements javax.activation.**DataSource**)
  - java.io.**InputStream** (implements java.io.**Closeable**)
    - java.io.**ByteArrayInputStream**
    - javax.mail.util.**SharedByteArrayInputStream** (implements javax.mail.internet.**SharedInputStream**)
    - java.io.**FilterInputStream**
      - java.io.**BufferedInputStream**
        - javax.mail.util.**SharedFileInputStream** (implements javax.mail.internet.**SharedInputStream**)

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.jms

The Java Message Service (JMS) API provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages.

See:   Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BytesMessage</td>
<td>A BytesMessage object is used to send a message containing a stream of uninterpreted bytes.</td>
</tr>
<tr>
<td>Connection</td>
<td>A Connection object is a client's active connection to its JMS provider.</td>
</tr>
<tr>
<td>ConnectionConsumer</td>
<td>For application servers, Connection objects provide a special facility for creating a ConnectionConsumer (optional).</td>
</tr>
<tr>
<td>ConnectionFactory</td>
<td>A ConnectionFactory object encapsulates a set of connection configuration parameters that has been defined by an administrator.</td>
</tr>
<tr>
<td>ConnectionMetaData</td>
<td>A ConnectionMetaData object provides information describing the Connection object.</td>
</tr>
<tr>
<td>DeliveryMode</td>
<td>The delivery modes supported by the JMS API are PERSISTENT and NON_PERSISTENT.</td>
</tr>
<tr>
<td>Destination</td>
<td>A Destination object encapsulates a provider-specific address.</td>
</tr>
<tr>
<td>ExceptionListener</td>
<td>If a JMS provider detects a serious problem with a Connection object, it informs the Connection object's ExceptionListener, if one has been registered.</td>
</tr>
<tr>
<td><strong>MapMessage</strong></td>
<td>A <code>MapMessage</code> object is used to send a set of name-value pairs.</td>
</tr>
<tr>
<td>----------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Message</strong></td>
<td>The <code>Message</code> interface is the root interface of all JMS messages.</td>
</tr>
<tr>
<td><strong>MessageConsumer</strong></td>
<td>A client uses a <code>MessageConsumer</code> object to receive messages from a destination.</td>
</tr>
<tr>
<td><strong>MessageListener</strong></td>
<td>A <code>MessageListener</code> object is used to receive asynchronously delivered messages.</td>
</tr>
<tr>
<td><strong>MessageProducer</strong></td>
<td>A client uses a <code>MessageProducer</code> object to send messages to a destination.</td>
</tr>
<tr>
<td><strong>ObjectMessage</strong></td>
<td>An <code>ObjectMessage</code> object is used to send a message that contains a serializable object in the Java programming language (&quot;Java object&quot;).</td>
</tr>
<tr>
<td><strong>Queue</strong></td>
<td>A <code>Queue</code> object encapsulates a provider-specific queue name.</td>
</tr>
<tr>
<td><strong>QueueBrowser</strong></td>
<td>A client uses a <code>QueueBrowser</code> object to look at messages on a queue without removing them.</td>
</tr>
<tr>
<td><strong>QueueConnection</strong></td>
<td>A <code>QueueConnection</code> object is an active connection to a point-to-point JMS provider.</td>
</tr>
<tr>
<td><strong>QueueConnectionFactory</strong></td>
<td>A client uses a <code>QueueConnectionFactory</code> object to create <code>QueueConnection</code> objects with a point-to-point JMS provider.</td>
</tr>
<tr>
<td><strong>QueueReceiver</strong></td>
<td>A client uses a <code>QueueReceiver</code> object to receive messages that have been delivered to a queue.</td>
</tr>
<tr>
<td><strong>QueueSender</strong></td>
<td>A client uses a <code>QueueSender</code> object to send messages to a queue.</td>
</tr>
<tr>
<td><strong>QueueSession</strong></td>
<td>A <code>QueueSession</code> object provides methods for creating <code>QueueReceiver</code>, <code>QueueSender</code>, <code>QueueBrowser</code>, and <code>TemporaryQueue</code> objects.</td>
</tr>
<tr>
<td><strong>ServerSession</strong></td>
<td>A <em>ServerSession</em> object is an application server object that is used by a server to associate a thread with a JMS session (optional).</td>
</tr>
<tr>
<td><strong>ServerSessionPool</strong></td>
<td>A <em>ServerSessionPool</em> object is an object implemented by an application server to provide a pool of <em>ServerSession</em> objects for processing the messages of a <em>ConnectionConsumer</em> (optional).</td>
</tr>
<tr>
<td><strong>Session</strong></td>
<td>A <em>Session</em> object is a single-threaded context for producing and consuming messages.</td>
</tr>
<tr>
<td><strong>StreamMessage</strong></td>
<td>A <em>StreamMessage</em> object is used to send a stream of primitive types in the Java programming language.</td>
</tr>
<tr>
<td><strong>TemporaryQueue</strong></td>
<td>A <em>TemporaryQueue</em> object is a unique <em>Queue</em> object created for the duration of a <em>Connection</em>.</td>
</tr>
<tr>
<td><strong>TemporaryTopic</strong></td>
<td>A <em>TemporaryTopic</em> object is a unique <em>Topic</em> object created for the duration of a <em>Connection</em>.</td>
</tr>
<tr>
<td><strong>TextMessage</strong></td>
<td>A <em>TextMessage</em> object is used to send a message containing a <code>java.lang.String</code>.</td>
</tr>
<tr>
<td><strong>Topic</strong></td>
<td>A <em>Topic</em> object encapsulates a provider-specific topic name.</td>
</tr>
<tr>
<td><strong>TopicConnection</strong></td>
<td>A <em>TopicConnection</em> object is an active connection to a publish/subscribe JMS provider.</td>
</tr>
<tr>
<td><strong>TopicConnectionFactory</strong></td>
<td>A client uses a <em>TopicConnectionFactory</em> object to create <em>TopicConnection</em> objects with a publish/subscribe JMS provider.</td>
</tr>
<tr>
<td><strong>TopicPublisher</strong></td>
<td>A client uses a <em>TopicPublisher</em> object to publish messages on a topic.</td>
</tr>
<tr>
<td></td>
<td>A <em>TopicSession</em> object provides methods for creating <em>TopicPublisher</em>,</td>
</tr>
<tr>
<td><strong>TopicSession</strong></td>
<td>TopicSubscriber, and TemporaryTopic objects.</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>TopicSubscriber</strong></td>
<td>A client uses a TopicSubscriber object to receive messages that have been published to a topic.</td>
</tr>
<tr>
<td><strong>XAConnection</strong></td>
<td>The XAConnection interface extends the capability of Connection by providing an XASession (optional).</td>
</tr>
<tr>
<td><strong>XAConnectionFactory</strong></td>
<td>The XAConnectionFactory interface is a base interface for the XAQueueConnectionFactory and XATopicConnectionFactory interfaces.</td>
</tr>
<tr>
<td><strong>XAQueueConnection</strong></td>
<td>An XAQueueConnection provides the same create options as QueueConnection (optional).</td>
</tr>
<tr>
<td><strong>XAQueueConnectionFactory</strong></td>
<td>An XAQueueConnectionFactory provides the same create options as a QueueConnectionFactory (optional).</td>
</tr>
<tr>
<td><strong>XAQueueSession</strong></td>
<td>An XAQueueSession provides a regular QueueSession, which can be used to create QueueReceiver, QueueSender, and QueueBrowser objects (optional).</td>
</tr>
<tr>
<td><strong>XASession</strong></td>
<td>The XASession interface extends the capability of Session by adding access to a JMS provider's support for the Java Transaction API (JTA) (optional).</td>
</tr>
<tr>
<td><strong>XATopicConnection</strong></td>
<td>An XATopicConnection provides the same create options as TopicConnection (optional).</td>
</tr>
<tr>
<td><strong>XATopicConnectionFactory</strong></td>
<td>An XATopicConnectionFactory provides the same create options as a TopicConnectionFactory (optional).</td>
</tr>
<tr>
<td><strong>XATopicSession</strong></td>
<td>An XATopicSession provides a regular TopicSession.</td>
</tr>
</tbody>
</table>
### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>QueueRequestor</td>
<td>The QueueRequestor helper class simplifies making service requests.</td>
</tr>
<tr>
<td>TopicRequestor</td>
<td>The TopicRequestor helper class simplifies making service requests.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IllegalStateException</td>
<td>This exception is thrown when a method is invoked at an illegal or inappropriate time or if the provider is not in an appropriate state for the requested operation.</td>
</tr>
<tr>
<td>InvalidClientIDException</td>
<td>This exception must be thrown when a client attempts to set a connection's client ID to a value that is rejected by a provider.</td>
</tr>
<tr>
<td>InvalidDestinationException</td>
<td>This exception must be thrown when a destination either is not understood by a provider or is no longer valid.</td>
</tr>
<tr>
<td>InvalidSelectorException</td>
<td>This exception must be thrown when a JMS client attempts to give a provider a message selector with invalid syntax.</td>
</tr>
<tr>
<td>JMSException</td>
<td>This is the root class of all JMS API exceptions.</td>
</tr>
<tr>
<td>JMSSecurityException</td>
<td>This exception must be thrown when a provider rejects a user name/password submitted by a client.</td>
</tr>
<tr>
<td>MessageEOFException</td>
<td>This exception must be thrown when an unexpected end of stream has been reached when a</td>
</tr>
<tr>
<td>Exception Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>StreamMessage Or BytesMessage Is being read.</code></td>
<td>This exception must be thrown when a JMS client attempts to use a data type not supported by a message or attempts to read data in a message as the wrong type.</td>
</tr>
<tr>
<td><code>MessageFormatException</code></td>
<td>This exception must be thrown when a JMS client attempts to use a data type not supported by a message or attempts to read data in a message as the wrong type.</td>
</tr>
<tr>
<td><code>MessageNotReadableException</code></td>
<td>This exception must be thrown when a JMS client attempts to read a write-only message.</td>
</tr>
<tr>
<td><code>MessageNotWriteableException</code></td>
<td>This exception must be thrown when a JMS client attempts to write to a read-only message.</td>
</tr>
<tr>
<td><code>ResourceAllocationException</code></td>
<td>This exception is thrown when a provider is unable to allocate the resources required by a method.</td>
</tr>
<tr>
<td><code>TransactionInProgressException</code></td>
<td>This exception is thrown when an operation is invalid because a transaction is in progress.</td>
</tr>
<tr>
<td><code>TransactionRolledBackException</code></td>
<td>This exception must be thrown when a call to <code>Session.commit</code> results in a rollback of the current transaction.</td>
</tr>
</tbody>
</table>
Package javax.jms Description

The Java Message Service (JMS) API provides a common way for Java programs to create, send, receive and read an enterprise messaging system's messages.

JMS Applications

A JMS application is composed of the following parts:

- JMS Provider - a messaging system that implements the JMS API in addition to the other administrative and control functionality required of a full-featured messaging product
- JMS Clients - the Java language programs that send and receive messages
- Messages - objects that are used to communicate information between the clients of an application
- Administered Objects - provider-specific objects that clients look up and use to interact portably with a JMS provider
- Non-JMS Clients - clients that use a message system's native client API instead of the JMS API. If the application predated the availability of the JMS API, it is likely that it will include both JMS clients and non-JMS clients.

Administration

JMS providers differ significantly in their implementations of underlying messaging technology. There are also major differences in how a JMS provider's system is installed and administered.

For JMS clients to be portable, they must be isolated from these proprietary aspects of a provider. This is done by defining JMS administered objects that are created and customized by a provider's administrator and later used by clients. The client uses them through JMS interfaces that are portable. The administrator creates them using provider-specific facilities.
There are two types of JMS administered objects:

- ConnectionFactory - the object a client uses to create a connection with a JMS provider
- Destination - the object a client uses to specify the destination of messages it is sending and the source of messages it receives

Administered objects are placed in a Java Naming and Directory Interface™ (JNDI) namespace by an administrator. A JMS client typically notes in its documentation the JMS administered objects it requires and how the JNDI names of these objects should be provided to it.

**Two Messaging Styles**

The JMS specification defines two styles of messaging: the point-to-point (PTP) or the publish-and-subscribe (Pub/Sub). These styles can be combined in a single application, or a given application can use just one of these styles.

The JMS API defines these two styles because they represent two of the dominant approaches to messaging currently in use. While the domains have many similarities, they also have some differences. JMS provides a unified programming interface to allow the client programmer to easily send and receive message using either domain, but the client programmer must also be aware of the differences between the domains. The key differences relate to how message persistence is handled, and the meaning of certain message attributes.

**JMS Interfaces**

When programming an application client, the programmer may either program using the domain specific interfaces, or may use the common interfaces. The key interfaces are listed in the table below. The preferred model is to use the common interfaces. The advantage to using the common interfaces is that both point-to-point and pub/sub tasks can be combined in one session, allowing transactions to operate over both domains.

In earlier versions of JMS, there were separate class hierarchies for the
pub/sub and point-to-point programming models that had to be used. These class hierarchies are retained to support backward compatibility with earlier versions of the JMS API, but client developers are encouraged to use the common interfaces.

### Relationship of PTP and Pub/Sub interfaces

<table>
<thead>
<tr>
<th>JMS Common</th>
<th>PTP Domain</th>
<th>Pub/Sub Domain</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectionFactory</td>
<td>QueueConnectionFactory</td>
<td>TopicConnectionFactory</td>
</tr>
<tr>
<td>Connection</td>
<td>QueueConnection</td>
<td>TopicConnection</td>
</tr>
<tr>
<td>Destination</td>
<td>Queue</td>
<td>Topic</td>
</tr>
<tr>
<td>Session</td>
<td>QueueSession</td>
<td>TopicSession</td>
</tr>
<tr>
<td>MessageProducer</td>
<td>QueueSender</td>
<td>TopicPublisher</td>
</tr>
<tr>
<td>MessageConsumer</td>
<td>QueueReceiver</td>
<td>TopicSubscriber</td>
</tr>
</tbody>
</table>

The following provides a brief definition of these JMS concepts. See the PTP and Pub/Sub chapters of the JMS specification for more information.

- ConnectionFactory - an administered object used by a client to create a Connection
- Connection - an active connection to a JMS provider
- Destination - an administered object that encapsulates the identity of a message destination
- Session - a single-threaded context for sending and receiving messages
- MessageProducer - an object created by a Session that is used for sending messages to a destination
- MessageConsumer - an object created by a Session that is used for receiving messages sent to a destination

The term *consume* is used in this document to mean the receipt of a message by a JMS client; that is, a JMS provider has received a message and has given it to its client. Since the JMS API supports both synchronous and asynchronous receipt of messages, the term *consume* is used when there is no need to make a distinction between them.

The term *produce* is used as the most general term for sending a message. It means giving a message to a JMS provider for delivery to a
Developing a JMS Application

Broadly speaking, a JMS application is one or more JMS clients that exchange messages. The application may also involve non-JMS clients; however, these clients use the JMS provider's native API in place of the JMS API.

A JMS application can be architected and deployed as a unit. In many cases, JMS clients are added incrementally to an existing application.

The message definitions used by an application may originate with JMS, or they may have been defined by the non-JMS part of the application.

Developing a JMS Client

A typical JMS client executes the following setup procedure:

- Use JNDI to find a ConnectionFactory object
- Use JNDI to find one or more Destination objects
- Use the ConnectionFactory to create a JMS Connection. At this point, message delivery is inhibited
- Use the Connection to create one or more JMS Sessions
- Use a Session and the Destinations to create the MessageProducers and MessageConsumers needed
- Start message delivery for the Connection. Messages will be delivered to MessageConsumers

At this point a client has the basic setup needed to produce and consume messages.

Package Specification

Java Message Service Specification - Version 1.1

Related Documentation

Java Message Service Tutorial
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.jms

Package Hierarchies:
All Packages

____________________________________________________________________
Class Hierarchy

- java.lang.**Object**
  - javax.jms.**QueueRequestor**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.jms.**JMSException**
        - javax.jms.**IllegalStateException**
        - javax.jms.**InvalidClientIDException**
        - javax.jms.**InvalidDestinationException**
        - javax.jms.**InvalidSelectorException**
        - javax.jms.**JMSSecurityException**
        - javax.jms.**MessageEOFException**
        - javax.jms.**MessageFormatException**
        - javax.jms.**MessageNotReadableException**
        - javax.jms.**MessageNotWriteableException**
        - javax.jms.**ResourceAllocationException**
        - javax.jms.**TransactionInProgressException**
        - javax.jms.**TransactionRolledBackException**
  - javax.jms.**TopicRequestor**
Interface Hierarchy

- `javax.jms.Connection`
  - `javax.jms.QueueConnection`
    - `javax.jms.XAQueueConnection` (also extends `javax.jms.XAConnection`)
  - `javax.jms.TopicConnection`
    - `javax.jms.XATopicConnection` (also extends `javax.jms.XAConnection`)
  - `javax.jms.XAConnection`
    - `javax.jms.XAQueueConnection` (also extends `javax.jms.QueueConnection`)
    - `javax.jms.XATopicConnection` (also extends `javax.jms.TopicConnection`)
- `javax.jms.ConnectionConsumer`
- `javax.jms.ConnectionFactory`
  - `javax.jms.QueueConnectionFactory`
    - `javax.jms.XAQueueConnectionFactory` (also extends `javax.jms.XAConnectionFactory`)
  - `javax.jms.TopicConnectionFactory`
    - `javax.jms.XATopicConnectionFactory` (also extends `javax.jms.XAConnectionFactory`)
- `javax.jms.ConnectionMetaData`
- `javax.jms.DeliveryMode`
- `javax.jms.Destination`
  - `javax.jms.Queue`
    - `javax.jms.TemporaryQueue`
  - `javax.jms.Topic`
    - `javax.jms.TemporaryTopic`
- `javax.jms.ExceptionListener`
- `javax.jms.Message`
  - `javax.jms.BytesMessage`
  - `javax.jms.MapMessage`
  - `javax.jms.ObjectMessage`
  - `javax.jms.StreamMessage`
  - `javax.jms.TextMessage`
- `javax.jms.MessageConsumer`
javax.jms.QueueReceiver
javax.jms.TopicSubscriber
javax.jms.MessageListener
javax.jms.MessageProducer
javax.jms.QueueSender
javax.jms.TopicPublisher
javax.jms.QueueBrowser
java.langRunnable
javax.jms.Session
javax.jms.QueueSession
javax.jms.TopicSession
javax.jms.XASession
javax.jms.XAQueueSession
javax.jms.XATopicSession
javax.jms.ServerSession
javax.jms.ServerSessionPool
javax.jms.XAConnectionFactory
javax.jms.XAQueueConnectionFactory (also extends javax.jms.QueueConnectionFactory)
javax.jms.XATopicConnectionFactory (also extends javax.jms.TopicConnectionFactory)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
**Package javax.xml.rpc**

This package contains the core JAX-RPC APIs for the client programming model.

See:  [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Call</td>
<td>The <code>javax.xml.rpc.Call</code> interface provides support for the dynamic invocation of a service endpoint.</td>
</tr>
<tr>
<td>Service</td>
<td><code>Service</code> class acts as a factory of the following: Dynamic proxy for the target service endpoint.</td>
</tr>
<tr>
<td>Stub</td>
<td>The interface <code>javax.xml.rpc.Stub</code> is the common base interface for the stub classes.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NamespaceConstants</td>
<td>Constants used in JAX-RPC for namespace prefixes and URIs</td>
</tr>
<tr>
<td>ParameterMode</td>
<td>The <code>javax.xml.rpc.ParameterMode</code> is a type-safe enumeration for parameter mode.</td>
</tr>
<tr>
<td>ServiceFactory</td>
<td>The <code>javax.xml.rpc.ServiceFactory</code> is an abstract class that provides a factory for the creation of instances of the type <code>javax.xml.rpc.Service</code>.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXRPCException</td>
<td>The <code>javax.xml.rpc.JAXRPCException</code> is thrown from the core JAX-RPC APIs to indicate an exception related to the JAX-RPC runtime mechanisms.</td>
</tr>
<tr>
<td></td>
<td>The <code>javax.xml.rpc.ServiceException</code> is thrown</td>
</tr>
</tbody>
</table>
ServiceException from the methods in the javax.xml.rpc.Service interface and ServiceFactory class.
This package contains the core JAX-RPC APIs for the client programming model.
Hierarchy For Package javax.xml.rpc

Package Hierarchies:
  All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.rpc.**NamespaceConstants**
  - javax.xml.rpc.**ParameterMode**
  - javax.xml.rpc.**ServiceFactory**
  - java.lang.**Throwable** (implements java.io.**Serializable**)  
    - java.lang.**Exception**  
      - java.lang.**RuntimeException**  
        - javax.xml.rpc.**JAXRPCException**
      - javax.xml.rpc.**ServiceException**
Interface Hierarchy

- javax.xml.rpc.Call
- javax.xml.rpc.Service
- javax.xml.rpc.Stub
Package `javax.persistence.spi`

The `javax.persistence.spi` package defines the classes and interfaces that are implemented by the persistence provider and the Java EE container for use by the container, provider, and/or Persistence bootstrap class in deployment and bootstrapping.

See:  Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ClassTransformer</code></td>
<td>A persistence provider supplies an instance of this interface to the</td>
</tr>
<tr>
<td></td>
<td><code>PersistenceUnitInfo.addTransformer</code> method.</td>
</tr>
<tr>
<td><code>PersistenceProvider</code></td>
<td>Interface implemented by a persistence provider.</td>
</tr>
<tr>
<td><code>PersistenceUnitInfo</code></td>
<td>Interface implemented by the container and used by the persistence provider</td>
</tr>
<tr>
<td></td>
<td>when creating an <code>EntityManagerFactory</code>.</td>
</tr>
</tbody>
</table>

### Enum Summary

<table>
<thead>
<tr>
<th>Enum</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>PersistenceUnitTransactionType</code></td>
<td>This enum class defines whether the entity managers created by the</td>
</tr>
<tr>
<td></td>
<td><code>EntityManagerFactory</code> will be JTA or resource-local entity managers.</td>
</tr>
</tbody>
</table>
Package javax.persistence.spi Description

The javax.persistence.spi package defines the classes and interfaces that are implemented by the persistence provider and the Java EE container for use by the container, provider, and/or Persistence bootstrap class in deployment and bootstrapping.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.persistence.spi

Package Hierarchies:
- All Packages
Interface Hierarchy

- javax.persistence.spi.ClassTransformer
- javax.persistence.spi.PersistenceProvider
- javax.persistence.spi.PersistenceUnitInfo
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)  
    - javax.persistence.spi.**PersistenceUnitTransactionType**
Package javax.enterprise.deploy.spi.status

Provides J2EE Product Vendor deployment status implementation classes.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ClientConfiguration</strong></td>
</tr>
<tr>
<td><strong>DeploymentStatus</strong></td>
</tr>
<tr>
<td><strong>ProgressListener</strong></td>
</tr>
<tr>
<td><strong>ProgressObject</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ProgressEvent</strong></td>
</tr>
</tbody>
</table>
Package javax.enterprise.deploy.spi.status

Description

Provides J2EE Product Vendor deployment status implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- [J2EE Tools](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package
javax.enterprise.deploy.spi.status

Package Hierarchies:
  All Packages
Class Hierarchy

- java.lang.**Object**
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.enterprise.deploy.spi.status.**ProgressEvent**
Interface Hierarchy

- `javax.enterprise.deploy.spi.status.DeploymentStatus`
- `java.util.EventListener`
  - `javax.enterprise.deploy.spi.status.ProgressListener`
- `javax.enterprise.deploy.spi.status.ProgressObject`
- `java.io.Serializable`
  - `javax.enterprise.deploy.spi.status.ClientConfiguration`

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.xml.bind.annotation.adapters

XmlAdapter and its spec-defined sub-classes to allow arbitrary Java classes to be used with JAXB.

See: [Description](#)

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CollapsedStringAdapter</td>
<td>Built-in XmlAdapter to handle xs:token and its derived types.</td>
</tr>
<tr>
<td>HexBinaryAdapter</td>
<td>XmlAdapter for xs:hexBinary.</td>
</tr>
<tr>
<td>NormalizedStringAdapter</td>
<td>XmlAdapter to handle xs:normalizedString.</td>
</tr>
<tr>
<td>XmlAdapter&lt; ValueType, BoundType &gt;</td>
<td>Adapts a Java type for custom marshaling.</td>
</tr>
<tr>
<td>XmlJavaTypeAdapter.DEFAULT</td>
<td>Used in XmlJavaTypeAdapter.type() to signal that the type be inferred from the signature of the field, property, parameter or the class.</td>
</tr>
</tbody>
</table>

### Annotation Types Summary

<table>
<thead>
<tr>
<th>Annotation Types Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XmlJavaTypeAdapter</td>
<td>Use an adapter that implements XmlAdapter for custom marshaling.</td>
</tr>
<tr>
<td>XmlJavaTypeAdapters</td>
<td>A container for multiple @XmlJavaTypeAdapter annotations.</td>
</tr>
</tbody>
</table>
Package javax.xml.bind.annotation.adapters

Description

XmlAdapter and its spec-defined sub-classes to allow arbitrary Java classes to be used with JAXB.
Package Specification

- JAXB Specification
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- The JAXB Website

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.xml.bind.annotation.adapters

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.bind.annotation.adapters.**XmlAdapter**<Value_Type,Bound_Type>
    - javax.xml.bind.annotation.adapters.**CollapsedStringAdapter**
    - javax.xml.bind.annotation.adapters.**HexBinaryAdapter**
    - javax.xml.bind.annotation.adapters.**NormalizedStringAdapter**
  - javax.xml.bind.annotation.adapters.**XmlJavaTypeAdapter.DEFA**
Annotation Type Hierarchy

- `javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.adapters.XmlJavaTypeAdapters` (implements `java.lang.annotation.Annotation`)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
## Package javax.resource.cci

The javax.resource.cci package contains API specification for the Common Client Interface (CCI).

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection</td>
<td>A Connection represents an application-level handle that is used by a client to access the underlying physical connection.</td>
</tr>
<tr>
<td>ConnectionFactory</td>
<td>ConnectionFactory provides an interface for getting connection to an EIS instance.</td>
</tr>
<tr>
<td>ConnectionMetaData</td>
<td>The interface ConnectionMetaData provides information about an EIS instance connected through a Connection instance.</td>
</tr>
<tr>
<td>ConnectionSpec</td>
<td>ConnectionSpec is used by an application component to pass connection request-specific properties to the ConnectionFactory.</td>
</tr>
<tr>
<td>IndexedRecord</td>
<td>IndexedRecord represents an ordered collection of record elements based on the java.util.List interface.</td>
</tr>
<tr>
<td>Interaction</td>
<td>The javax.resource.cci.Interaction enables a component to execute EIS functions.</td>
</tr>
<tr>
<td>InteractionSpec</td>
<td>An InteractionSpec holds properties for driving an Interaction with an EIS instance.</td>
</tr>
<tr>
<td>LocalTransaction</td>
<td>The LocalTransaction defines a transaction demarcation interface for resource manager local transactions.</td>
</tr>
<tr>
<td>MappedRecord</td>
<td>The interface javax.resource.cci.MappedRecord is used for key-value map based representation of record elements.</td>
</tr>
<tr>
<td><strong>MessageListener</strong></td>
<td>This serves as a request-response message listener type that message endpoints (message-driven beans) may implement.</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Record</strong></td>
<td>The <code>javax.resource.cci.Record</code> interface is the base interface for the representation of an input or output to the execute methods defined on an Interaction.</td>
</tr>
<tr>
<td><strong>RecordFactory</strong></td>
<td>The <code>RecordFactory</code> interface is used for creating <code>MappedRecord</code> and <code>IndexedRecord</code> instances.</td>
</tr>
<tr>
<td><strong>ResultSet</strong></td>
<td>A <code>ResultSet</code> represents tabular data that is retrieved from an EIS instance by the execution of an Interaction..</td>
</tr>
<tr>
<td><strong>ResultSetInfo</strong></td>
<td>The interface <code>javax.resource.cci.ResultSetInfo</code> provides information on the support provided for <code>ResultSet</code> by a connected EIS instance.</td>
</tr>
<tr>
<td><strong>Streamable</strong></td>
<td>Streamable interface enables a resource adapter to extract data from an input Record or set data into an output Record as a stream of bytes.</td>
</tr>
</tbody>
</table>

### Exception Summary

| **ResourceWarning** | A `ResourceWarning` provides information on warnings related to execution of an interaction with an EIS. |
Package javax.resource.cci Description

The javax.resource.cci package contains API specification for the Common Client Interface (CCI).

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.resource.**ResourceException**
      - javax.resource.cci.**ResourceWarning**
Interface Hierarchy

- java.lang.**Cloneable**
  - javax.resource.cci.**Record** (also extends java.io.**Serializable**)
  - javax.resource.cci.**IndexedRecord** (also extends java.util.**List**<E>, java.io.**Serializable**)
  - javax.resource.cci.**MappedRecord** (also extends java.util.**Map**<K,V>, java.io.**Serializable**)
  - javax.resource.cci.**ResultSet** (also extends java.sql.**ResultSet**)
- javax.resource.cci.**Connection**
- javax.resource.cci.**ConnectionMetaData**
- javax.resource.cci.**ConnectionSpec**
- javax.resource.cci.**Interaction**
- java.lang.**Iterable**<T>
  - java.util.**Collection**<E>
  - java.util.**List**<E>
    - javax.resource.cci.**IndexedRecord** (also extends javax.resource.cci.**Record**, java.io.**Serializable**)
- javax.resource.cci.**LocalTransaction**
- java.util.**Map**<K,V>
  - javax.resource.cci.**MappedRecord** (also extends javax.resource.cci.**Record**, java.io.**Serializable**)
- javax.resource.cci.**MessageListener**
- javax.resource.cci.**RecordFactory**
- javax.naming.**Referenceable**
  - javax.resource.**Referenceable**
    - javax.resource.cci.**ConnectionFactory** (also extends java.io.**Serializable**)
- javax.resource.cci.**ResourceAdapterMetaData**
- java.sql.**ResultSet**
  - javax.resource.cci.**ResultSet** (also extends javax.resource.cci.**Record**)
- javax.resource.cci.**ResultSetInfo**
- java.io.**Serializable**
  - javax.resource.cci.**ConnectionFactory** (also extends javax.resource.**Referenceable**)


javax.resource.cci.IndexedRecord (also extends java.util.List<E>, javax.resource.cci.Record)
javax.resource.cci.InteractionSpec
javax.resource.cci.MappedRecord (also extends java.util.Map<K,V>, javax.resource.cci.Record)
javax.resource.cci.Record (also extends java.lang.Cloneable)
  javax.resource.cci.IndexedRecord (also extends java.util.List<E>, java.io.Serializable)
  javax.resource.cci.MappedRecord (also extends java.util.Map<K,V>, java.io.Serializable)
  javax.resource.cci.ResultSet (also extends java.sql.ResultSet)
javax.resource.cci.Streamable

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
## Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionListener</code></td>
<td>This is the Listener interface for Connection events.</td>
</tr>
<tr>
<td><code>FolderListener</code></td>
<td>This is the Listener interface for Folder events.</td>
</tr>
<tr>
<td><code>MessageChangedListener</code></td>
<td>This is the Listener interface for MessageChanged events</td>
</tr>
<tr>
<td><code>MessageCountListener</code></td>
<td>This is the Listener interface for MessageCount events.</td>
</tr>
<tr>
<td><code>StoreListener</code></td>
<td>This is the Listener interface for Store Notifications.</td>
</tr>
<tr>
<td><code>TransportListener</code></td>
<td>This is the Listener interface for Transport events.</td>
</tr>
</tbody>
</table>

## Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ConnectionAdapter</code></td>
<td>The adapter which receives connection events.</td>
</tr>
<tr>
<td><code>ConnectionEvent</code></td>
<td>This class models Connection events.</td>
</tr>
<tr>
<td><code>FolderAdapter</code></td>
<td>The adapter which receives Folder events.</td>
</tr>
<tr>
<td><code>FolderEvent</code></td>
<td>This class models Folder existence events.</td>
</tr>
<tr>
<td><code>MailEvent</code></td>
<td>Common base class for mail events, defining the dispatch method.</td>
</tr>
<tr>
<td><code>MessageChangedEvent</code></td>
<td>This class models Message change events.</td>
</tr>
<tr>
<td><code>MessageCountAdapter</code></td>
<td>The adapter which receives MessageCount events.</td>
</tr>
<tr>
<td><strong>MessageCountEvent</strong></td>
<td>This class notifies changes in the number of messages in a folder.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>StoreEvent</strong></td>
<td>This class models notifications from the Store connection.</td>
</tr>
<tr>
<td><strong>TransportAdapter</strong></td>
<td>The adapter which receives Transport events.</td>
</tr>
<tr>
<td><strong>TransportEvent</strong></td>
<td>This class models Transport events.</td>
</tr>
</tbody>
</table>
Package javax.mail.event Description

Listeners and events for the JavaMail API. This package defines listener classes and event classes used by the classes defined in the javax.mail package.
Hierarchy For Package javax.mail.event

Package Hierarchies:

All Packages
Class Hierarchy

- `java.lang.Object`
  - `javax.mail.event.ConnectionAdapter` (implements `javax.mail.event.ConnectionListener`)
  - `java.util.EventObject` (implements `java.io.Serializable`)
    - `javax.mail.event.MailEvent`
      - `javax.mail.event.ConnectionEvent`
      - `javax.mail.event.FolderEvent`
      - `javax.mail.event.MessageChangedEvent`
      - `javax.mail.event.MessageCountEvent`
      - `javax.mail.event.StoreEvent`
      - `javax.mail.event.TransportEvent`
  - `javax.mail.event.FolderAdapter` (implements `javax.mail.event.FolderListener`)
  - `javax.mail.event.MessageCountAdapter` (implements `javax.mail.event.MessageCountListener`)
  - `javax.mail.event.TransportAdapter` (implements `javax.mail.event.TransportListener`)**
Interface Hierarchy

- java.util.EventListener
  - javax.mail.event.ConnectionListener
  - javax.mail.event.FolderListener
  - javax.mail.event.MessageChangedListener
  - javax.mail.event.MessageCountListener
  - javax.mail.event.StoreListener
  - javax.mail.event.TransportListener
Package javax.servlet.http

The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.

See:  

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HttpServletRequest</td>
<td>Extends the ServletRequest interface to provide request information for HTTP servlets.</td>
</tr>
<tr>
<td>HttpServletResponse</td>
<td>Extends the ServletResponse interface to provide HTTP-specific functionality in sending a response.</td>
</tr>
<tr>
<td>HttpSession</td>
<td>Provides a way to identify a user across more than one page request or visit to a Web site and to store information about that user.</td>
</tr>
<tr>
<td>HttpSessionActivationListener</td>
<td>Objects that are bound to a session may listen to container events notifying them that sessions will be passivated and that session will be activated.</td>
</tr>
<tr>
<td>HttpSessionAttributeListener</td>
<td>This listener interface can be implemented in order to get notifications of changes to the attribute lists of sessions within this web application.</td>
</tr>
<tr>
<td>HttpSessionBindingListener</td>
<td>Causes an object to be notified when it is bound to or unbound from a session.</td>
</tr>
<tr>
<td>HttpSessionContext</td>
<td>Deprecated. As of Java(tm) Servlet API 2.1 for security reasons, with no</td>
</tr>
<tr>
<td>Class Name</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Cookie</strong></td>
<td>Creates a cookie, a small amount of information sent by a servlet to a Web browser, saved by the browser, and later sent back to the server.</td>
</tr>
<tr>
<td><strong>HttpServlet</strong></td>
<td>Provides an abstract class to be subclassed to create an HTTP servlet suitable for a Web site.</td>
</tr>
<tr>
<td><strong>HttpServletRequestWrapper</strong></td>
<td>Provides a convenient implementation of the HttpServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet.</td>
</tr>
<tr>
<td><strong>HttpServletResponseWrapper</strong></td>
<td>Provides a convenient implementation of the HttpServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet.</td>
</tr>
<tr>
<td><strong>HttpSessionBindingEvent</strong></td>
<td>Events of this type are either sent to an object that implements HttpSessionBindingListener when it is bound or unbound from a session, or to a HttpSessionAttributeListener that has been configured in the deployment descriptor when any attribute is bound, unbound or replaced in a session.</td>
</tr>
<tr>
<td><strong>HttpSessionEvent</strong></td>
<td>This is the class representing event notifications for changes to sessions within a web application.</td>
</tr>
<tr>
<td><strong>HttpUtils</strong></td>
<td></td>
</tr>
<tr>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td><strong>Deprecated. As of Java(tm) Servlet API 2.3.</strong></td>
<td></td>
</tr>
</tbody>
</table>
Package javax.servlet.http Description

The javax.servlet.http package contains a number of classes and interfaces that describe and define the contracts between a servlet class running under the HTTP protocol and the runtime environment provided for an instance of such a class by a conforming servlet container.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.servlet.http

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.servlet.http.**Cookie** (implements java.lang.**Cloneable**)
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.servlet.http.**HttpSessionEvent**
    - javax.servlet.http.**HttpSessionBindingEvent**
  - javax.servlet.**GenericServlet** (implements java.io.**Serializable**, java.servlet.**Servlet**, java.servlet.**ServletConfig**)
    - javax.servlet.http.**HttpServletRequest**
    - javax.servlet.http.**ServletResponse**
      - javax.servlet.http.**HttpServletResponse**
        - javax.servlet.http.**HttpServletRequestWrapper** (implements javax.servlet.http.**HttpServletRequest**)
Interface Hierarchy

- java.util.EventListener
  - javax.servlet.http.HttpSessionActivationListener
  - javax.servlet.http.HttpSessionAttributeListener
  - javax.servlet.http.HttpSessionBindingListener
  - javax.servlet.http.HttpSessionListener
- javax.servlet.http.HttpSession
- javax.servlet.http.HttpSessionContext
- javax.servlet.ServletRequest
  - javax.servlet.HttpServletRequest
- javax.servlet.ServletResponse
  - javax.servlet.HttpServletResponse
Package `javax.enterprise.deploy.spi`

Provides J2EE Product Vendor implementation classes.

See: [Description](#)

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DConfigBean</strong></td>
</tr>
<tr>
<td><strong>DConfigBeanRoot</strong></td>
</tr>
<tr>
<td><strong>DeploymentConfiguration</strong></td>
</tr>
<tr>
<td><strong>DeploymentManager</strong></td>
</tr>
<tr>
<td><strong>Target</strong></td>
</tr>
<tr>
<td><strong>TargetModuleID</strong></td>
</tr>
</tbody>
</table>
Package javax.enterprise.deploy.spi Description

Provides J2EE Product Vendor implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.spi

Package Hierarchies:
All Packages
Interface Hierarchy

- javax.enterprise.deploy.spi.DConfigBean
  - javax.enterprise.deploy.spi.DConfigBeanRoot
- javax.enterprise.deploy.spi.DeploymentConfiguration
- javax.enterprise.deploy.spi.DeploymentManager
- javax.enterprise.deploy.spi.Target
- javax.enterprise.deploy.spi.TargetModuleID

Overview  Package  Class  Deprecated  Index  Help
PREV  NEXT  FRAMES  NO FRAMES

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.enterprise.deploy.model

Provides Tool Vendor implementation classes.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDBean</td>
<td>An interface for beans that represent a fragment of a standard deployment descriptor.</td>
</tr>
<tr>
<td>DDBeanRoot</td>
<td>An interface that represents the root of a standard deployment descriptor.</td>
</tr>
<tr>
<td>DeployableObject</td>
<td>The DeployableObject interface is an abstract representation of a J2EE deployable module (JAR, WAR, RAR, EAR).</td>
</tr>
<tr>
<td>J2eeApplicationObject</td>
<td>J2eeApplicationObject is an interface that represents a J2EE application (EAR); it maintains a DeployableObject for each module in the archive.</td>
</tr>
<tr>
<td>XpathListener</td>
<td>The listener interface for receiving XpathEvents</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XpathEvent</td>
<td>An Event class describing ConfigBeans being added/subtracted from a server configuration.</td>
</tr>
</tbody>
</table>
Package javax.enterprise.deploy.model

Description

Provides Tool Vendor implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.model

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.enterprise.deploy.model.**XpathEvent**
Interface Hierarchy

- javax.enterprise.deploy.model.DDBean
  - javax.enterprise.deploy.model.DDBeanRoot
- javax.enterprise.deploy.model.DeployableObject
  - javax.enterprise.deploy.model.J2eeApplicationObject
- javax.enterprise.deploy.model.XPathListener

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package
javax.enterprise.deploy.model.exceptions

Provides Tool Vendor exception implementation classes.

See:  Description

<table>
<thead>
<tr>
<th>Exception Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DDBeanCreateException</strong></td>
</tr>
<tr>
<td>This exception reports errors in generating a DDBean.</td>
</tr>
</tbody>
</table>
Package
javax.enterprise.deploy.model.exceptions

Description

Provides Tool Vendor exception implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.model.exceptions

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.enterprise.deploy.model.exceptions.**DDBeanCreateException**
## Annotation Types Summary

<table>
<thead>
<tr>
<th>Annotation Types</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeclareRoles</strong></td>
<td>Used by application to declare roles.</td>
</tr>
<tr>
<td><strong>DenyAll</strong></td>
<td>Specifies that no security roles are allowed to invoke the specified method(s) - i.e. that the methods are to be excluded from execution in the J2EE container.</td>
</tr>
<tr>
<td><strong>PermitAll</strong></td>
<td>Specifies that all security roles are allowed to invoke the specified method(s) i.e. that the specified method(s) are &quot;unchecked&quot;.</td>
</tr>
<tr>
<td><strong>RolesAllowed</strong></td>
<td>Specifies the list of roles permitted to access method(s) in an application.</td>
</tr>
<tr>
<td><strong>RunAs</strong></td>
<td>Defines the identity of the application during execution in a J2EE container.</td>
</tr>
</tbody>
</table>
Hierarchy For Package javax.annotation.security

Package Hierarchies:
   All Packages
Annotation Type Hierarchy

- javax.annotation.security.**DeclareRoles** (implements java.lang.annotation.**Annotation**)
- javax.annotation.security.**DenyAll** (implements java.lang.annotation.**Annotation**)
- javax.annotation.security.**PermitAll** (implements java.lang.annotation.**Annotation**)
- javax.annotation.security.**RolesAllowed** (implements java.lang.annotation.**Annotation**)
- javax.annotation.security.**RunAs** (implements java.lang.annotation.**Annotation**)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://example.com/license).
Package javax.enterprise.deploy.spi.factories

Provides J2EE Product Vendor deployment factory implementation classes.

See:  Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeploymentFactory</td>
</tr>
</tbody>
</table>
Package javax.enterprise.deploy.spi.factories

Description

Provides J2EE Product Vendor deployment factory implementation classes.
Package Specification

- [JSR 88, J2EE Application Deployment](#)
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.spi.factories

Package Hierarchies:
All Packages
Interface Hierarchy

- `javax.enterprise.deploy.spi.factories.DeploymentFactory`
Package
javax.enterprise.deploy.shared.factories

Provides shared factory manager object for Tool Vendor and Product Vendor implementation classes.

See:  Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeploymentFactoryManager</strong></td>
</tr>
</tbody>
</table>
Package
tjavax.enterprise.deploy.shared.factories
Description

Provides shared factory manager object for Tool Vendor and Product Vendor implementation classes.
Package Specification

- JSR 88, J2EE Application Deployment
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- J2EE Tools

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.enterprise.deploy.shared.factories

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.enterprise.deploy.shared.factories.**DeploymentFactoryMa**
Package javax.xml.rpc.encoding

This package defines APIs for the extensible type mapping framework.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DeserializationContext</strong></td>
</tr>
<tr>
<td><strong>Deserializer</strong></td>
</tr>
<tr>
<td><strong>DeserializerFactory</strong></td>
</tr>
<tr>
<td><strong>SerializationContext</strong></td>
</tr>
<tr>
<td><strong>Serializer</strong></td>
</tr>
<tr>
<td><strong>SerializerFactory</strong></td>
</tr>
<tr>
<td><strong>TypeMapping</strong></td>
</tr>
<tr>
<td><strong>TypeMappingRegistry</strong></td>
</tr>
<tr>
<td>Class Summary</td>
</tr>
<tr>
<td>---------------</td>
</tr>
<tr>
<td><strong>XMLType</strong></td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.encoding Description

This package defines APIs for the extensible type mapping framework.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.rpc.encoding

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.rpc.encoding.**XMLType**
Interface Hierarchy

- javax.xml.rpc.encoding.**DeserializationContext**
- java.io.**Serializable**
  - javax.xml.rpc.encoding.**Deserializer**
  - javax.xml.rpc.encoding.**DeserializerFactory**
  - javax.xml.rpc.encoding.**Serializer**
  - javax.xml.rpc.encoding.**SerializerFactory**
  - javax.xml.rpc.encoding.**TypeMappingRegistry**
- javax.xml.rpc.encoding.**SerializationContext**
- javax.xml.rpc.encoding.**TypeMapping**
Package javax.xml.bind.annotation

Defines annotations for customizing Java program elements to XML Schema mapping.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DomHandler&lt;ElementT,ResultT&gt; extends Result&gt;</td>
<td>Converts an element (and its descendants) from/to DOM (or similar) representation.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>W3CDomHandler</td>
<td>DomHandler implementation for W3C DOM (org.w3c.dom package.)</td>
</tr>
<tr>
<td>XmlElement.DEFAULT</td>
<td>Used in XmlElement.type() to signal that the type be inferred from the signature of the property.</td>
</tr>
<tr>
<td>XmlElementDecl.GLOBAL</td>
<td>Used in XmlElementDecl.scope() to signal that the declaration is in the global scope.</td>
</tr>
<tr>
<td>XmlElementRef.DEFAULT</td>
<td>Used in XmlElementRef.type() to signal that the type be inferred from the signature of the property.</td>
</tr>
<tr>
<td>XmlSchemaType.DEFAULT</td>
<td>Used in XmlSchemaType.type() to signal that the type be inferred from the signature of the property.</td>
</tr>
<tr>
<td>XmlType.DEFAULT</td>
<td>Used in XmlType.factoryClass() to signal that either factory method is not used or that it's in the class with this XmlType itself.</td>
</tr>
</tbody>
</table>

### Enum Summary
<table>
<thead>
<tr>
<th>Annotation Types Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XmlAccessorOrder</strong></td>
</tr>
<tr>
<td><strong>XmlAccessorType</strong></td>
</tr>
<tr>
<td><strong>XmlAnyAttribute</strong></td>
</tr>
<tr>
<td><strong>XmlAnyElement</strong></td>
</tr>
<tr>
<td><strong>XmlAttachmentRef</strong></td>
</tr>
<tr>
<td><strong>XmlAttribute</strong></td>
</tr>
<tr>
<td><strong>XmlElement</strong></td>
</tr>
<tr>
<td><strong>XmlElementDecl</strong></td>
</tr>
<tr>
<td><strong>XmlElementRef</strong></td>
</tr>
<tr>
<td><strong>XmlElementRefs</strong></td>
</tr>
<tr>
<td><strong>XmlElements</strong></td>
</tr>
<tr>
<td><strong>XmlElementWrapper</strong></td>
</tr>
<tr>
<td><strong>XmlEnum</strong></td>
</tr>
<tr>
<td><strong>XmlEnumValue</strong></td>
</tr>
<tr>
<td>Annotation</td>
</tr>
<tr>
<td>--------------</td>
</tr>
<tr>
<td>XmlID</td>
</tr>
<tr>
<td>XmlIDREF</td>
</tr>
<tr>
<td>XmlInlineBinaryData</td>
</tr>
<tr>
<td>XmlList</td>
</tr>
<tr>
<td>XmlMimeType</td>
</tr>
<tr>
<td>XmlMixed</td>
</tr>
<tr>
<td>XmlNs</td>
</tr>
<tr>
<td>XmlRegistry</td>
</tr>
<tr>
<td>XmlRootElement</td>
</tr>
<tr>
<td>XmlSchema</td>
</tr>
<tr>
<td>XmlSchemaType</td>
</tr>
<tr>
<td>XmlSchemaTypes</td>
</tr>
<tr>
<td>XmlTransient</td>
</tr>
<tr>
<td>XmlType</td>
</tr>
<tr>
<td>XmlValue</td>
</tr>
</tbody>
</table>
Package javax.xml.bind.annotation Description

Defines annotations for customizing Java program elements to XML Schema mapping.
Package Specification

The following table shows the JAXB mapping annotations that can be associated with each program element.

<table>
<thead>
<tr>
<th>Program Element</th>
<th>JAXB annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Package</td>
<td><code>XmlAccessorOrder</code>, <code>XmlAccessorType</code>, <code>XmlSchema</code>, <code>XmlSchemaType</code>, <code>XmlSchemaTypes</code>, <code>XmlJavaTypeAdapter</code>, <code>XmlJavaTypeAdapters</code></td>
</tr>
<tr>
<td>Class</td>
<td><code>XmlAccessorOrder</code>, <code>XmlAccessorType</code>, <code>XmlInlineBinaryData</code>, <code>XmlRootElement</code>, <code>XmlType</code>, <code>XmlJavaTypeAdapter</code></td>
</tr>
<tr>
<td>Enum type</td>
<td><code>XmlEnum</code>, <code>XmlEnumValue (enum constant only)</code>, <code>XmlRootElement</code>, <code>XmlType</code>, <code>XmlJavaTypeAdapter</code></td>
</tr>
<tr>
<td></td>
<td><code>XmlElement</code>, <code>XmlElements</code>, <code>XmlElementRef</code>, <code>XmlElementRefs</code>, <code>XmlElementWrapper</code>, <code>XmlAnyElement</code>, <code>XmlAttribute</code></td>
</tr>
</tbody>
</table>
| JavaBean Property/field | $\text{XmlAnyAttribute}$  
| | $\text{XmlTransient}$  
| | $\text{XmlValue}$  
| | $\text{XmlID}$  
| | $\text{XmlIDREF}$  
| | $\text{XmlList}$  
| | $\text{XmlMixed}$  
| | $\text{XmlMimeType}$  
| | $\text{XmlAttachmentRef}$  
| | $\text{XmlInlineBinaryData}$  
| | $\text{XmlElementDecl (only on method)}$  
| | $\text{XmlJavaTypeAdapter}$  
| Parameter | $\text{XmlList}$  
| | $\text{XmlAttachmentRef}$  
| | $\text{XmlMimeType}$  
| | $\text{XmlJavaTypeAdapter}$  

**Terminology**

**JavaBean property and field:** For the purposes of mapping, there is no semantic difference between a field and a JavaBean property. Thus, an annotation that can be applied to a JavaBean property can always be applied to a field. Hence in the Javadoc documentation, for brevity, the term JavaBean property or property is used to mean either JavaBean property or a field. Where required, both are explicitly mentioned.

**top level class:** For the purpose of mapping, there is no semantic difference between a top level class and a static nested class. Thus, an annotation that can be applied to a top level class, can always be applied to a nested static class. Hence in the Javadoc documentation, for brevity, the term "top level class" or just class is used to mean either a top level class or a nested static class.

**mapping annotation:** A JAXB 2.0 defined program annotation based on the JSR 175 programming annotation facility.
Common Usage Constraints

The following usage constraints are defined here since they apply to more than annotation:

- For a property, a given annotation can be applied to either read or write property but not both.
- A property name must be different from any other property name in any of the super classes of the class being mapped.
- A mapped field name or the decapitalized name of a mapped property must be unique within a class.

Notations

Namespace prefixes

The following namespace prefixes are used in the XML Schema fragments in this package.

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>xs</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>Namespace of XML Schema namespace</td>
</tr>
<tr>
<td>ref</td>
<td><a href="http://ws-i.org/profiles/basic/1.1/xsd">http://ws-i.org/profiles/basic/1.1/xsd</a></td>
<td>Namespace for swaRef schema component</td>
</tr>
<tr>
<td>xsi</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
<td>XML Schema namespace for instances</td>
</tr>
</tbody>
</table>

Since:
JAXB 2.0
Hierarchy For Package
javax.xml.bind.annotation

Package Hierarchies:
  All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.bind.annotation.**W3CDomHandler** (implements javax.xml.bind.annotation.**DomHandler**<ElementT,ResultT>)
  - javax.xml.bind.annotation.**XmlElement.DEFAULT**
  - javax.xml.bind.annotation.**XmlElementDecl.GLOBAL**
  - javax.xml.bind.annotation.**XmlElementRef.DEFAULT**
  - javax.xml.bind.annotation.**XmlSchemaType.DEFAULT**
  - javax.xml.bind.annotation.**XmlType.DEFAULT**
Interface Hierarchy

- javax.xml.bind.annotation.DomHandler<
  ElementT, ResultT>
Annotation Type Hierarchy

- `javax.xml.bind.annotation.XmlAccessorOrder` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlAccessorType` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlAnyAttribute` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlAnyElement` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlAttachmentRef` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlAttribute` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElement` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElementDecl` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElementRef` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElementRefs` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElementWrapper` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlElements` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlEnum` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlEnumValue` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlID` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlIDREF` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlInlineBinaryData` (implements `java.lang.annotation.Annotation`)
- `javax.xml.bind.annotation.XmlList` (implements `java.lang.annotation.Annotation`...
java.lang.annotation.Annotation
- javax.xml.bind.annotation.XmlMimeType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlMixed (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlNs (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlRegistry (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlRootElement (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchema (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaTypes (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlTransient (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlValue (implements java.lang.annotation.Annotation)
Enum Hierarchy

- java.lang.Object
    - javax.xml.bind.annotation.XmlAccessOrder
    - javax.xml.bind.annotation.XmlAccessType
    - javax.xml.bind.annotation.XmlNsForm

### Overview Package Class Deprecated Index Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.faces.validator`

Interface defining the validator model, and concrete validator implementation classes.

See:  [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Validator</code></td>
<td>A <code>Validator</code> implementation is a class that can perform validation (correctness checks) on a <code>EditableValueHolder</code>.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>DoubleRangeValidator</code></td>
<td>A <code>DoubleRangeValidator</code> is a <code>Validator</code> that checks the value of the corresponding component against specified minimum and maximum values.</td>
</tr>
<tr>
<td><code>LengthValidator</code></td>
<td>A <code>LengthValidator</code> is a <code>Validator</code> that checks the number of characters in the String representation of the value of the associated component.</td>
</tr>
<tr>
<td><code>LongRangeValidator</code></td>
<td>A <code>LongRangeValidator</code> is a <code>Validator</code> that checks the value of the corresponding component against specified minimum and maximum values.</td>
</tr>
<tr>
<td><code>MethodExpressionValidator</code></td>
<td>A <code>MethodExpressionValidator</code> is a <code>Validator</code> that wraps a <code>MethodExpression</code>, and it performs validation by executing a method on an object identified by the <code>MethodExpression</code>.</td>
</tr>
</tbody>
</table>

### Exception Summary

| Exception Summary | |
|-------------------| |
| **ValidatorException** | A **ValidatorException** is an exception thrown by the `validate()` method of a **Validator** to indicate that validation failed. |
Package javax.faces.validator Description

Interface defining the validator model, and concrete validator implementation classes.
Hierarchy For Package javax.faces.validator

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.validator.**DoubleRangeValidator** (implements javax.faces.component.**StateHolder**, javax.faces.validator.**Validator**)
  - javax.faces.validator.**LengthValidator** (implements javax.faces.component.**StateHolder**, javax.faces.validator.**Validator**)
  - javax.faces.validator.**LongRangeValidator** (implements javax.faces.component.**StateHolder**, javax.faces.validator.**Validator**)
  - javax.faces.validator.**MethodExpressionValidator** (implements javax.faces.component.**StateHolder**, javax.faces.validator.**Validator**)
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - java.lang.**RuntimeException**
        - javax.faces.**FacesException**
          - javax.faces.validator.**ValidatorException**
Interface Hierarchy

- java.util.EventListener
- javax.faces.validator.Validator

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.security.jacc

This package contains the Java Authorization Contract for Containers API

See:  Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PolicyConfiguration</strong></td>
<td>The methods of this interface are used by containers to create policy statements in a Policy provider.</td>
</tr>
<tr>
<td><strong>PolicyContextHandler</strong></td>
<td>This interface defines the methods that must be implemented by handlers that are to be registered and activated by the PolicyContext class.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EJBMethedPermission</strong></td>
<td>Class for EJB method permissions.</td>
</tr>
<tr>
<td><strong>EJBRoleRefPermission</strong></td>
<td>Class for EJB isCallerInRole (String reference) permissions.</td>
</tr>
<tr>
<td><strong>PolicyConfigurationFactory</strong></td>
<td>Abstract factory and finder class for obtaining the instance of the class that implements the PolicyConfigurationFactory of a provider.</td>
</tr>
<tr>
<td><strong>PolicyContext</strong></td>
<td>This utility class is used by containers to communicate policy context identifiers and other policy relevant context to Policy providers.</td>
</tr>
<tr>
<td><strong>WebResourcePermission</strong></td>
<td>Class for Servlet web resource permissions.</td>
</tr>
<tr>
<td><strong>WebRoleRefPermission</strong></td>
<td>Class for Servlet isUserInRole (String reference) permissions.</td>
</tr>
</tbody>
</table>
WebUserDataPermission

Class for Servlet Web user data permissions.

Exception Summary

<table>
<thead>
<tr>
<th>PolicyConfigurationException</th>
</tr>
</thead>
<tbody>
<tr>
<td>This checked exception is thrown by implementations of the javax.security.jacc.PolicyConfiguration Interface, the javax.security.jacc.PolicyConfigurationFactory abstract class, the javax.security.jacc.PolicyContext utility class and implementations of the javax.security.jacc.PolicyContextException Interface.</td>
</tr>
</tbody>
</table>
Package javax.security.jacc Description

This package contains the Java Authorization Contract for Containers API
Hierarchy For Package javax.security.jacc

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.security.**Permission** (implements java.security.**Guard**, java.io.**Serializable**)
    - javax.security.jacc.**EJBMethoPermission** (implements java.io.**Serializable**)
    - javax.security.jacc.**EJBRoleRefPermission** (implements java.io.**Serializable**)
    - javax.security.jacc.**WebResourcePermission** (implements java.io.**Serializable**)
    - javax.security.jacc.**WebRoleRefPermission** (implements java.io.**Serializable**)
    - javax.security.jacc.**WebUserDataPermission** (implements java.io.**Serializable**)
  - javax.security.jacc.**PolicyConfigurationFactory**
  - javax.security.jacc.**PolicyContext**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.security.jacc.**PolicyContextException**
Interface Hierarchy

- `javax.security.jacc.PolicyConfiguration`
- `javax.security.jacc.PolicyContextHandler`
**Package javax.servlet.jsp.el**

Provides the `ELResolver` classes that define the object resolution rules that must be supported by a JSP container with the new unified Expression Language.

**See:** [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>FunctionMapper</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>FunctionMapper</code></td>
</tr>
<tr>
<td><code>VariableResolver</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>ELResolver</code></td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Expression</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>ValueExpression</code></td>
</tr>
<tr>
<td><code>ExpressionEvaluator</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>ExpressionFactory</code></td>
</tr>
<tr>
<td><code>ImplicitObjectELResolver</code></td>
<td>Defines variable resolution behavior for the EL implicit objects defined in the JSP specification.</td>
</tr>
<tr>
<td><code>ScopedAttributeELResolver</code></td>
<td>Defines variable resolution behavior for scoped attributes.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ELException</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>ELException</code></td>
</tr>
<tr>
<td><code>ELParseException</code></td>
<td><strong>Deprecated.</strong> As of JSP 2.1, replaced by <code>ELException</code></td>
</tr>
</tbody>
</table>
Package javax.servlet.jsp.el Description

Provides the ELResolver classes that define the object resolution rules that must be supported by a JSP container with the new unified Expression Language.

The package also defines programmatic access to the old Expression Language evaluator (pre JSP 2.1).

Please note that as of JSP 2.1, all classes and interfaces that were in package javax.servlet.jsp.el have been deprecated in favor of the new unified Expression Language APIs (javax.el). See the Expression Language specification document for more details.

While a JSP container must still support the deprecated APIs defined in javax.servlet.jsp.el, developers should only rely on the new javax.el APIs for new development work.

Two ELResolver classes have been added in JSP 2.1 to implement object resolution rules that must be supported by a JSP container with the new unified Expression Language: ImplicitObjectELResolver and ScopedAttributeELResolver.
Documentation on the old and deprecated API

The JavaServer Pages(tm) (JSP) 2.0 specification provides a portable API for evaluating "EL Expressions". As of JSP 2.0, EL expressions can be placed directly in the template text of JSP pages and tag files.

This package contains a number of classes and interfaces that describe and define programmatic access to the Expression Language evaluator. This API can also be used by an implementation of JSP to evaluate the expressions, but other implementations, like open-coding into Java bytecodes, are allowed. This package is intended to have no dependencies on other portions of the JSP 2.0 specification.

Expression Evaluator

Programmatic access to the EL Expression Evaluator is provided through the following types:

- ExpressionEvaluator
- Expression
- FunctionMapper
- VariableResolver

An ExpressionEvaluator object can be obtained from a JspContext object through the getExpressionEvaluator method. An ExpressionEvaluator encapsulates the EL processor. An EL expression provided as a String can then be evaluated directly, or it can be parsed first into an Expression object. The parse step, can be used to factor out the cost of parsing the expression, or even the cost of optimizing the implementation.

The parsing of an expression string is done against a target type, a default prefix (that applies when a function has no prefix), and a FunctionMapper. The FunctionMapper object maps a prefix and a local name part into a java.lang.reflect.Method object.

The interpretation or evaluation of a parsed expression is done using a VariableResolver object. This object resolves top level object names into Objects. A VariableResolver can be obtained from a JspContext object.
through the getVariableResolver method.

Exceptions

The ELException exception is used by the expression language to denote any exception that may arise during the parsing or evaluation of an expression. The ELParseException exception is a subclass of ELException that corresponds to parsing errors.

Parsing errors are conveyed as exceptions to simplify the API. It is expected that many JSP containers will use additional mechanisms to parse EL expressions and report their errors - a run-time API cannot provide accurate line-error numbers without additional machinery.

Code Fragment

Below is a non-normative code fragment outlining how the APIs can be used.

```java
// Get an instance of an ExpressionEvaluator

ExpressionEvaluator ee = myJspContext.getExpressionEvaluator();
VariableResolver vr = myJspContext.getVariableResolver();

FunctionMapper fm; // we don't have a portable implementation yet

// Example of compiling an expression. See [ISSUE-2]
// Errors detected this way may have higher quality than those
// found with a simple validate() invocation.

ExpressionCompilation ce;

try {
    ce = ee.prepareExpression(expr,
                               targetClass,
                               fm,
                               null // no prefixes
                          );
} catch (ELParseException e) {
    log (e.getMessage());
}

try {
```
ce.evaluate(vr);
} catch (E1Exception e) {
    log (e);
}

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.servlet.jsp.el

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.el.**ELResolver**
    - javax.servlet.jsp.el.**ImplicitObjectELResolver**
    - javax.servlet.jsp.el.**ScopedAttributeELResolver**
  - javax.servlet.jsp.el.**Expression**
  - javax.servlet.jsp.el.**ExpressionEvaluator**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.servlet.jsp.el.**ELException**
      - javax.servlet.jsp.el.**ELParseException**
Interface Hierarchy

- javax.servlet.jsp.el.FunctionMapper
- javax.servlet.jsp.el.VariableResolver
Package `javax.servlet.jsp`

Classes and interfaces for the Core JSP 2.1 API.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>HttpJspPage</code></td>
<td>The <code>HttpJspPage</code> interface describes the interaction that a JSP Page Implementation Class must satisfy when using the HTTP protocol.</td>
</tr>
<tr>
<td><code>JspApplicationContext</code></td>
<td>Stores application-scoped information relevant to JSP containers.</td>
</tr>
<tr>
<td><code>JspPage</code></td>
<td>The <code>JspPage</code> interface describes the generic interaction that a JSP Page Implementation class must satisfy; pages that use the HTTP protocol are described by the <code>HttpJspPage</code> interface.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ErrorData</code></td>
<td>Contains information about an error, for error pages.</td>
</tr>
<tr>
<td><code>JspContext</code></td>
<td><code>JspContext</code> serves as the base class for the <code>PageContext</code> class and abstracts all information that is not specific to servlets.</td>
</tr>
<tr>
<td><code>JspEngineInfo</code></td>
<td>The <code>JspEngineInfo</code> is an abstract class that provides information on the current JSP engine.</td>
</tr>
<tr>
<td><code>JspFactory</code></td>
<td>The <code>JspFactory</code> is an abstract class that defines a number of factory methods available to a JSP page at runtime for the purposes of creating instances of various interfaces and classes used to support the JSP implementation.</td>
</tr>
<tr>
<td></td>
<td>The actions and template data in a JSP page is written...</td>
</tr>
</tbody>
</table>
### Exception Summary

<table>
<thead>
<tr>
<th>Exception Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>JspException</strong></td>
<td>A generic exception known to the JSP engine; uncaught JspExceptions will result in an invocation of the errorpage machinery.</td>
</tr>
<tr>
<td><strong>JspTagException</strong></td>
<td>Exception to be used by a Tag Handler to indicate some unrecoverable error.</td>
</tr>
<tr>
<td><strong>SkipPageException</strong></td>
<td>Exception to indicate the calling page must cease evaluation.</td>
</tr>
</tbody>
</table>
Package javax.servlet.jsp Description

Classes and interfaces for the Core JSP 2.1 API.

The javax.servlet.jsp package contains a number of classes and interfaces that describe and define the contracts between a JSP page implementation class and the runtime environment provided for an instance of such a class by a conforming JSP container.
JSP Page Implementation Object Contract

This section describes the basic contract between a JSP Page implementation object and its container.

The main contract is defined by the classes `JspPage` and `HttpJspPage`. The `JspFactory` class describes the mechanism to portably instantiate all needed runtime objects, and `JspEngineInfo` provides basic information on the current JSP container. Class `JspApplicationContext` stores application-scoped information relevant to JSP containers. It was added in JSP 2.1 to support the integration of the unified Expression Language.

None of these classes are intended to be used by JSP page authors; an example of how these classes may be used is included below.
Implicit Objects

The PageContext object and the JspWriter are available by default as implicit objects.
Exceptions

The `JspException` class is the base class for all JSP exceptions. The `JspTagException` and `SkipPageException` exceptions are used by the tag extension mechanism.

For JSP error pages, the `ErrorData` class encapsulates information about the error.
An Implementation Example

An instance of an implementation dependent subclass of the `PageContext` abstract base class can be created by a JSP implementation class at the beginning of it's `_jspService()` method via an implementation default `JspFactory`.

Here is one example of how to use these classes

```java
public class foo implements Servlet {

    // ...

    public void _jspService(HttpServletRequest request,
        HttpServletResponse response)
        throws IOException, ServletException {

        JspFactory factory = JspFactory.getDefaultFactory();
        PageContext pageContext = factory.getPageContext(
            this, request, response,
            null, // errorPageURL
            false, // needsSession
            JspWriter.DEFAULT_BUFFER,
            true   // autoFlush
        );

        // initialize implicit variables for scripting env ...

        HttpSession session = pageContext.getSession();
        JspWriter out = pageContext.getOut();
        Object page = this;

        try {
            // body of translated JSP here ...
            } catch (Exception e) {
                out.clear();
                pageContext.handlePageException(e);
            } finally {
                out.close();
                factory.releasePageContext(pageContext);
            }
        }
    }
}
Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.servlet.jsp

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.servlet.jsp.**ErrorData**
  - javax.servlet.jsp.**JspContext**
    - javax.servlet.jsp.**PageContext**
  - javax.servlet.jsp.**JspEngineInfo**
  - javax.servlet.jsp.**JspFactory**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.servlet.jsp.**JspException**
      - javax.servlet.jsp.**JspTagException**
      - javax.servlet.jsp.**SkipPageException**
  - java.io.**Writer** (implements java.lang.**Appendable**, java.io.**Closeable**, java.io.**Flushable**)
    - javax.servlet.jsp.**JspWriter**
Interface Hierarchy

- javax.servlet.jsp.JspApplicationContext
- javax.servlet.Servlet
  - javax.servlet.jsp.JspPage
  - javax.servlet.jsp.HttpJspPage

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.faces.el

**DEPRECATED** Classes and interfaces for evaluating and processing reference expressions.

See: [Description](#)

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MethodBinding</td>
<td>Deprecated. <em>This has been replaced by MethodExpression.</em></td>
</tr>
<tr>
<td>PropertyResolver</td>
<td>Deprecated. <em>This has been replaced by ELResolver.</em></td>
</tr>
<tr>
<td>ValueBinding</td>
<td>Deprecated. <em>This has been replaced by ValueExpression.</em></td>
</tr>
<tr>
<td>VariableResolver</td>
<td>Deprecated. <em>This has been replaced by ELResolver when operating with a null base argument.</em></td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EvaluationException</td>
<td>Deprecated. <em>This has been replaced by ELException.</em></td>
</tr>
<tr>
<td>MethodNotFoundException</td>
<td>Deprecated. <em>This has been replaced by MethodNotFoundException.</em></td>
</tr>
<tr>
<td>PropertyNotFoundException</td>
<td>Deprecated. <em>This has been replaced by PropertyNotFoundException.</em></td>
</tr>
<tr>
<td>ReferenceSyntaxException</td>
<td>Deprecated. <em>This has been replaced by ELException.</em></td>
</tr>
</tbody>
</table>
Package javax.faces.el Description

DEPRECATED Classes and interfaces for evaluating and processing reference expressions. The main class in this package is ValueBinding, which is the runtime representation of a reference expression. ValueBinding provides methods to get and set the value of the expression.
Hierarchy For Package javax.faces.el

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.el.**MethodBinding**
  - javax.faces.el.**PropertyResolver**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - java.lang.**RuntimeException**
        - javax.faces.**FacesException**
          - javax.faces.el.**EvaluationException**
            - javax.faces.el.**MethodNotFoundException**
            - javax.faces.el.**PropertyNotFoundException**
            - javax.faces.el.**ReferenceSyntaxException**
  - javax.faces.el.**ValueBinding**
  - javax.faces.el.**VariableResolver**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.xml.stream`

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EventFilter</strong></td>
<td>This interface declares a simple filter interface that one can create to filter XMLEventReaders</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Provides information on the location of an event.</td>
</tr>
<tr>
<td><strong>StreamFilter</strong></td>
<td>This interface declares a simple filter interface that one can create to filter XMLStreamReaders</td>
</tr>
<tr>
<td><strong>XMLEventReader</strong></td>
<td>This is the top level interface for parsing XML Events.</td>
</tr>
<tr>
<td><strong>XMLEventWriter</strong></td>
<td>This is the top level interface for writing XML documents.</td>
</tr>
<tr>
<td><strong>XMLReporter</strong></td>
<td>This interface is used to report non-fatal errors.</td>
</tr>
<tr>
<td><strong>XMLResolver</strong></td>
<td>This interface is used to resolve resources during an XML parse.</td>
</tr>
<tr>
<td><strong>XMLStreamConstants</strong></td>
<td>This interface declares the constants used in this API.</td>
</tr>
<tr>
<td><strong>XMLStreamReader</strong></td>
<td>The XMLStreamReader interface allows forward, read-only access to XML.</td>
</tr>
<tr>
<td><strong>XMLStreamWriter</strong></td>
<td>The XMLStreamWriter interface specifies how to write XML.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>XMLEventFactory</strong></td>
<td>This interface defines a utility class for creating instances of XMLEvents</td>
</tr>
<tr>
<td><strong>XMLInputFactory</strong></td>
<td>Defines an abstract implementation of a factory for getting streams.</td>
</tr>
<tr>
<td><strong>XMLOutputFactory</strong></td>
<td>Defines an abstract implementation of a factory for getting XMLEventWriters and XMLStreamWriters.</td>
</tr>
</tbody>
</table>
Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLStreamException</td>
<td>The base exception for unexpected processing errors.</td>
</tr>
</tbody>
</table>

Error Summary

<table>
<thead>
<tr>
<th>Error Class</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryConfigurationException</td>
<td>An error class for reporting factory configuration errors.</td>
</tr>
</tbody>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Class Hierarchy

- java.lang.Object
  - java.lang.Throwables (implements java.io.Serializable)
    - java.lang.Error
    - javax.xml.stream.FactoryConfigurationError
  - java.lang.Exception
    - javax.xml.stream.XMLStreamException
- javax.xml.stream.XMLEventFactory
- javax.xml.stream.XMLInputFactory
- javax.xml.stream.XMLOutputFactory
Interface Hierarchy

- javax.xml.stream.EventFilter
- java.util.Iterator<E>
  - javax.xml.stream.XMLEventReader
- javax.xml.stream.Location
- javax.xml.stream.StreamFilter
- javax.xml.stream.util.XMLEventConsumer
  - javax.xml.stream.XMLEventWriter
- javax.xml.stream.XMLReporter
- javax.xml.stream.XMLResolver
- javax.xml.stream.XMLStreamConstants
  - javax.xml.stream.XMLStreamReader
- javax.xml.stream.XMLStreamWriter
Package javax.xml.stream.util

Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XMLEventAllocator</td>
<td>This interface defines a class that allows a user to register a way to allocate events given an XMLStreamReader.</td>
</tr>
<tr>
<td>XMLEventConsumer</td>
<td>This interface defines an event consumer interface.</td>
</tr>
</tbody>
</table>

Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EventReaderDelegate</td>
<td>This is the base class for deriving an XMLEventReader filter.</td>
</tr>
<tr>
<td>StreamReaderDelegate</td>
<td>This is the base class for deriving an XMLStreamReader filter. This class is designed to sit between an XMLStreamReader and an application's XMLStreamReader.</td>
</tr>
</tbody>
</table>
Hierarchy For Package javax.xml.stream.util

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.stream.util.**EventReaderDelegate** (implements javax.xml.stream.**XMLEventReader**)
  - javax.xml.stream.util.**StreamReaderDelegate** (implements javax.xml.stream.**XMLStreamReader**)

Interface Hierarchy

- javax.xml.stream.util.XMLEventAllocator
- javax.xml.stream.util.XMLEventConsumer
Package javax.resource.spi.work

This package contains APIs for the work management contract.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Work</td>
<td>This models a Work instance that would be executed by a WorkManager upon submission.</td>
</tr>
<tr>
<td>WorkListener</td>
<td>This models a WorkListener instance which would be notified by the WorkManager when the various Work processing events (work accepted, work rejected, work started, work completed) occur.</td>
</tr>
<tr>
<td>WorkManager</td>
<td>This interface models a WorkManager which provides a facility to submit Work instances for execution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Class Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ExecutionContext</td>
<td>This class models an execution context (transaction, security, etc) with which the Work instance must be executed.</td>
</tr>
<tr>
<td>WorkAdapter</td>
<td>This class is provided as a convenience for easily creating WorkListener instances by extending this class and overriding only those methods of interest.</td>
</tr>
<tr>
<td>WorkEvent</td>
<td>This class models the various events that occur during the processing of a Work instance.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exception Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>WorkCompletedException</td>
<td>This exception is thrown by a WorkManager to indicate that a submitted Work instance has completed with an exception.</td>
</tr>
<tr>
<td>WorkException</td>
<td>A common base class for all work processing related exceptions.</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>WorkRejectedException</td>
<td>This exception is thrown by a WorkManager to indicate that a submitted Work instance has been rejected.</td>
</tr>
</tbody>
</table>
Package javax.resource.spi.work Description

This package contains APIs for the work management contract.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.resource.spi.work

Package Hierarchies:
  All Packages
Class Hierarchy

- java.lang.**Object**
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.resource.spi.work.**WorkEvent**
  - javax.resource.spi.work.**ExecutionContext**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.resource.**ResourceException**
        - javax.resource.spi.work.**WorkException**
          - javax.resource.spi.work.**WorkCompletedException**
          - javax.resource.spi.work.**WorkRejectedException**
  - javax.resource.spi.work.**WorkAdapter** (implements javax.resource.spi.work.**WorkListener**)
## Interface Hierarchy

- java.util.**EventListener**
  - javax.resource.spi.work.**WorkListener**
- java.lang.**Runnable**
  - javax.resource.spi.work.**Work**
- javax.resource.spi.work.**WorkManager**

---

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Class</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.faces.context

Classes and interfaces defining per-request state information.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ExternalContext</strong></td>
</tr>
<tr>
<td><strong>FacesContext</strong></td>
</tr>
<tr>
<td><strong>FacesContextFactory</strong></td>
</tr>
<tr>
<td><strong>ResponseStream</strong></td>
</tr>
<tr>
<td><strong>ResponseWriter</strong></td>
</tr>
<tr>
<td><strong>ResponseWriterWrapper</strong></td>
</tr>
</tbody>
</table>
Package javax.faces.context Description

Classes and interfaces defining per-request state information. The main class in this package is \texttt{FacesContext}, which is the access point for all per-request information, as well as the gateway to several other helper classes.
Hierarchy For Package javax.faces.context

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.context.**ExternalContext**
  - javax.faces.context.**FacesContext**
  - javax.faces.context.**FacesContextFactory**
  - java.io.**OutputStream** (implements java.io.**Closeable**, java.io.**Flushable**)
    - javax.faces.context.**ResponseStream**
  - java.io.**Writer** (implements java.lang.**Appendable**, java.io.**Closeable**, java.io.**Flushable**)
    - javax.faces.context.**ResponseWriter**
      - javax.faces.context.**ResponseWriterWrapper**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.faces

Top level classes for the JavaServer(tm) Faces API.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>FactoryFinder</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exception Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>FacesException</td>
</tr>
</tbody>
</table>
Package javax.faces

Description

Top level classes for the JavaServer(tm) Faces API. The most important class in the package is FactoryFinder, which is the mechanism by which users can override many of the key pieces of the implementation with their own.
Hierarchy For Package javax.faces

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.Object
  - javax.faces.FactoryFinder
- java.lang.ThrowException (implements java.io.Serializable)
  - java.lang.Exception
    - java.lang.RuntimeException
      - javax.faces.FacesException
Package javax.servlet

The javax.servlet package contains a number of classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for an instance of such a class by a conforming servlet container.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Filter</strong></td>
<td>A filter is an object that performs filtering tasks on either the request to a resource (a servlet or static content), or on the response from a resource, or both.</td>
</tr>
<tr>
<td><strong>FilterChain</strong></td>
<td>A FilterChain is an object provided by the servlet container to the developer giving a view into the invocation chain of a filtered request for a resource.</td>
</tr>
<tr>
<td><strong>FilterConfig</strong></td>
<td>A filter configuration object used by a servlet container to pass information to a filter during initialization.</td>
</tr>
<tr>
<td><strong>RequestDispatcher</strong></td>
<td>Defines an object that receives requests from the client and sends them to any resource (such as a servlet, HTML file, or JSP file) on the server.</td>
</tr>
<tr>
<td><strong>Servlet</strong></td>
<td>Defines methods that all servlets must implement.</td>
</tr>
<tr>
<td><strong>ServletConfig</strong></td>
<td>A servlet configuration object used by a servlet container to pass information to a servlet during initialization.</td>
</tr>
</tbody>
</table>
|                  | Defines a set of methods that a
<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ServletContext</td>
<td>A servlet uses to communicate with its servlet container, for example, to get the MIME type of a file, dispatch requests, or write to a log file.</td>
</tr>
<tr>
<td>ServletContextAttributeListener</td>
<td>Implementations of this interface receive notifications of changes to the attribute list on the servlet context of a web application.</td>
</tr>
<tr>
<td>ServletContextListener</td>
<td>Implementations of this interface receive notifications about changes to the servlet context of the web application they are part of.</td>
</tr>
<tr>
<td>ServletRequest</td>
<td>Defines an object to provide client request information to a servlet.</td>
</tr>
<tr>
<td>ServletRequestAttributeListener</td>
<td>A ServletRequestAttributeListener can be implemented by the developer interested in being notified of request attribute changes.</td>
</tr>
<tr>
<td>ServletRequestListener</td>
<td>A ServletRequestListener can be implemented by the developer interested in being notified of requests coming in and out of scope in a web component.</td>
</tr>
<tr>
<td>ServletResponse</td>
<td>Defines an object to assist a servlet in sending a response to the client.</td>
</tr>
<tr>
<td>SingleThreadModel</td>
<td><strong>Deprecated.</strong> As of Java Servlet API 2.4, with no direct replacement.</td>
</tr>
</tbody>
</table>

**Class Summary**

<table>
<thead>
<tr>
<th>Class Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GenericServlet</td>
<td>Defines a generic, protocol-independent servlet.</td>
</tr>
<tr>
<td>ServletContextAttributeEvent</td>
<td>This is the event class for notifications about changes to the attributes of the servlet context of a web application.</td>
</tr>
<tr>
<td><strong>ServletContextEvent</strong></td>
<td>This is the event class for notifications about changes to the servlet context of a web application.</td>
</tr>
<tr>
<td>-------------------------</td>
<td>--------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>ServletInputStream</strong></td>
<td>Provides an input stream for reading binary data from a client request, including an efficient <code>readLine</code> method for reading data one line at a time.</td>
</tr>
<tr>
<td><strong>ServletOutputStream</strong></td>
<td>Provides an output stream for sending binary data to the client.</td>
</tr>
<tr>
<td><strong>ServletRequestAttributeEvent</strong></td>
<td>This is the event class for notifications of changes to the attributes of the servlet request in an application.</td>
</tr>
<tr>
<td><strong>ServletRequestEvent</strong></td>
<td>Events of this kind indicate lifecycle events for a ServletRequest.</td>
</tr>
<tr>
<td><strong>ServletRequestWrapper</strong></td>
<td>Provides a convenient implementation of the ServletRequest interface that can be subclassed by developers wishing to adapt the request to a Servlet.</td>
</tr>
<tr>
<td><strong>ServletResponseWrapper</strong></td>
<td>Provides a convenient implementation of the ServletResponse interface that can be subclassed by developers wishing to adapt the response from a Servlet.</td>
</tr>
</tbody>
</table>

**Exception Summary**

<table>
<thead>
<tr>
<th><strong>ServletException</strong></th>
<th>Defines a general exception a servlet can throw when it encounters difficulty.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>UnavailableException</strong></td>
<td>Defines an exception that a servlet or filter throws to indicate that it is permanently or temporarily unavailable.</td>
</tr>
</tbody>
</table>
Package javax.servlet Description

The javax.servlet package contains a number of classes and interfaces that describe and define the contracts between a servlet class and the runtime environment provided for an instance of such a class by a conforming servlet container.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.servlet

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.util.**EventObject** (implements java.io.**Serializable**)
    - javax.servlet.**ServletContextEvent**
    - javax.servlet.**ServletContextAttributeEvent**
    - javax.servlet.**ServletRequestEvent**
    - javax.servlet.**ServletRequestAttributeEvent**
  - javax.servlet.**GenericServlet** (implements java.io.**Serializable**, javax.servlet.**Servlet**, javax.servlet.**ServletConfig**)
  - java.io.**InputStream** (implements java.io.**Closeable**)
    - javax.servlet.**ServletInputStream**
  - java.io.**OutputStream** (implements java.io.**Closeable**, java.io.**Flushable**)
    - javax.servlet.**ServletOutputStream**
  - javax.servlet.**ServletRequestWrapper** (implements javax.servlet.**ServletRequest**)
  - javax.servlet.**ServletResponseWrapper** (implements javax.servlet.**ServletResponse**)
- java.lang.**Throwable** (implements java.io.**Serializable**)
  - java.lang.**Exception**
    - javax.servlet.**ServletException**
  - javax.servlet.**UnavailableException**
## Interface Hierarchy

- `java.util.EventListener`
  - `javax.servlet.ServletContextAttributeListener`
  - `javax.servlet.ServletContextListener`
  - `javax.servlet.ServletRequestAttributeListener`
  - `javax.servlet.ServletRequestListener`
- `javax.servlet.Filter`
- `javax.servlet.FilterChain`
- `javax.servlet.FilterConfig`
- `javax.servlet.RequestDispatcher`
- `javax.servlet.ServletException`
- `javax.servlet.ServletException`
- `javax.servlet.SingleThreadModel`

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Package javax.annotation

## Enum Summary

<table>
<thead>
<tr>
<th>Enum Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Resource.AuthenticationType</code></td>
<td>The two possible authentication types for a resource.</td>
</tr>
</tbody>
</table>

## Annotation Types Summary

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>Generated</code></td>
<td>The Generated annotation is used to mark source code that has been generated.</td>
</tr>
<tr>
<td><code>PostConstruct</code></td>
<td>The PostConstruct annotation is used on a method that needs to be executed after dependency injection is done to perform any initialization.</td>
</tr>
<tr>
<td><code>PreDestroy</code></td>
<td>The PreDestroy annotation is used on methods as a callback notification to signal that the instance is in the process of being removed by the container.</td>
</tr>
<tr>
<td><code>Resource</code></td>
<td>The Resource annotation marks a resource that is needed by the application.</td>
</tr>
<tr>
<td><code>Resources</code></td>
<td>This class is used to allow multiple resources declarations.</td>
</tr>
</tbody>
</table>
Hierarchy For Package javax.annotation

Package Hierarchies:
All Packages
Annotation Type Hierarchy

- `javax.annotation.Generated` (implements `java.lang.annotation.Annotation`)
- `javax.annotation.PostConstruct` (implements `java.lang.annotation.Annotation`)
- `javax.annotation.PreDestroy` (implements `java.lang.annotation.Annotation`)
- `javax.annotation.Resource` (implements `java.lang.annotation.Annotation`)
- `javax.annotation.Resources` (implements `java.lang.annotation.Annotation`
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
  - javax.annotation.**Resource.AuthenticationType**
## Package javax.resource.spi.security

The javax.resource.spi.security package contains APIs for the security management contract.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GenericCredential</strong></td>
<td>Deprecated. The preferred way to represent generic credential information is via the org.ietf.jgss.GSSCredential interface in J2SE Version 1.4, which provides similar functionality.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PasswordCredential</strong></td>
<td>The class PasswordCredential acts as a holder for username and password.</td>
</tr>
</tbody>
</table>
Package javax.resource.spi.security Description

The javax.resource.spi.security package contains APIs for the security management contract.
Hierarchy For Package
javax.resource.spi.security

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.resource.spi.security.**PasswordCredential** (implements java.io.**Serializable**)
Interface Hierarchy

- javax.resource.spi.security.GenericCredential

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.xml.rpc.handler

This package defines APIs for SOAP Message Handlers

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Handler</strong></td>
<td>The javax.xml.rpc.handler.Handler interface is required to be implemented by a SOAP message handler.</td>
</tr>
<tr>
<td><strong>HandlerChain</strong></td>
<td>The javax.xml.rpc.handler.HandlerChain represents a list of handlers.</td>
</tr>
<tr>
<td><strong>HandlerRegistry</strong></td>
<td>The javax.xml.rpc.handler.HandlerRegistry provides support for the programmatic configuration of handlers in a HandlerRegistry.</td>
</tr>
<tr>
<td><strong>MessageContext</strong></td>
<td>The interface MessageContext abstracts the message context that is processed by a handler in the handle method.</td>
</tr>
</tbody>
</table>

### Class Summary

<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GenericHandler</strong></td>
<td>The javax.xml.rpc.handler.GenericHandler class implements the Handler interface.</td>
</tr>
<tr>
<td><strong>HandlerInfo</strong></td>
<td>The javax.xml.rpc.handler.HandlerInfo represents information about a handler in the HandlerChain.</td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.handler Description

This package defines APIs for SOAP Message Handlers

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.rpc.handler

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.rpc.handler.**GenericHandler** (implements javax.xml.rpc.handler.**Handler**)
  - javax.xml.rpc.handler.**HandlerInfo** (implements java.io.**Serializable**)

```java
Class Hierarchy

- java.lang.**Object**
  - javax.xml.rpc.handler.**GenericHandler** (implements javax.xml.rpc.handler.**Handler**)
  - javax.xml.rpc.handler.**HandlerInfo** (implements java.io.**Serializable**)
```
Interface Hierarchy

- `javax.xml.rpc.handler.Handler`
- `java.lang.Iterable<T>`
  - `java.util.Collection<E>`
    - `java.util.List<E>`
      - `javax.xml.rpc.handler.HandlerChain`
- `javax.xml.rpc.handler.MessageContext`
- `java.io.Serializable`
  - `javax.xml.rpc.handler.HandlerRegistry`
The javax.ejb.spi package defines interfaces that are implemented by the EJB container.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HandleDelegate</td>
<td>The HandleDelegate interface is implemented by the EJB container.</td>
</tr>
</tbody>
</table>
Package javax.ejb.spi Description

The javax.ejb.spi package defines interfaces that are implemented by the EJB container. These interfaces are not used by application components.
Hierarchy For Package javax.ejb.spi

Package Hierarchies:
All Packages
## Interface Hierarchy

- javax.ejb.spi.HandleDelegate

<table>
<thead>
<tr>
<th>Overview</th>
<th>Package</th>
<th>Class</th>
<th>Deprecated</th>
<th>Index</th>
<th>Help</th>
</tr>
</thead>
</table>

[Submit a bug or feature](#)
Package javax.xml.ws.handler

This package defines APIs for message handlers.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Handler&lt;C extends MessageContext&gt;</td>
<td>The Handler interface is the base interface for JAX-WS handlers.</td>
</tr>
<tr>
<td>HandlerResolver</td>
<td>HandlerResolver is an interface implemented by an application to get control over the handler chain set on proxy/dispatch objects at the time of their creation.</td>
</tr>
<tr>
<td>LogicalHandler&lt;C extends LogicalMessageContext&gt;</td>
<td>The LogicalHandler extends Handler to provide typesafety for the message context parameter.</td>
</tr>
<tr>
<td>LogicalMessageContext</td>
<td>The LogicalMessageContext interface extends MessageContext to provide access to a the contained message as a protocol neutral LogicalMessage</td>
</tr>
<tr>
<td>MessageContext</td>
<td>The interface MessageContext abstracts the message context that is processed by a handler in the handle method.</td>
</tr>
<tr>
<td>PortInfo</td>
<td>The PortInfo interface is used by a HandlerResolver to query information about the port it is being asked to create a handler chain for.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Enum Summary</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MessageContext.Scope</td>
<td>Property scope.</td>
</tr>
</tbody>
</table>
This package defines APIs for message handlers.
Hierarchy For Package javax.xml.ws.handler

Package Hierarchies:
  All Packages
Interface Hierarchy

- javax.xml.ws.handler.Handler<C>
  - javax.xml.ws.handler.LogicalHandler<C>
- javax.xml.ws.handler.HandlerResolver
- java.util.Map<K,V>
  - javax.xml.ws.handler.MessageContext
    - javax.xml.ws.handler.LogicalMessageContext
- javax.xml.ws.handler.PortInfo
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
  - javax.xml.ws.handler.**MessageContext.Scope**
Package javax.jws

Enum Summary

| WebParam.Mode | The direction in which the parameter flows |

Annotation Types Summary

| HandlerChain | Associates the Web Service with an externally defined handler chain. |
| Oneway | Indicates that the given @WebMethod has only an input message and no output. |
| WebMethod | Customizes a method that is exposed as a Web Service operation. |
| WebParam | Customizes the mapping of an individual parameter to a Web Service message part and XML element. |
| WebResult | Customizes the mapping of the return value to a WSDL part and XML element. |
| WebService | Marks a Java class as implementing a Web Service, or a Java interface as defining a Web Service interface. |
Hierarchy For Package javax.jws

Package Hierarchies:

All Packages
Annotation Type Hierarchy

- javax.jws.HandlerChain (implements java.lang.annotation.Annotation)
- javax.jws.Oneway (implements java.lang.annotation.Annotation)
- javax.jws.WebMethod (implements java.lang.annotation.Annotation)
- javax.jws.WebParam (implements java.lang.annotation.Annotation)
- javax.jws.WebResult (implements java.lang.annotation.Annotation)
- javax.jws.WebService (implements java.lang.annotation.Annotation)
Enum Hierarchy

○ java.lang.**Object**
  ○ java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
    ○ javax.jws.**WebParam.Mode**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package `javax.transaction`

Provides the API that defines the contract between the transaction manager and the various parties involved in a distributed transaction namely: resource manager, application, and application server.

See: [Description](#)

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td>The Status interface defines static variables used for transaction status codes.</td>
</tr>
<tr>
<td><strong>Synchronization</strong></td>
<td>The transaction manager supports a synchronization mechanism that allows the interested party to be notified before and after the transaction completes.</td>
</tr>
<tr>
<td><strong>Transaction</strong></td>
<td>The Transaction interface allows operations to be performed against the transaction in the target Transaction object.</td>
</tr>
<tr>
<td><strong>TransactionManager</strong></td>
<td>The TransactionManager interface defines the methods that allow an application server to manage transaction boundaries.</td>
</tr>
<tr>
<td><strong>TransactionSynchronizationRegistry</strong></td>
<td>This interface is intended for use by system level application server components such as persistence managers, resource adapters, as well as EJB and Web application components.</td>
</tr>
<tr>
<td><strong>UserTransaction</strong></td>
<td>The UserTransaction interface</td>
</tr>
</tbody>
</table>
**UserTransaction**
defines the methods that allow an application to explicitly manage transaction boundaries.

### Exception Summary

<table>
<thead>
<tr>
<th>Exception Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HeuristicCommitException</td>
<td>This exception is thrown by the rollback operation on a resource to report that a heuristic decision was made and that all relevant updates have been committed.</td>
</tr>
<tr>
<td>HeuristicMixedException</td>
<td>This exception is thrown to report that a heuristic decision was made and that some relevant updates have been committed and others have been rolled back.</td>
</tr>
<tr>
<td>HeuristicRollbackException</td>
<td>This exception is thrown by the commit operation to report that a heuristic decision was made and that all relevant updates have been rolled back.</td>
</tr>
<tr>
<td>InvalidTransactionException</td>
<td>This exception indicates that the request carried an invalid transaction context.</td>
</tr>
<tr>
<td>NotSupportedException</td>
<td>NotSupportedException exception indicates that the request cannot be executed because the operation is not a supported feature.</td>
</tr>
<tr>
<td>RollbackException</td>
<td>RollbackException exception is thrown when the transaction has been marked for rollback only or the transaction has been rolled back instead of committed.</td>
</tr>
<tr>
<td>SystemException</td>
<td>The SystemException is thrown by the transaction manager to indicate that it has encountered an error.</td>
</tr>
<tr>
<td>Exception</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SystemException</strong></td>
<td>unexpected error condition that prevents future transaction services from proceeding.</td>
</tr>
<tr>
<td><strong>TransactionRequiredException</strong></td>
<td>This exception indicates that a request carried a null transaction context, but the target object requires an activate transaction.</td>
</tr>
<tr>
<td><strong>TransactionRolledbackException</strong></td>
<td>This exception indicates that the transaction associated with processing of the request has been rolled back, or marked to roll back.</td>
</tr>
</tbody>
</table>
Package javax.transaction Description

Provides the API that defines the contract between the transaction manager and the various parties involved in a distributed transaction namely: resource manager, application, and application server. The implementation of this API is provided by the application server vendor and the resource manager driver vendor.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.transaction

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.transaction.**HeuristicCommitException**
      - javax.transaction.**HeuristicMixedException**
      - javax.transaction.**HeuristicRollbackException**
      - java.io.**IOException**
      - java.rmi.**RemoteException**
        - javax.transaction.**InvalidTransactionException**
        - javax.transaction.**TransactionRequiredException**
        - javax.transaction.**TransactionRolledbackException**
      - javax.transaction.**NotSupportedException**
      - javax.transaction.**RollbackException**
      - javax.transaction.**SystemException**
Interface Hierarchy

- javax.transaction.Status
- javax.transaction.Synchronization
- javax.transaction.Transaction
- javax.transaction.TransactionManager
- javax.transaction.TransactionSynchronizationRegistry
- javax.transaction.UserTransaction
Package javax.faces.component.html

Specialized user interface component classes for HTML.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HtmlColumn</strong></td>
<td>Represents a column that will be rendered in an HTML table element.</td>
</tr>
<tr>
<td><strong>HtmlCommandButton</strong></td>
<td>Represents an HTML input element for a button of type submit or reset.</td>
</tr>
<tr>
<td><strong>HtmlCommandLink</strong></td>
<td>Represents an HTML a element for a hyperlink that acts like a submit button.</td>
</tr>
<tr>
<td><strong>HtmlDataTable</strong></td>
<td>Represents a set of repeating data (segregated into columns by child UIColumn components) that will be rendered in an HTML table element.</td>
</tr>
<tr>
<td><strong>HtmlForm</strong></td>
<td>Represents an HTML form element.</td>
</tr>
<tr>
<td><strong>HtmlGraphicImage</strong></td>
<td>Represents an HTML img element, used to retrieve and render a graphical image.</td>
</tr>
<tr>
<td><strong>HtmlInputHidden</strong></td>
<td>Represents an HTML input element of type hidden.</td>
</tr>
<tr>
<td><strong>HtmlInputSecret</strong></td>
<td>Represents an HTML input element of type password.</td>
</tr>
<tr>
<td><strong>HtmlInputText</strong></td>
<td>Represents an HTML input element of type text.</td>
</tr>
<tr>
<td><strong>HtmlInputTextarea</strong></td>
<td>Represents an HTML textarea element.</td>
</tr>
<tr>
<td><strong>HtmlMessage</strong></td>
<td>By default, the rendererType property must be set to &quot;javax.faces.Message&quot;.</td>
</tr>
<tr>
<td><strong>HtmlMessages</strong></td>
<td>By default, the rendererType property must be set to &quot;javax.faces.Messages&quot;.</td>
</tr>
<tr>
<td>HTMLComponent</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>HtmlOutputFormat</td>
<td>Represents a component that looks up a localized message in a resource bundle, optionally uses it as a MessageFormat pattern string and substitutes in parameter values from nested UIParameter components, and renders the result.</td>
</tr>
<tr>
<td>HtmlOutputLabel</td>
<td>Represents an HTML label element, used to define an accessible label for a corresponding input element.</td>
</tr>
<tr>
<td>HtmlOutputLink</td>
<td>Represents an HTML a (hyperlink) element that may be used to link to an arbitrary URL defined by the value property.</td>
</tr>
<tr>
<td>HtmlOutputText</td>
<td>Renders the component value as text, optionally wrapping in a span element if I18N attributes, CSS styles or style classes are specified.</td>
</tr>
<tr>
<td>HtmlPanelGrid</td>
<td>Renders child components in a table, starting a new row after the specified number of columns.</td>
</tr>
<tr>
<td>HtmlPanelGroup</td>
<td>Causes all child components of this component to be rendered.</td>
</tr>
<tr>
<td>HtmlSelectBooleanCheckbox</td>
<td>Represents an HTML input element of type checkbox.</td>
</tr>
<tr>
<td>HtmlSelectManyCheckbox</td>
<td>Represents a multiple-selection component that is rendered as a set of HTML input elements of type checkbox.</td>
</tr>
<tr>
<td>HtmlSelectManyListbox</td>
<td>Represents a multiple-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.</td>
</tr>
</tbody>
</table>
| HtmlSelectManyMenu | Represents a multiple-selection component that is rendered as an HTML select element, showing a single
<table>
<thead>
<tr>
<th><strong>HtmlSelectOneListbox</strong></th>
<th>Represents a single-selection component that is rendered as an HTML select element, showing either all available options or the specified number of options.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HtmlSelectOneMenu</strong></td>
<td>Represents a single-selection component that is rendered as an HTML select element, showing a single available option at a time.</td>
</tr>
<tr>
<td><strong>HtmlSelectOneRadio</strong></td>
<td>Represents a single-selection component that is rendered as a set of HTML input elements of type radio.</td>
</tr>
</tbody>
</table>
Package javax.faces.component.html

Description

Specialized user interface component classes for HTML.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.faces.component.html

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.component.**UIComponent** (implements javax.faces.component.**StateHolder**)
    - javax.faces.component.**UIComponentBase**
      - javax.faces.component.**UIColumn**
        - javax.faces.component.html.**HtmlColumn**
      - javax.faces.component.**UICommand** (implements javax.faces.component.**ActionSource2**)
        - javax.faces.component.html.**HtmlCommandButton**
        - javax.faces.component.html.**HtmlCommandLink**
    - javax.faces.component.**UIData** (implements javax.faces.component.**NamingContainer**)
      - javax.faces.component.html.**HtmlDataTable**
    - javax.faces.component.**UIForm** (implements javax.faces.component.**NamingContainer**)
      - javax.faces.component.html.**HtmlForm**
    - javax.faces.component.**UIGraphic**
      - javax.faces.component.html.**HtmlGraphicImage**
    - javax.faces.component.**UIMessage**
      - javax.faces.component.html.**HtmlMessage**
    - javax.faces.component.**UIMessages**
      - javax.faces.component.html.**HtmlMessages**
    - javax.faces.component.**UIOutput** (implements javax.faces.component.**ValueHolder**)
      - javax.faces.component.html.**HtmlOutputFormat**
      - javax.faces.component.html.**HtmlOutputLabel**
      - javax.faces.component.html.**HtmlOutputLink**
      - javax.faces.component.html.**HtmlOutputText**
    - javax.faces.component.**UIInput** (implements javax.faces.component.**EditableValueHolder**)
      - javax.faces.component.html.**HtmlInputHidden**
      - javax.faces.component.html.**HtmlInputSecret**
      - javax.faces.component.html.**HtmlInputText**
      - javax.faces.component.html.**HtmlInputTextarea**
      - javax.faces.component.**UISelectBoolean**
○ javax.faces.component.html.HtmlSelectB
○ javax.faces.component.UISelectMany
○ javax.faces.component.html.HtmlSelectM
○ javax.faces.component.html.HtmlSelectM
○ javax.faces.component.html.HtmlSelectM
○ javax.faces.component.UISelectOne
○ javax.faces.component.html.HtmlSelectO
○ javax.faces.component.html.HtmlSelectO
○ javax.faces.component.html.HtmlSelectO
○ javax.faces.component.html.HtmlSelectO
○ javax.faces.component.UIPanel
○ javax.faces.component.html.HtmlPanelGrid
○ javax.faces.component.html.HtmlPanelGroup
Package `javax.xml.ws.http`

This package defines APIs specific to the HTTP binding.

See: [Description](#)

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPBinding</td>
<td>The <code>HTTPBinding</code> interface is an abstraction for the XML/HTTP binding.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTPException</td>
<td>The <code>HTTPException</code> exception represents a XML/HTTP fault.</td>
</tr>
</tbody>
</table>
Package javax.xml.ws.http Description

This package defines APIs specific to the HTTP binding.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.ws.http

Package Hierarchies:
  All Packages
Class Hierarchy

- java.lang.\textbf{Object}
  - java.lang.\textbf{Throwable} (implements java.io.\textbf{Serializable})
    - java.lang.\textbf{Exception}
      - java.lang.\textbf{RuntimeException}
        - javax.xml.ws.\textbf{WebServiceException}
          - javax.xml.ws.\textbf{ProtocolException}
            - javax.xml.ws.http.\textbf{HTTPException}
Interface Hierarchy

- javax.xml.ws.Binding
  - javax.xml.ws.http.HTTPBinding
### Enum Summary

<table>
<thead>
<tr>
<th>Enum Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SOAPBinding.ParameterStyle</code></td>
<td>The style of mapping parameters onto SOAP messages</td>
</tr>
<tr>
<td><code>SOAPBinding.Style</code></td>
<td>The SOAP binding style</td>
</tr>
<tr>
<td><code>SOAPBinding.Use</code></td>
<td>The SOAP binding use</td>
</tr>
</tbody>
</table>

### Annotation Types Summary

<table>
<thead>
<tr>
<th>Annotation Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>InitParam</code></td>
<td>Deprecated. As of JSR-181 2.0 with no replacement.</td>
</tr>
<tr>
<td><code>SOAPBinding</code></td>
<td>Specifies the mapping of the Web Service onto the SOAP message protocol.</td>
</tr>
<tr>
<td><code>SOAPMessageHandler</code></td>
<td>Deprecated. As of JSR-181 2.0 with no replacement.</td>
</tr>
<tr>
<td><code>SOAPMessageHandlers</code></td>
<td>Deprecated. As of JSR-181 2.0 with no replacement.</td>
</tr>
</tbody>
</table>

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package javax.jws.soap

Package Hierarchies:
   All Packages
Annotation Type Hierarchy

- javax.jws.soap.InitParam (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPBinding (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPMessageHandler (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPMessageHandlers (implements java.lang.annotation.Annotation)
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
    - javax.jws.soap.**SOAPBinding.Style**
    - javax.jws.soap.**SOAPBinding.Use**
    - javax.jws.soap.**SOAPBinding.ParameterStyle**

---

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.xml.bind.util

Useful client utility classes.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAXBResult</td>
</tr>
<tr>
<td>JAXBSource</td>
</tr>
<tr>
<td>ValidationEventCollector</td>
</tr>
</tbody>
</table>
Package javax.xml.bind.util Description

Useful client utility classes.
Package Specification

- JAXB Specification
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- The [JAXB Website](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package javax.xml.bind.util

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.xml.transform.sax.**SAXResult** (implements javax.xml.transform.**Result**)
    - javax.xml.bind.util.**JAXBResult**
  - javax.xml.transform.sax.**SAXSource** (implements javax.xml.transform.**Source**)
    - javax.xml.bind.util.**JAXBSource**
  - javax.xml.bind.util.**ValidationEventCollector** (implements javax.xml.bind.**ValidationEventHandler**)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.faces.lifecycle

Classes and interfaces defining lifecycle management for the JavaServer Faces implementation.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Lifecycle</strong></td>
</tr>
<tr>
<td><strong>LifecycleFactory</strong></td>
</tr>
</tbody>
</table>
Package javax.faces.lifecycle Description

Classes and interfaces defining lifecycle management for the JavaServer Faces implementation. The main class in this package is Lifecycle. Lifecycle is the gateway to executing the request processing lifecycle.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.faces.lifecycle

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - javax.faces.lifecycle.**Lifecycle**
  - javax.faces.lifecycle.**LifecycleFactory**
Package `javax.management.j2ee`

Provides the J2EE Management Enterprise Bean component (MEJB) interfaces.

See: [Description](#)

<table>
<thead>
<tr>
<th>Interface Summary</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ListenerRegistration</strong></td>
<td>ListenerRegistration defines the methods which clients of the MEJB use to add and remove event listeners.</td>
</tr>
<tr>
<td><strong>Management</strong></td>
<td>The Management interface provides the APIs to navigate and manipulate managed objects.</td>
</tr>
<tr>
<td><strong>ManagementHome</strong></td>
<td>The required home interface for the J2EE Management EJB component (MEJB).</td>
</tr>
</tbody>
</table>
Package javax.management.j2ee Description

Provides the J2EE Management Enterprise Bean component (MEJB) interfaces.
Package Specification

- JSR 77, J2EE Management
Related Documentation

For overviews, tutorials, examples, guides, and tool documentation, please see:

- [J2EE Tools](#)

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](#).
Hierarchy For Package javax.management.j2ee

Package Hierarchies:

All Packages
Interface Hierarchy

- java.rmi.Remote
  - javax.ejb.EJBHome
    - javax.management.j2ee.ManagementHome
  - javax.ejb.EJBObject
    - javax.management.j2ee.Management
- java.io.Serializable
  - javax.management.j2ee.ListenerRegistration

Overview  Package  Class  Deprecated  Index  Help

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Package javax.resource.spi.endpoint

This package contains system contracts for service endpoint interactions.

See: Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MessageEndpoint</strong></td>
</tr>
<tr>
<td><strong>MessageEndpointFactory</strong></td>
</tr>
</tbody>
</table>
Package javax.resource.spi.endpoint

Description

This package contains system contracts for service endpoint interactions.
Hierarchy For Package
javax.resource.spi.endpoint

Package Hierarchies:
All Packages
Interface Hierarchy

- javax.resource.spi.endpoint.MessageEndpoint
- javax.resource.spi.endpoint.MessageEndpointFactory
Package javax.resource

The javax.resource package is the top-level package for the J2EE Connector API specification.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Referenceable</td>
<td>The Referenceable interface extends the javax.naming.Referenceable interface.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NotSupportedException</td>
<td>A NotSupportedException is thrown to indicate that callee (resource adapter or application server for system contracts) cannot execute an operation because the operation is not a supported feature.</td>
</tr>
<tr>
<td>ResourceException</td>
<td>This is the root interface of the exception hierarchy defined for the Connector architecture.</td>
</tr>
</tbody>
</table>
Package javax.resource Description

The javax.resource package is the top-level package for the J2EE Connector API specification.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.resource

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.resource.**ResourceException**
      - javax.resource.**NotSupportedException**
Interface Hierarchy

- javax.naming.Referenceable
  - javax.resource.Referenceable
Package javax.xml.ws.spi

This package defines SPIs for JAX-WS 2.0.

See: Description

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Provider</strong></td>
</tr>
<tr>
<td><strong>ServiceDelegate</strong></td>
</tr>
</tbody>
</table>
Package javax.xml.ws.spi Description

This package defines SPIs for JAX-WS 2.0.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.ws.spi

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.Object
  - javax.xml.ws.spi.Provider
  - javax.xml.ws.spi.ServiceDelegate
Package **javax.faces.render**

Classes and interfaces defining the rendering model.

See: [Description](#)

<table>
<thead>
<tr>
<th>Class Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Renderer</strong></td>
</tr>
<tr>
<td>A <strong>Renderer</strong> converts the internal representation of <strong>UIComponent</strong> s into the output stream (or writer) associated with the response we are creating for a particular request.</td>
</tr>
<tr>
<td><strong>RenderKit</strong></td>
</tr>
<tr>
<td><strong>RenderKit</strong> represents a collection of <strong>Renderer</strong> instances that, together, know how to render JavaServer Faces <strong>UIComponent</strong> instances for a specific client.</td>
</tr>
<tr>
<td><strong>RenderKitFactory</strong></td>
</tr>
<tr>
<td><strong>RenderKitFactory</strong> is a factory object that registers and returns <strong>RenderKit</strong> instances.</td>
</tr>
<tr>
<td><strong>ResponseStateManager</strong></td>
</tr>
<tr>
<td><strong>ResponseStateManager</strong> is the helper class to <strong>StateManager</strong> that knows the specific rendering technology being used to generate the response.</td>
</tr>
</tbody>
</table>
Package javax.faces.render Description

Classes and interfaces defining the rendering model. The main class in this package is RenderKit. RenderKit vends a set of Renderer instances which provide rendering capability for a specific client device type.
Hierarchy For Package javax.faces.render

Package Hierarchies:
   All Packages
Class Hierarchy

- java.lang.**Object**
- javax.faces.render.**Renderer**
- javax.faces.render.**RenderKit**
- javax.faces.render.**RenderKitFactory**
- javax.faces.render.**ResponseStateManager**

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
javax.faces.application Class StateManager.SerializedView

java.lang.Object
  ↓ javax.faces.application.StateManager.SerializedView

Enclosing class: StateManager

---

**Deprecated.** *This class was not marked* Serializable *in the 1.0 version of the spec. It was also not a static inner class, so it can't be made to be Serializable. Therefore, it is being deprecated in version 1.2 of the spec. The replacement is to use an implementation dependent object.*

**Contained within:** StateManager

---

struct API  JSP  JSP

deprecated 1.0

Serializable  Serializable

Object

---

public class StateManager.SerializedView

extends Object

Convenience struct for encapsulating tree structure and component state. This is necessary to allow the API to be flexible enough to work in JSP and non-JSP environments.

---

**Constructor Summary**

| StateManager.SerializedView(Object newStructure, Object newState) | Deprecated. |

---

**Method Summary**
<table>
<thead>
<tr>
<th>Object</th>
<th>getStructure()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Object</th>
<th>getState()</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deprecated.</td>
<td></td>
</tr>
</tbody>
</table>

Methods inherited from class java.lang.Object
- clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

public StateManager.SerializedView(Object newStructure, Object newState)

StateManager.SerializedView

public StateManager.SerializedView(Object newStructure, Object newState)

    Deprecated.

## Method Detail

public Object getStructure()

getStructure

public Object getStructure()

    Deprecated.

public Object getState()
getState

public Object getState()

Deprecated.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.

PS:
Package javax.xml.rpc.server

This package defines APIs for the servlet based JAX-RPC endpoint model.

See: [Description](#)

## Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ServiceLifecycle</strong></td>
<td>The javax.xml.rpc.server.ServiceLifecycle defines a lifecycle interface for a JAX-RPC service endpoint.</td>
</tr>
<tr>
<td><strong>ServletEndpointContext</strong></td>
<td>The ServletEndpointContext provides an endpoint context maintained by the underlying servlet container based JAX-RPC runtime system.</td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.server Description

This package defines APIs for the servlet based JAX-RPC endpoint model.

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.xml.rpc.server

Package Hierarchies:

All Packages
Interface Hierarchy

- javax.xml.rpc.server.ServiceLifecycle
- javax.xml.rpc.server.ServletEndpointContext
Package javax.xml.ws.soap

This package defines APIs specific to the SOAP binding.

See:  Description

<table>
<thead>
<tr>
<th>Interface Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOAPBinding</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Exception Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SOAPFaultException</strong></td>
</tr>
</tbody>
</table>
Package javax.xml.ws.soap Description

This package defines APIs specific to the SOAP binding.
Hierarchy For Package javax.xml.ws.soap

Package Hierarchies:
All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)  
    - java.lang.**Exception**
      - java.lang.**RuntimeException**
        - javax.xml.ws.**WebServiceException**
        - javax.xml.ws.**ProtocolException**
          - javax.xml.ws.soap.**SOAPFaultException**
Interface Hierarchy

- javax.xml.ws.**Binding**
  - javax.xml.ws.soap.**SOAPBinding**
Package javax.xml.rpc.soap

This package defines APIs specific to the SOAP binding.

See: Description

<table>
<thead>
<tr>
<th>Exception Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPFaultException</td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.soap Description

This package defines APIs specific to the SOAP binding.
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
  - java.lang.**Exception**
  - java.lang.**RuntimeException**
  - javax.xml.rpc.soap.**SOAPFaultException**
Package `javax.xml.ws.handler.soap`

This package defines APIs for SOAP message handlers.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>SOAPHandler&lt;T extends SOAPMessageContext&gt;</code></td>
<td>The <code>SOAPHandler</code> class extends <code>Handler</code> to provide typesafety for the message context parameter and add a method to obtain access to the headers that may be processed by the handler.</td>
</tr>
<tr>
<td><code>SOAPMessageContext</code></td>
<td>The interface <code>SOAPMessageContext</code> provides access to the SOAP message for either RPC request or response.</td>
</tr>
</tbody>
</table>
Package javax.xml.ws.handler.soap Description

This package defines APIs for SOAP message handlers.
Hierarchy For Package
javax.xml.ws.handler.soap

Package Hierarchies:
All Packages
Interface Hierarchy

- javax.xml.ws.handler.Handler\<C>
  - javax.xml.ws.handler.soap.SOAPHandler\<T>
- java.util.Map\<K,V>
  - javax.xml.ws.handler.MessageContext
  - javax.xml.ws.handler.soap.SOAPMessageContext
Package javax.xml.rpc.handler.soap

This package defines APIs for SOAP Message Handlers

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOAPMessageContext</td>
<td>The interface <code>javax.xml.rpc.soap.SOAPMessageContext</code> provides access to the SOAP message for either RPC request or response.</td>
</tr>
</tbody>
</table>
Package javax.xml.rpc.handler.soap

Description

This package defines APIs for SOAP Message Handlers

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package
javax.xml.rpc.handler.soap

Package Hierarchies:
All Packages
Interface Hierarchy

- javax.xml.rpc.handler.MessageContext
- javax.xml.rpc.handler.soap.SOAPMessageContext
Package javax.transaction.xa

Provides the API that defines the contract between the transaction manager and the resource manager, which allows the transaction manager to enlist and delist resource objects (supplied by the resource manager driver) in JTA transactions.

See: Description

### Interface Summary

<table>
<thead>
<tr>
<th>Interface</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAResource</td>
<td>The XAResource interface is a Java mapping of the industry standard XA interface based on the X/Open CAE Specification (Distributed Transaction Processing: The XA Specification).</td>
</tr>
<tr>
<td>Xid</td>
<td>The Xid interface is a Java mapping of the X/Open transaction identifier XID structure.</td>
</tr>
</tbody>
</table>

### Exception Summary

<table>
<thead>
<tr>
<th>Exception</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>XAException</td>
<td>The XAException is thrown by the Resource Manager (RM) to inform the Transaction Manager of an error encountered by the involved transaction.</td>
</tr>
</tbody>
</table>
Package javax.transaction.xa Description

Provides the API that defines the contract between the transaction manager and the resource manager, which allows the transaction manager to enlist and delist resource objects (supplied by the resource manager driver) in JTA transactions. The driver vendor for a specific resource manager provides the implementation of this API.

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For Package javax.transaction.xa

Package Hierarchies:

All Packages
Class Hierarchy

- java.lang.**Object**
  - java.lang.**Throwable** (implements java.io.**Serializable**)
    - java.lang.**Exception**
      - javax.transaction.xa.**XAException**
Interface Hierarchy

- javax.transaction.xa.XAResource
- javax.transaction.xa.Xid

Submit a bug or feature

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to license terms.
Hierarchy For All Packages

Package Hierarchies:
javax.activation, javax.annotation, javax.annotation.security,
javax.ejb, javax.ejb.spi, javax.el, javax.enterprise.deploy.model,
javax.enterprise.deploy.model.exceptions,
javax.enterprise.deploy.shared,
javax.enterprise.deploy.shared.factories, javax.enterprise.deploy.spi,
javax.enterprise.deploy.spi.exceptions,
javax.enterprise.deploy.spi.factories,
javax.enterprise.deploy.spi.status, javax.faces,
javax.faces.application, javax.faces.component,
javax.faces.component.html, javax.faces.context,
javax.faces.convert, javax.faces.el, javax.faces.event,
javax.faces.lifecycle, javax.faces.model, javax.faces.render,
javax.faces.validator, javax.faces.webapp, javax.interceptor,
javax.jms, javax.jws, javax.jws.soap, javax.mail, javax.mail.event,
javax.mail.internet, javax.mail.search, javax.mail.util,
javax.management.j2ee, javax.management.j2ee.statistics,
javax.persistence, javax.persistence.spi, javax.resource,
javax.resource.cci, javax.resource.spi, javax.resource.spi.endpoint,
javax.resource.spi.security, javax.resource.spi.work,
javax.security.jacc, javax.servlet, javax.servlet.http, javax.servlet.jsp,
javax.servlet.jsp.el, javax.servlet.jsp.tagext, javax.transaction,
javax.transaction.xa, javax.xml.bind, javax.xml.bind.annotation,
javax.xml.bind.annotation.adapters, javax.xml.bind.attachment,
javax.xml.bind.helpers, javax.xml.bind.util, javax.xml.registry,
javax.xml.registry.infomodel, javax.xml.rpc, javax.xml.rpc.encoding,
javax.xml.rpc.handler, javax.xml.rpc.handler.soap,
javax.xml.rpc.holders, javax.xml.rpc.server, javax.xml.rpc.soap,
javax.xml.soap, javax.xml.stream, javax.xml.stream.events,
javax.xml.stream.util, javax.xml.ws, javax.xml.ws.handler,
javax.xml.ws.handler.soap, javax.xml.ws.http, javax.xml.ws.soap,
javax.xml.ws.spi
Class Hierarchy

- `java.lang.Object`
  - `javax.xml.bind.helpers.AbstractMarshallerImpl` (implements `javax.xml.bind.Marshaller`)
  - `javax.xml.bind.helpers.AbstractUnmarshallerImpl` (implements `javax.xml.bind.Unmarshaller`)
  - `javax.xml.bind.attachment.AttachmentMarshaller`
  - `javax.xml.bind.attachment.AttachmentUnmarshaller`
  - `javax.mail.Authenticator`
  - `javax.el.BeanELResolver.BeanProperties`
  - `javax.el.BeanELResolver.BeanProperty`
  - `javax.faces.convert.BigDecimalConverter` (implements `javax.faces.convert.Converter`)
  - `javax.xml.rpc.holders.BigDecimalHolder` (implements `javax.xml.rpc.holders.Holder`)
  - `javax.faces.convert.BigIntegerConverter` (implements `javax.faces.convert.Converter`)
  - `javax.xml.rpc.holders.BigIntegerHolder` (implements `javax.xml.rpc.holders.Holder`)
  - `javax.xml.bind.Binder<XmlNode>`
  - `javax.mail.BodyPart` (implements `javax.mail.Part`)
    - `javax.mail.internet.MimeBodyPart` (implements `javax.mail.internet.MimePart`)
    - `javax.mail.internet.PreencodedMimeBodyPart`
  - `javax.xml.rpc.holders.BooleanHolder` (implements `javax.xml.rpc.holders.Holder`)
  - `javax.mail.internet.Address` (implements `java.io.Serializable`)
    - `javax.mail.internet.InternetAddress` (implements `java.lang.Cloneable`)
    - `javax.mail.internet.NewsAddress`
- javax.xml.rpc.holders.\texttt{BooleanWrapperHolder} (implements javax.xml.rpc.holders.\texttt{Holder})
- javax.mail.util.\texttt{ByteArrayDataSource} (implements javax.activation.\texttt{DataSource})
- javax.xml.rpc.holders.\texttt{ByteArrayHolder} (implements javax.xml.rpc.holders.\texttt{Holder})
- javax.faces.convert.\texttt{ByteConverter} (implements javax.faces.convert.\texttt{Converter})
- javax.xml.rpc.holders.\texttt{ByteHolder} (implements javax.xml.rpc.holders.\texttt{Holder})
- javax.xml.rpc.holders.\texttt{ByteWrapperHolder} (implements javax.xml.rpc.holders.\texttt{Holder})
- javax.xml.rpc.holders.\texttt{CalendarHolder} (implements javax.xml.rpc.holders.\texttt{Holder})
- javax.faces.convert.\texttt{CharacterConverter} (implements javax.faces.convert.\texttt{Converter})
- javax.activation.\texttt{CommandInfo}
- javax.activation.\texttt{CommandMap}
  - javax.activation.\texttt{MailcapCommandMap}
- javax.enterprise.deploy.shared.\texttt{CommandType}
- javax.mail.event.\texttt{ConnectionAdapter} (implements javax.mail.event.\texttt{ConnectionListener})
- javax.xml.registry.\texttt{ConnectionFactory}
- javax.mail.internet.\texttt{ContentDisposition}
- javax.mail.internet.\texttt{ContentType}
- javax.servlet.http.\texttt{Cookie} (implements java.lang.\texttt{Cloneable})
- java.awt.datatransfer.\texttt{DataFlavor} (implements java.lang.\texttt{Cloneable}, java.io.\texttt{Externalizable})
  - javax.activation.\texttt{ActivationDataFlavor}
- javax.activation.\texttt{DataHandler} (implements java.awt.datatransfer.\texttt{Transferable})
- javax.faces.model.\texttt{DataModel}
  - javax.faces.model.\texttt{ArrayDataModel}
  - javax.faces.model.\texttt{ListDataModel}
  - javax.faces.model.\texttt{ResultDataModel}
  - javax.faces.model.\texttt{ResultSetDataModel}
  - javax.faces.model.\texttt{ScalarDataModel}
- javax.xml.bind.\texttt{DatatypeConverter}
- javax.faces.convert.\texttt{DateTimeConverter} (implements
javax.faces.convert.Converter,
javax.faces.component.StateHolder

javax.enterprise.deploy.shared.DConfigBeanVersionType

javax.xml.bind.helpers.DefaultValidationEventHandler (implements javax.xml.bind.ValidationEventHandler)

javax.enterprise.deploy.shared.factories.DeploymentFactoryManager

javax.xml.transform.dom.DOMResult (implements javax.xml.transform.Result)
  ○ javax.xml.soap.SAAJResult

javax.faces.convert.DoubleConverter (implements javax.faces.convert.Converter)

javax.xml.rpc.holders.DoubleHolder (implements javax.xml.rpc.holders.Holder)

javax.faces.validator.DoubleRangeValidator (implements javax.faces.component.StateHolder,
javax.faces.validator.Validator)

javax.xml.rpc.holders.DoubleWrapperHolder (implements javax.xml.rpc.holders.Holder)

javax.el.ELContext

javax.el.ELResolver
  ○ javax.el.ArrayELResolver
  ○ javax.el.BeanELResolver
  ○ javax.el.CompositeELResolver
  ○ javax.servlet.jsp.el.ImplicitObjectELResolver
  ○ javax.el.ListELResolver
  ○ javax.el.MapELResolver
  ○ javax.el.ResourceBundleELResolver
  ○ javax.servlet.jsp.el.ScopedAttributeELResolver

javax.xml.ws.Endpoint

javax.faces.convert.EnumConverter (implements javax.faces.convert.Converter,
javax.faces.component.StateHolder)

javax.servlet.jsp.ErrorData

java.util.EventObject (implements java.io.Serializable)
  ○ javax.resource.spi.ConnectionEvent
  ○ javax.faces.model.DataModelEvent
  ○ javax.el.ELContextEvent
  ○ javax.faces.event.FacesEvent
    ○ javax.faces.event.ActionEvent

javax.faces.event.ValueChangeEvent
javax.servlet.http.HttpSessionEvent
javax.servlet.http.HttpSessionBindingEvent
javax.mail.event.MailEvent
javax.mail.event.ConnectionEvent
javax.mail.event.FolderEvent
javax.mail.event.MessageChangedEvent
javax.mail.event.MessageCountEvent
javax.mail.event.StoreEvent
javax.mail.event.TransportEvent
javax.faces.event.PhaseEvent
javax.enterprise.deploy.spi.status.ProgressEvent
javax.servlet.ServletContextEvent
javax.servlet.ServletContextAttributeEvent
javax.servlet.ServletRequestEvent
javax.servlet.ServletRequestAttributeEvent
javax.resource.spi.work.WorkEvent
javax.xml.stream.util.EventReaderDelegate (implements javax.xml.stream.XMLEventReader)
javax.resource.spi.work.ExecutionContext
javax.el.Expression (implements java.io.Serializable)
javax.el.MethodExpression
javax.el.ValueExpression
javax.servlet.jsp.el.Expression
javax.servlet.jsp.el.ExpressionEvaluator
javax.el.ExpressionFactory
javax.faces.context.ExternalContext
javax.faces.context.FacesContext
javax.faces.context.FacesContextFactory
javax.faces.application.FacesMessage (implements java.io.Serializable)
javax.faces.application.FacesMessage.Severity (implements java.lang.Comparable<T>)
javax.faces.webapp.FacesServlet (implements javax.servlet.Servlet)
javax.faces.FactoryFinder
javax.mail.FetchProfile
javax.mail.FetchProfile.Item
javax.mail.UIDFolder.FetchProfileItem
javax.activation.
FileDataSource (implements javax.activation.
DataSource)
javax.activation.
FileTypeMap
  ◦ javax.activation.
MimeTypesFileTypeMap
javax.mail.
Flags (implements java.lang.
Cloneable, java.io.
Serializable)
javax.mail.
Flags.Flag
javax.faces.convert.
FloatConverter (implements javax.faces.convert.
Converter)
javax.xml.rpc.holders.
FloatHolder (implements javax.xml.rpc.holders.
Holder)
javax.xml.rpc.holders.
FloatWrapperHolder (implements javax.xml.rpc.holders.
Holder)
javax.mail.
Folder
javax.mail.event.
FolderAdapter (implements javax.mail.event.
FolderListener)
java.text.
Format (implements java.lang.
Cloneable, java.io.
Serializable)
  ◦ java.text.
DateFormat
    ◦ java.text.
SimpleDateFormat
      ◦ javax.mail.internet.
MailDateFormat
javax.servlet.jsp.tagext.
FunctionInfo
javax.el.
FunctionMapper
javax.xml.rpc.handler.
GenericHandler (implements javax.xml.rpc.handler.
Handler)
javax.servlet.
GenericServlet (implements java.io.
Serializable, javax.servlet.
Servlet, javax.servlet.
ServletConfig)
  ◦ javax.servlet.http.
HttpServletRequest (implements java.io.
Serializable)
javax.xml.rpc.handler.
HandlerInfo (implements java.io.
Serializable)
javax.mail.
Header
  ◦ javax.mail.internet.
InternetHeaders.InternetHeader
javax.mail.internet.
HeaderTokenizer
javax.mail.internet.
HeaderTokenizer.Token
javax.xml.ws.
Holder<T>
javax.servlet.http.
HttpServletRequest
java.io.
InputStream (implements java.io.
Closeable)
  ◦ java.io.
ByteArrayInputStream
- `javax.mail.util.SharedByteArrayInputStream` (implements `javax.mail.internet.SharedInputStream`)
- `java.io.FilterInputStream`
- `java.io.BufferedInputStream`
- `javax.mail.util.SharedFileInputStream` (implements `javax.mail.internet.SharedInputStream`)
- `javax.servlet.ServletInputStream`
- `javax.faces.convert.IntegerConverter` (implements `javax.faces.convert.Converter`)
- `javax.xml.rpc.holders.IntegerWrapperHolder` (implements `javax.xml.rpc.holders.Holder`)
- `javax.mail.internet.InternetHeaders`
- `javax.xml.rpc.holders.IntHolder` (implements `javax.xml.rpc.holders.Holder`)
- `javax.xml.bind.JAXBContext`
- `javax.xml.bind.JAXBElement<T>` (implements `java.io.Serializable`)
- `javax.xml.bind.JAXBIntrospector`
- `javax.servlet.jsp.JspContext`
- `javax.servlet.jsp.PageContext`
- `javax.servlet.jsp.JspEngineInfo`
- `javax.servlet.jsp.JspFactory`
- `javax.servlet.jsp.tagext.JspFragment`
- `javax.faces.validator.LengthValidator` (implements `javax.faces.component.StateHolder`, `javax.faces.validator.Validator`)
- `javax.faces.lifecycle.Lifecycle`
- `javax.faces.lifecycle.LifecycleFactory`
- `javax.faces.convert.LongConverter` (implements `javax.faces.convert.Converter`)
- `javax.xml.rpc.holders.LongWrapperHolder` (implements `javax.xml.rpc.holders.Holder`)
- `javax.xml.rpc.holders.LongHolder` (implements `javax.xml.rpc.holders.Holder`)
- `javax.faces.lifecycle.LifecycleFactory`
- javax.xml.bind.Marshaller.Listener
- javax.mail.Message (implements javax.mail.Part)
  - javax.mail.internet.MimeMessage (implements javax.mail.internet.MimePart)
- javax.mail.Message.RecipientType (implements java.io.Serializable)
  - javax.mail.internet.MimeMessage.RecipientType
- javax.mail.MessageContext
- javax.mail.event.MessageCountAdapter (implements javax.mail.event.MessageCountListener)
- javax.xml.soap.MessageFactory
- javax.faces.el.MethodBinding
- javax.el.MethodInfo
- javax.xml.soap.MimeHeader
- javax.xml.soap.MimeHeaders
- javax.mail.internet.MimePartDataSource (implements javax.activation.DataSource, javax.mail.MessageAware)
- javax.activation.MimeType (implements java.io.Externalizable)
- javax.activation.MimeTypeParameterList
- javax.mail.internet.MimeUtility
- javax.mail.enterprise.deploy.shared.ModuleType
- javax.mail.Multipart
  - javax.mail.internet.MimeMultipart
- javax.xml.rpc.NamespaceConstants
- javax.faces.application.NamingHandler
  - javax.xml.rpc.holders.ObjectHolder (implements javax.xml.rpc.holders.Holder)
  - `javax.faces.context.ResponseStream`
  - `javax.servlet.ServletOutputStream`
- `javax.servlet.jsp.tagext.PageData`
- `javax.mail.internet.ParameterList`
- `javax.xml.rpc.ParameterMode`
- `javax.mail.PasswordAuthentication`
- `javax.resource.spi.security.PasswordCredential` (implements `java.io.Serializable`)
- `java.security.Permission` (implements `java.security.Guard`, `java.io.Serializable`)
  - `java.security.BasicPermission` (implements `java.io.Serializable`)
  - `javax.xml.ws.WebServicePermission`
- `javax.security.jacc.EJBMETHOD Permission` (implements `java.io.Serializable`)
- `javax.security.jacc.EJBROLE RefPermission` (implements `java.io.Serializable`)
- `javax.security.jacc.WebResourcePermission` (implements `java.io.Serializable`)
- `javax.security.jacc.WebRoleRefPermission` (implements `java.io.Serializable`)
- `javax.security.jacc.WebUserDataPermission` (implements `java.io.Serializable`)
- `javax.persistence.Persistence`
- `javax.faces.event.PhaseId` (implements `java.lang.Comparable<T>`)
- `javax.security.jacc.PolicyConfigurationFactory`
- `javax.security.jacc.PolicyContext`
- `javax.faces.el.PropertyResolver`
- `javax.xml.ws.spi.Provider`
- `javax.mail.Provider`
- `javax.mail.Provider.Type`
- `javax.xml.rpc.holders.QNameHolder` (implements `javax.xml.rpc.holders.Holder`)
- `javax.jms.QueueRequestor`
- `javax.mail.Quota`
- `javax.mail.Quota.Resource`
- javax.faces.render_Renderer
- javax.faces.render_RenderKit
- javax.faces.render_RenderKitFactory
- javax.faces.render_ResponseStateManager
- javax.xml.soap_SAAJMetaFactory
- javax.xml.transform.sax_SAXResult (implements javax.xml.transform_Result)
  - javax.xml.bind.util_JAXBResult
- javax.xml.transform.sax_SAXSource (implements javax.xml.transform_Source)
  - javax.xml.bind.util_JAXBSource
- javax.xml.bind_SchemaOutputResolver
- javax.mail.search_SearchTerm (implements java.io.Serializable)
  - javax.mail.search_AddressTerm
    - javax.mail.search_FromTerm
    - javax.mail.search_RecipientTerm
  - javax.mail.search_AndTerm
  - javax.mail.search_ComparisonTerm
    - javax.mail.search_DateTerm
      - javax.mail.search_ReceivedDateTerm
      - javax.mail.search_SendDateTerm
    - javax.mail.search_IntegerComparisonTerm
      - javax.mail.search_MessageNumberTerm
      - javax.mail.search_SizeTerm
  - javax.mail.search_FlagTerm
  - javax.mail.search_NotTerm
  - javax.mail.search_OrTerm
  - javax.mail.search_StringTerm
    - javax.mail.search_AddressStringTerm
      - javax.mail.search_FromStringTerm
      - javax.mail.search_RecipientStringTerm
    - javax.mail.search_BodyTerm
    - javax.mail.search_IngredientTerm
    - javax.mail.search_MessageIDTerm
    - javax.mail.search_SubjectTerm
- javax.faces.model_SelectItem (implements java.io.Serializable)
  - javax.faces.model_SelectItemGroup
- javax.xml.ws_Service
- javax.mail_Service
- javax.mail.Store
- javax.mail.Transport
- javax.xml.ws.spi.ServiceDelegate
- javax.xml.rpc.ServiceFactory
- javax.servlet.ServletRequestWrapper (implements javax.servlet.ServletRequest)
- javax.mail.Session
- javax.faces.convert.ShortConverter (implements javax.faces.convert.Converter)
- javax.xml.rpc.holders.ShortHolder (implements javax.xml.rpc.holders.Holder)
- javax.xml.rpc.holders.ShortWrapperHolder (implements javax.xml.rpc.holders.Holder)
- javax.servlet.jsp.tagext.SimpleTagSupport (implements javax.servlet.jsp.tagext.SimpleTag)
- javax.xml.soap.SOAPConnection
- javax.xml.soap.SOAPConnectionFactory
- javax.xml.soap.SOAPElementFactory
- javax.xml.soap.SOAPFactory
- javax.xml.soap.SOAPMessage
- javax.faces.application.StateManager
- javax.faces.application.StateManagerWrapper
- javax.faces.application.StateManager.SerializedView
- javax.enterprise.deploy.shared.StateType
- javax.xml.stream.util.StreamReaderDelegate (implements javax.xml.stream.XMLStreamReader)
- javax.xml.rpc.holders.StringHolder (implements javax.xml.rpc.holders.Holder)
- javax.servlet.jsp.tagext.TagAdapter (implements javax.servlet.jsp.tagext.Tag)
- javax.servlet.jsp.tagext.TagAttributeInfo
- `javax.servlet.jsp.tagext.TagData` (implements `java.lang.Cloneable`)
- `javax.servlet.jsp.tagext.TagExtraInfo`
- `javax.servlet.jsp.tagext.TagFileInfo`
- `javax.servlet.jsp.tagext.TagInfo`
- `javax.servlet.jsp.tagext.TagLibraryInfo`
- `javax.servlet.jsp.tagext.TagLibraryValidator`
- `javax.servlet.jsp.tagext.TagSupport` (implements `javax.servlet.jsp.tagext.IterationTag`, `java.io.Serializable`)
  - `javax.faces.webapp.AttributeTag`
  - `javax.faces.webapp.ConverterELTag`
  - `javax.faces.webapp.ConverterTag`
  - `javax.faces.webapp.FacetTag`
  - `javax.faces.webapp.ValidatorELTag`
  - `javax.faces.webapp.ValidatorTag`
- `javax.servlet.jsp.tagext.TagVariableInfo`
- `java.lang.Throwable` (implements `java.io.Serializable`)
  - `java.lang.Error`
    - `javax.xml.stream.FactoryConfigurationError`
  - `java.lang.Exception`
    - `javax.enterprise.deploy.spi.exceptions.BeanNotFoundException`
    - `javax.enterprise.deploy.spi.exceptions.ClientExecuteException`
    - `javax.enterprise.deploy.spi.exceptions.ConfigurationException`
    - `javax.ejb.CreateException`
    - `javax.ejb.DuplicateKeyException`
    - `javax.enterprise.deploy.spi.exceptions.DConfigBeanVersionUnsupportedException`
    - `javax.enterprise.deploy.model.exceptions.DDBeanCreateException`
    - `javax.enterprise.deploy.spi.exceptions.DeploymentManagerCreationException`
    - `javax.servlet.jsp.el.ELException`
    - `javax.servlet.jsp.el.ELParseException`
    - `javax.ejb.FinderException`
    - `javax.ejb.ObjectNotFoundException`
    - `javax.transaction.HeuristicCommitException`
    - `javax.transaction.HeuristicMixedException`
    - `javax.transaction.HeuristicRollbackException`
    - `javax.enterprise.deploy.spi.exceptions.InvalidModuleException`
    - `java.io.IOException`
- java.rmi.**RemoteException**
- javax.transaction.**InvalidTransactionException**
- javax.transaction.**TransactionRequiredException**
- javax.transaction.**TransactionRolledbackException**
- javax.activation.**UnsupportedDataTypeException**
- javax.xml.bind.**JAXBException**
- javax.xml.bind.**MarshalException**
- javax.xml.bind.**PropertyException**
- javax.xml.bind.**UnmarshalException**
- javax.xml.bind.**ValidationException**
- javax.xml.registry.**JAXRException** (implements javax.xml.registry.**JAXRResponse**)
  - javax.xml.registry.**InvalidRequestException**
  - javax.xml.registry.**RegistryException**
    - javax.xml.registry.**DeleteException**
    - javax.xml.registry.**FindException**
    - javax.xml.registry.**SaveException**
  - javax.xml.registry.**UnexpectedObjectException**
  - javax.xml.registry.**UnsupportedCapabilityException**
- javax.jms.**JMSException**
  - javax.jms.**IllegalStateException**
  - javax.jms.**InvalidClientIDException**
  - javax.jms.**InvalidDestinationException**
  - javax.jms.**InvalidSelectorException**
  - javax.jms.**JMSSecurityException**
  - javax.jms.**MessageEOFException**
  - javax.jms.**MessageFormatException**
  - javax.jms.**MessageNotReadableException**
  - javax.jms.**MessageNotWriteableException**
  - javax.jms.**ResourceAllocationException**
  - javax.jms.**TransactionInProgressException**
  - javax.jms.**TransactionRolledBackException**
- javax.servlet.jsp.**JspException**
  - javax.servlet.jsp.**JspTagException**
  - javax.servlet.jsp.**SkipPageException**
- javax.mail.**MessagingException**
  - javax.mail.**AuthenticationFailedException**
  - javax.mail.**FolderClosedException**
  - javax.mail.**FolderNotFoundException**
- javax.mail.IllegalWriteException
- javax.mail.MessageRemovedException
- javax.mail.MethodNotSupportedException
- javax.mail.NoSuchProviderException
- javax.mail.internet.ParseException
  - javax.mail.internet.AddressException
- javax.mail.ReadOnlyFolderException
- javax.mail.search.SearchException
- javax.mail.SendFailedException
- javax.mail.StoreClosedException
- javax.activation.MimeTypeParseException
- javax.transaction.NotSupportedException
- javax.enterprise.deploy.spi.exceptions.OperationUnsupportedException
- javax.security.jacc.PolicyContextException
- javax.ejb.RemoveException
- javax.resource.ResourceException
  - javax.resource.spi.ApplicationServerInternalException
  - javax.resource.spi.CommException
  - javax.resource.spi.EISSystemException
  - javax.resource.spi.IllegalStateException
  - javax.resource.spi.InvalidPropertyException
  - javax.resource.spi.LocalTransactionException
  - javax.resource.spi.NotSupportedException
  - javax.resource.spi.ResourceAdapterInternalException
  - javax.resource.spi.ResourceAllocationException
  - javax.resource.cci.ResourceWarning
  - javax.resource.spi.SecurityException
  - javax.resource.spi.SharingViolationException
  - javax.resource.spi.UnavailableException
  - javax.resource.spi.work.WorkException
    - javax.resource.spi.work.WorkCompletedException
    - javax.resource.spi.work.WorkRejectedException
- javax.transaction.RollbackException
- java.lang.RuntimeException
- javax.ejb.EJBException
  - javax.ejb.AccessLocalException
  - javax.ejb.ConcurrentAccessException
  - javax.ejb.EJBAccessException
  - javax.ejb.EJBAccessException
  - javax.ejb.EJBTransactionRequiredException
- javax.ejb. EJBTransactionRolledbackException
- javax.ejb. NoSuchEJBException
- javax.ejb. NoSuchEntityException
- javax.ejb. NoSuchObjectLocalException
- javax.ejb. TransactionRequiredLocalException
- javax.ejb. TransactionRolledbackLocalException
- javax.el.ELException
- javax.el. MethodNotFoundException
- javax.el. PropertyNotFoundException
- javax.el. PropertyNotWritableException
- javax.faces. FacesException
- javax.faces.event. AbortProcessingException
- javax.faces.convert. ConverterException
- javax.faces.el. EvaluationException
- javax.faces.el. MethodNotFoundException
- javax.faces.el. PropertyNotFoundException
- javax.faces.el. ReferenceSyntaxException
- javax.faces.validator. ValidatorException
- javax.faces.application. ViewExpiredException
- javax.xml.rpc. JAXRPCException
- javax.persistence. PersistenceException
- javax.persistence. EntityExistsException
- javax.persistence. EntityNotFoundException
- javax.persistence. NonUniqueResultException
- javax.persistence. NoResultException
- javax.persistence. OptimisticLockException
- javax.persistence. RollbackException
- javax.persistence. TransactionRequiredException
- javax.xml.rpc.soap. SOAPFaultException
- javax.xml.bind. TypeConstraintException
- javax.xml.ws. WebServiceException
- javax.xml.ws. ProtocolException
- javax.xml.ws.http. HTTPException
- javax.xml.ws.soap. SOAPFaultException
- javax.xml.rpc. ServiceException
- javax.servlet. ServletException
- javax.servlet. UnavailableException
- javax.xml.soap. SOAPException
- javax.transaction. SystemException
- javax.enterprise.deploy.spi.exceptions.**TargetException**
- javax.transaction.xa.**XAException**
- javax.xml.stream.**XMLStreamException**
- javax.jms.**TopicRequestor**
- javax.mail.event.**TransportAdapter** (implements javax.mail.event.**TransportListener**)
- javax.faces.component.**UIComponent** (implements javax.faces.component.**StateHolder**)
  - javax.faces.component.**UIComponentBase**
    - javax.faces.component.**UIComponent**
      - javax.faces.component.html.**HtmlColumn**
    - javax.faces.component.**UICommand** (implements javax.faces.component.**ActionSource2**)
      - javax.faces.component.html.**HtmlCommandButton**
      - javax.faces.component.html.**HtmlCommandLink**
    - javax.faces.component.**UIData** (implements javax.faces.component.**NamingContainer**)
      - javax.faces.component.html.**HtmlDataTable**
    - javax.faces.component.**UIForm** (implements javax.faces.component.**NamingContainer**)
      - javax.faces.component.html.**HtmlForm**
    - javax.faces.component.**UIGraphic**
      - javax.faces.component.html.**HtmlGraphicImage**
    - javax.faces.component.**UIMessage**
      - javax.faces.component.html.**HtmlMessage**
    - javax.faces.component.**UIMessages**
      - javax.faces.component.html.**HtmlMessages**
    - javax.faces.component.**UINamingContainer** (implements javax.faces.component.**NamingContainer**)
  - javax.faces.component.**UIOutput** (implements javax.faces.component.**ViewHolder**)
    - javax.faces.component.html.**HtmlOutputFormat**
    - javax.faces.component.html.**HtmlOutputLabel**
    - javax.faces.component.html.**HtmlOutputLink**
    - javax.faces.component.html.**HtmlOutputText**
  - javax.faces.component.**UIInput** (implements javax.faces.component.**EditableValueHolder**)
    - javax.faces.component.html.**HtmlInputHidden**
    - javax.faces.component.html.**HtmlInputSecret**
- javax.faces.component.html.Htm1InputText
- javax.faces.component.html.Htm1InputTextarea
- javax.faces.component.UISelectBoolean
  - javax.faces.component.html.Htm1SelectB
- javax.faces.component.UISelectMany
  - javax.faces.component.html.Htm1SelectM
  - javax.faces.component.html.Htm1SelectM
  - javax.faces.component.html.Htm1SelectM
- javax.faces.component.UISelectOne
  - javax.faces.component.html.Htm1SelectO
  - javax.faces.component.html.Htm1SelectO
  - javax.faces.component.html.Htm1SelectO
- javax.faces.component.UIPanel
  - javax.faces.component.html.Htm1PanelGrid
  - javax.faces.component.html.Htm1PanelGroup
- javax.faces.component.UIParameter
- javax.faces.component.UISelectItem
- javax.faces.component.UISelectItems
- javax.faces.component.UIViewRoot
- javax.faces.webapp.UIComponentTagBase (implements javax.servlet.jsp.tagext.JspTag)
  - javax.faces.webapp.UIComponentELTag (implements javax.servlet.jsp.tagext.Tag)
  - javax.faces.webapp.UIComponentTag (implements javax.servlet.jsp.tagext.Tag)
  - javax.faces.webapp.UIComponentBodyTag
- javax.xml.bind.Unmarshaller.Listener
- javax.activation.URLDataSource (implements javax.activation.DataSource)
- javax.mail.URLName
- javax.xml.bind.util.ValidationEventCollector (implements javax.xml.bind.ValidationEventHandler)
- javax.xml.bind.helpers.ValidationEventImpl (implements javax.xml.bind.NotIdentifiableEvent)
- javax.xml.bind.helpers.**ParseConversionEventImpl** (implements javax.xml.bind.**ParseConversionEvent**)
- javax.xml.bind.helpers.**PrintConversionEventImpl** (implements javax.xml.bind.**PrintConversionEvent**)
- javax.xml.bind.helpers.**ValidationEventLocatorImpl** (implements javax.xml.bind.**ValidationEventLocator**)
- javax.servlet.jsp.tagext.**ValidationMessage**
- javax.faces.el.**ValueBinding**
- javax.servlet.jsp.tagext.**VariableInfo**
- javax.el.**VariableMapper**
- javax.faces.el.**VariableResolver**
- javax.faces.application.**ViewHandler**
  - javax.faces.application.**ViewHandlerWrapper**
- javax.xml.bind.annotation.**W3CDomHandler** (implements javax.xml.bind.annotation.**DomHandler**<ElementT,ResultT>)
- javax.resource.spi.work.**WorkAdapter** (implements javax.resource.spi.work.**WorkListener**)
- java.io.**Writer** (implements java.lang.**Appendable**, java.io.**Closeable**, java.io.**Flushable**)
  - javax.servlet.jsp.**JspWriter**
    - javax.servlet.jsp.tagext.**BodyContent**
  - javax.faces.context.**ResponseWriter**
    - javax.faces.context.**ResponseWriterWrapper**
- javax.xml.bind.annotation.adapters.**XmlAdapter**<ValueType,BoundType>
  - javax.xml.bind.annotation.adapters.**CollapsedStringAdapter**
  - javax.xml.bind.annotation.adapters.**HexBinaryAdapter**
  - javax.xml.bind.annotation.adapters.**NormalizedStringAdapter**
- javax.xml.bind.annotation.**XmlElement.DEFAULT**
- javax.xml.bind.annotation.**XmlElementDecl.GLOBAL**
- javax.xml.bind.annotation.**XmlElementRef.DEFAULT**
- javax.xml.stream.**XMLEventFactory**
- javax.xml.stream.**XMLInputFactory**
- javax.xml.bind.annotation.adapters.**XmlJavaTypeAdapter.DEFA**
- javax.xml.stream.**XMLOutputFactory**
- javax.xml.bind.annotation.**XmlSchemaType.DEFAULT**
- javax.xml.rpc.encoding.**XMLType**
- javax.xml.bind.annotation.**XmlType.DEFAULT**
- javax.enterprise.deploy.model.**XpathEvent**
Interface Hierarchy

- javax.faces.component.**ActionSource**
  - javax.faces.component.**ActionSource2**
- javax.xml.ws.**AsyncHandler**<T>
- javax.xml.ws.**Binding**
  - javax.xml.ws.http.**HTTPBinding**
  - javax.xml.ws.soap.**SOAPBinding**
- javax.xml.ws.**BindingProvider**
  - javax.xml.ws.**Dispatch**<T>
- javax.resource.spi.**BootstrapContext**
- javax.xml.rpc.**Call**
- javax.xml.registry.**CapabilityProfile**
- javax.persistence.spi.**ClassTransformer**
- java.lang.**Cloneable**
  - javax.resource.cci.**Record** (also extends java.io.**Serializable**)  
    - javax.resource.cci.**IndexedRecord** (also extends java.util.**List**<E>, java.io.**Serializable**)  
    - javax.resource.cci.**MappedRecord** (also extends java.util.**Map**<K,V>, java.io.**Serializable**)  
    - javax.resource.cci.**ResultSet** (also extends java.sql.**ResultSet**)  
- javax.activation.**CommandObject**
- javax.resource.cci.**Connection**
- javax.jms.**Connection**
  - javax.jms.**QueueConnection**
    - javax.jms.**XAQueueConnection** (also extends javax.jms.**XAConnection**)  
  - javax.jms.**TopicConnection**
    - javax.jms.**XATopicConnection** (also extends javax.jms.**XAConnection**)  
- javax.jms.**XAConnection**
  - javax.jms.**XAQueueConnection** (also extends javax.jms.**QueueConnection**)  
  - javax.jms.**XATopicConnection** (also extends javax.jms.**TopicConnection**)  
- javax.jms.**XAQueueConnection** (also extends
javax.jms.QueueConnection, javax.jms.XAConnection
  ○ javax.jms.XATopicConnection (also extends javax.jms.TopicConnection, javax.jms.XAConnection)
  ○ javax.xml.registry.Connection
    ○ javax.xml.registry.FederatedConnection
  ○ javax.jms.ConnectionConsumer
  ○ javax.jms.ConnectionFactory
    ○ javax.jms.QueueConnectionFactory
      ○ javax.jms.XAQueueConnectionFactory (also extends javax.jms.XAConnectionFactory)
    ○ javax.jms.TopicConnectionFactory
      ○ javax.jms.XATopicConnectionFactory (also extends javax.jms.XAConnectionFactory)
    ○ javax.jms.XAQueueConnectionFactory (also extends javax.jms.QueueConnectionFactory, javax.jms.XAConnectionFactory)
    ○ javax.jms.XATopicConnectionFactory (also extends javax.jms.TopicConnectionFactory, javax.jms.XAConnectionFactory)
  ○ javax.resource.cci.ConnectionMetaData
  ○ javax.jms.ConnectionMetaData
  ○ javax.resource.spi.ConnectionRequestInfo
  ○ javax.resource.cci.ConnectionSpec
  ○ org.xml.sax.ContentHandler
    ○ javax.xml.bind.UnmarshallerHandler
  ○ javax.faces.component.ContextCallback
  ○ javax.faces.convert.Converter
  ○ javax.activation.DataContentHandler
  ○ javax.activation.DataContentHandlerFactory
  ○ javax.activation.DataSource
    ○ javax.mail.MultipartDataSource
  ○ javax.xml.bind.DatatypeConverterInterface
  ○ javax.enterprise.deploy.spi.DConfigBean
    ○ javax.enterprise.deploy.spi.DConfigBeanRoot
  ○ javax.enterprise.deploy.model.DDBean
    ○ javax.enterprise.deploy.model.DDBeanRoot
  ○ javax.jms.DeliveryMode
  ○ javax.enterprise.deploy.model.DeployableObject
    ○ javax.enterprise.deploy.model.J2eeApplicationObject
- javax.enterprise.deploy.spi.DeploymentConfiguration
- javax.enterprise.deploy.spi.factories.DeploymentFactory
- javax.enterprise.deploy.spi.DeploymentManager
- javax.enterprise.deploy.spi.status.DeploymentStatus
- javax.xml.rpc.encoding.DeserializationContext
- javax.jms.Destination
  - javax.jms.Queue
    - javax.jms.TemporaryQueue
  - javax.jms.TemporaryQueue
  - javax.jms.TemporaryTopic
  - javax.jms.Topic
    - javax.jms.TemporaryTopic
- javax.resource.spi.DissociatableManagedConnection
- javax.xml.bind.annotation.DomHandler<ElementT,ResultT>
- javax.servlet.jsp.tagext.DynamicAttributes
- javax.ejb.EJBContext
  - javax.ejb.EntityContext
  - javax.ejb.MessageDrivenContext
  - javax.ejb.SessionContext
- javax.ejb.EJBLocalHome
- javax.ejb.EJBLocalObject
- javax.ejb.EJBMetaData
- javax.xml.bind.Element
- javax.xml.registry.infomodel.EmailAddress
- javax.persistence.EntityManager
- javax.persistence.EntityManagerFactory
- javax.persistence.EntityTransaction
- javax.xml.stream.EventFilter
- java.util.EventListener
  - javax.resource.spi.ConnectionEventListener
- javax.mail.event.ConnectionListener
- javax.faces.model.DataModelListener
- javax.el.ELContextListener
- javax.faces.event.FacesListener
  - javax.faces.event.ActionListener
  - javax.faces.event.ValueChangeListener
- javax.mail.event.FolderListener
- javax.servlet.http.HttpSessionActivationListener
- javax.servlet.http.HttpSessionAttributeListener
- javax.servlet.http.HttpSessionBindingListener
- javax.servlet.http.HttpSessionListener
- javax.mail.event.MessageChangedListener
- javax.mail.event.MessageCountListener
- javax.faces.event.PhaseListener
  (also extends java.io.Serializable)
- javax.enterprise.deploy.spi.status.ProgressListener
- javax.servlet.ServletContextAttributeListener
- javax.servlet.ServletContextListener
- javax.servletServletRequestAttributeListener
- javax.servlet.ServletRequestListener
- javax.mail.event.StoreListener
- javax.mail.event.TransportListener
- javax.faces.validator.Validator
- javax.resource.spi.work.WorkListener
- javax.jms.ExceptionListener
- javax.xml.registry.infomodel.ExtensibleObject
  - javax.xml.registry.infomodel.Association
  - javax.xml.registry.infomodel.AuditableEvent
  - javax.xml.registry.infomodel.Classification
  - javax.xml.registry.infomodel.ClassificationScheme
  - javax.xml.registry.infomodel.Concept
  - javax.xml.registry.infomodel.ExternalIdentifier
  - javax.xml.registry.infomodel.ExternalLink
    (also extends javax.xml.registry.infomodel.RegistryObject,
     javax.xml.registry.infomodel.URIValidator)
  - javax.xml.registry.infomodel.ExtrinsicObject
  - javax.xml.registry.infomodel.Organization
  - javax.xml.registry.infomodel.PostalAddress
  - javax.xml.registry.infomodel.RegistryEntry
    (also extends javax.xml.registry.infomodel.RegistryObject,
     javax.xml.registry.infomodel.Versionable)
    - javax.xml.registry.infomodel.ClassificationScheme
    - javax.xml.registry.infomodel.ExtrinsicObject
    - javax.xml.registry.infomodel.RegistryPackage
    - javax.xml.registry.infomodel.Service
  - javax.xml.registry.infomodel.RegistryObject
    - javax.xml.registry.infomodel.Association
    - javax.xml.registry.infomodel.AuditableEvent
- javax.xml.registry.infomodel.*Classification*
- javax.xml.registry.infomodel.*ClassificationScheme*
- javax.xml.registry.infomodel.*Concept*
- javax.xml.registry.infomodel.*ExternalIdentifier*
- javax.xml.registry.infomodel.*ExternalLink* (also extends javax.xml.registry.infomodel.*URIValidator*
- javax.xml.registry.infomodel.*ExtrinsicObject*
- javax.xml.registry.infomodel.*Organization*
- javax.xml.registry.infomodel.*RegistryEntry* (also extends javax.xml.registry.infomodel.*Versionable*)
  - javax.xml.registry.infomodel.*ClassificationScheme*
  - javax.xml.registry.infomodel.*ExtrinsicObject*
  - javax.xml.registry.infomodel.*RegistryPackage*
  - javax.xml.registry.infomodel.*Service*
- javax.xml.registry.infomodel.*RegistryPackage*
- javax.xml.registry.infomodel.*Service*
- javax.xml.registry.infomodel.*ServiceBinding* (also extends javax.xml.registry.infomodel.*URIValidator*)
  - javax.xml.registry.infomodel.*SpecificationLink*
  - javax.xml.registry.infomodel.*User*
- javax.xml.registry.infomodel.*RegistryPackage*
- javax.xml.registry.infomodel.*Service*
- javax.xml.registry.infomodel.*ServiceBinding* (also extends javax.xml.registry.infomodel.*RegistryObject*, javax.xml.registry.infomodel.*URIValidator*)
  - javax.xml.registry.infomodel.*SpecificationLink*
  - javax.xml.registry.infomodel.*User*
- javax.servlet.*Filter*
- javax.servlet.*FilterChain*
- javax.servlet.*FilterConfig*
- javax.xml.registry.*FindQualifier*
- javax.servlet.jsp.el.*FunctionMapper*
- java.util.concurrent.*Future*<V>
  - javax.xml.ws.*Response*<T>
- javax.resource.spi.security.*GenericCredential*
- javax.ejb.spi.*HandleDelegate*
- javax.xml.rpc.handler.*Handler*
- javax.xml.ws.handler.*Handler*C
  - javax.xml.ws.handler.*LogicalHandler*C
- `javax.xml.ws.handler.soap.SOAPHandler<T>`
- `javax.xml.ws.handler.HandlerResolver`
- `javax.xml.rpc.holders.Holder`
- `javax.servlet.http.HttpSession`
- `javax.servlet.http.HttpSessionContext`
- `javax.resource.cci.Interaction`
- `javax.xml.registry.infomodel.InternationalString`
- `javax.interceptor.InvocationContext`
- `java.lang.Iterable<T>`
  - `java.util.Collection<E>`
    - `java.util.List<E>`
      - `javax.xml.rpc.handler.HandlerChain`
      - `javax.resource.cci.IndexedRecord` (also extends `javax.resource.cci.Record`, `java.io.Serializable`)
  - `java.util.Iterator<E>`
    - `javax.xml.stream.XMLEventReader`
- `javax.management.j2ee.statistics.JavaMailStats`
- `javax.xml.registry.JAXRResponse`
  - `javax.xml.registry.BulkResponse`
- `javax.servlet.jsp.JspApplicationContext`
- `javax.servlet.jsp.tagext.JspIdConsumer`
- `javax.servlet.jsp.tagext.JspTag`
  - `javax.servlet.jsp.tagext.BodyTag`
  - `javax.servlet.jsp.tagext.IterationTag`
    - `javax.servlet.jsp.tagext.BodyTag`
  - `javax.servlet.jsp.tagext.SimpleTag`
  - `javax.servlet.jsp.tagext.Tag`
    - `javax.servlet.jsp.tagext.BodyTag`
    - `javax.servlet.jsp.tagext IterationTag`
      - `javax.servlet.jsp.tagext.BodyTag`
- `javax.xml.registry.infomodel.Key`
- `javax.resource.spi.LazyAssociatableConnectionManager`
- `javax.resource.spi.LazyEnlistableConnectionManager`
- `javax.resource.spi.LazyEnlistableManagedConnection`
- `javax.xml.registry.LifecycleManager`
  - `javax.xml.registry.BusinessLifecycleManager`
- `javax.xml.registry.infomodel.LocalizationString`
- `javax.resource.cci.LocalTransaction`
- `javax.resource.spi.LocalTransaction`
javax.xml.stream.**Location**
javax.xml.ws.**LogicalMessage**
javax.resource.spi.**ManagedConnection**
javax.resource.spi.**ManagedConnectionMetaData**
java.util.**Map**<K,V>
  - javax.xml.ws.cci.**MappedRecord** (also extends javax.xml.ws.cci.**Record**, java.io.**Serializable**)
  - javax.xml.ws.handler.**MessageContext**
    - javax.xml.ws.handler.**LogicalMessageContext**
    - javax.xml.ws.handler.soap.**SOAPMessageContext**
javax.xml.bind.**Marshaller**
javax.jms.**Message**
  - javax.jms.**BytesMessage**
  - javax.jms.**MapMessage**
  - javax.jms.**ObjectMessage**
  - javax.jms.**StreamMessage**
  - javax.jms.**TextMessage**
javax.mail.**MessageAware**
javax.jms.**MessageConsumer**
  - javax.jms.**QueueReceiver**
  - javax.jms.**TopicSubscriber**
javax.xml.rpc.handler.**MessageContext**
  - javax.xml.rpc.handler.soap.**SOAPMessageContext**
javax.resource.spi.endpoint.**MessageEndpoint**
javax.resource.spi.endpoint.**MessageEndpointFactory**
javax.resource.cci.**MessageListener**
javax.jms.**MessageListener**
javax.jms.**MessageProducer**
  - javax.jms.**QueueSender**
  - javax.jms.**TopicPublisher**
javax.xml.soap.**Name**
javax.faces.component.**NamingContainer**
org.w3c.dom.**Node**
  - org.w3c.dom.**CharacterData**
    - org.w3c.dom.**Text** (also extends javax.xml.soap.**Text**) (also extends javax.xml.soap.**Node**)
  - org.w3c.dom.**Element**
    - javax.xml.soap.**SOAPElement** (also extends
javax.xml.soap.Node
- javax.xml.soap.Detail
- javax.xml.soap.DetailEntry
- javax.xml.soap.SOAPBody
- javax.xml.soap.SOAPBodyElement
  - javax.xml.soap.SOAPFault
- javax.xml.soap.SOAPEnvelope
- javax.xml.soap.SOAPFault
- javax.xml.soap.SOAPFaultElement
  - javax.xml.soap.Detail
- javax.xml.soap.SOAPHeader
- javax.xml.soap.SOAPHeaderElement
- javax.xml.soap.Node
  - javax.xml.soap.Detail
  - javax.xml.soap.DetailEntry
  - javax.xml.soap.SOAPBody
  - javax.xml.soap.SOAPBodyElement
    - javax.xml.soap.SOAPFault
  - javax.xml.soap.SOAPEnvelope
  - javax.xml.soap.SOAPFault
  - javax.xml.soap.SOAPFaultElement
    - javax.xml.soap.Detail
  - javax.xml.soap.SOAPHeader
  - javax.xml.soap.SOAPHeaderElement
  - javax.xml.soap.SOAPEnvelope
  - javax.xml.soap.SOAPFault
  - javax.xml.soap.SOAPFaultElement
    - javax.xml.soap.Detail
  - javax.xml.soap.SOAPHeader
  - javax.xml.soap.SOAPHeaderElement
- javax.xml.soap.SOAPEnvelope
- javax.xml.soap.SOAPFault
- javax.xml.soap.SOAPFaultElement
  - javax.xml.soap.Detail
- javax.xml.soap.SOAPHeader
- javax.xml.soap.SOAPHeaderElement
- javax.mail.Part
- javax.mail.internet.MimePart
- javax.persistence.spi.PersistenceProvider
- javax.persistence.spi.PersistenceUnitInfo
- javax.xml.registry.infomodel.PersonName
- javax.security.jacc.PolicyConfiguration
- javax.security.jacc.PolicyContextHandler
- javax.xml.ws.handler.PortInfo
- javax.enterprise.deploy.spi.status.ProgressObject
- javax.xml.ws.Provider<T>
- javax.xml.registry.Query
- javax.persistence.Query
- javax.xml.registry.QueryManager
  - javax.xml.registry.BusinessQueryManager
  - javax.xml.registry.DeclarativeQueryManager
- javax.jms.QueueBrowser
- javax.mail.QuotaAwareStore
- javax.resource.cci.RecordFactory
- javax.naming.Referenceable
  - javax.resource.cci.ConnectionFactory (also extends java.io.Serializable)
- javax.xml.registry.RegistryService
- java.rmi.Remote
  - javax.ejb.EJBHome
    - javax.management.j2ee.ManagementHome
  - javax.ejb.EJBObject
    - javax.management.j2ee.Management
- javax.servlet.RequestDispatcher
- javax.resource.spi.ResourceAdapter
- javax.resource.spi.ResourceAdapterAssociation
  - javax.resource.spi.ActivationSpec
- javax.resource.cci.ResourceAdapterMetaData
- java.sql.ResultSet
  - javax.resource.cci.ResultSet (also extends javax.resource.cci.Record)
- javax.resource.cci.ResultSetInfo
- java.langRunnable
- javax.jms.Session
  - javax.jms.QueueSession
- javax.jms.**TopicSession**
- javax.jms.**XAQueueSession**
- javax.jms.**XASession**
  - javax.jms.**XAQueueSession**
  - javax.jms.**XATopicSession**
- javax.jms.**XATopicSession**
- javax.resource.spi.work.**Work**
- java.io.**Serializable**
- javax.enterprise.deploy.spi.status.**ClientConfiguration**
- javax.resource.cci.**ConnectionFactory** (also extends javax.resource.**Referenceable**)
- javax.resource.spi.**ConnectionManager**
- javax.xml.rpc.encoding.**Deserializer**
- javax.xml.rpc.encoding.**DeserializerFactory**
- javax.ejb.**EnterpriseBean**
  - javax.ejb.**EntityBean**
  - javax.ejb.**MessageDrivenBean**
  - javax.ejb.**SessionBean**
- javax.ejb.**Handle**
- javax.xml.rpc.handler.**HandlerRegistry**
- javax.ejb.**HomeHandle**
- javax.resource.cci.**IndexedRecord** (also extends java.util.**List<E>**, javax.resource.cci.**Record**)
- javax.resource.cci.**InteractionSpec**
- javax.management.j2ee.**ListenerRegistration**
- javax.resource.spi.**ManagedConnectionFactory**
- javax.resource.cci.**MappedRecord** (also extends java.util.**Map<K,V>**, javax.resource.cci.**Record**)
- javax.faces.event.**PhaseListener** (also extends java.util.**EventListener**)
- javax.resource.cci.**Record** (also extends java.lang.**Cloneable**)
  - javax.resource.cci.**IndexedRecord** (also extends java.util.**List<E>**, java.io.**Serializable**)
  - javax.resource.cci.**MappedRecord** (also extends java.util.**Map<K,V>**, java.io.**Serializable**)
  - javax.resource.cci.**ResultSet** (also extends java.sql.**ResultSet**)
- javax.xml.rpc.encoding.**Serializer**
- javax.xml.rpc.encoding.**SerializerFactory**
- javax.ejb. TimerHandle
- javax.xml.rpc.encoding. TypeMappingRegistry
- javax.xml.rpc.encoding. SerializationContext
- javax.jms. ServerSession
- javax.jms. ServerSessionPool
- javax.xml.rpc. Service
- javax.xml.rpc.server. ServiceLifecycle
- javax.servlet. Servlet
  - javax.servlet.jsp. HttpJspPage
  - javax.servlet.jsp. JspPage
    - javax.servlet.jsp. HttpJspPage
- javax.servlet. ServletConfig
- javax.servlet. ServletContext
- javax.xml.rpc.server. ServletEndpointContext
- javax.servlet. ServletRequest
  - javax.servlet.http. HttpServletRequest
  - javax.servlet.http. HttpServletResponse
- javax.ejb. SessionSynchronization
- javax.mail.internet. SharedInputStream
- javax.servlet. SingleThreadModel
- javax.xml.registry.infomodel. Slot
- javax.xml.soap. SOAPConstants
- javax.faces.component. StateHolder
- javax.management.j2ee.statistics. Statistic
  - javax.management.j2ee.statistics. BoundaryStatistic
    - javax.management.j2ee.statistics. BoundedRangeStatistic
      (also extends
       javax.management.j2ee.statistics. RangeStatistic)
  - javax.management.j2ee.statistics. BoundedRangeStatistic
    (also extends
     javax.management.j2ee.statistics. BoundaryStatistic, 
     javax.management.j2ee.statistics. RangeStatistic)
  - javax.management.j2ee.statistics. CountStatistic
  - javax.management.j2ee.statistics. RangeStatistic
    - javax.management.j2ee.statistics. BoundedRangeStatistic
      (also extends
       javax.management.j2ee.statistics. BoundaryStatistic)
  - javax.management.j2ee.statistics. TimeStatistic
- javax.management.j2ee.statistics.Stats
  - javax.management.j2ee.statistics.EJBStats
    - javax.management.j2ee.statistics.EntityBeanStats
    - javax.management.j2ee.statistics.MessageDrivenBeanStats
    - javax.management.j2ee.statistics.SessionBeanStats
      - javax.management.j2ee.statistics.StatefulSessionBeanStats
      - javax.management.j2ee.statistics.StatelessSessionBeanStats
    - javax.management.j2ee.statistics.EntityBeanStats
  - javax.management.j2ee.statistics.JCAConnectionPoolingStats
    - javax.management.j2ee.statistics.JCAClassStats
    - javax.management.j2ee.statistics.JDBCConnectionPoolingStats
      - javax.management.j2ee.statistics.JDBCClassStats
    - javax.management.j2ee.statistics.JMSSharedConnectionStats
    - javax.management.j2ee.statistics.JMSClassStats
    - javax.management.j2ee.statistics.JTAMethodStats
    - javax.management.j2ee.statistics.JVMStats
  - javax.management.j2ee.statistics.MessageDrivenBeanStats
    - javax.management.j2ee.statistics.StatefulSessionBeanStats
    - javax.management.j2ee.statistics.StatelessSessionBeanStats
  - javax.management.j2ee.statistics.ServletStats
  - javax.management.j2ee.statistics.SessionBeanStats
    - javax.management.j2ee.statistics.StatefulSessionBeanStats
    - javax.management.j2ee.statistics.StatelessSessionBeanStats
  - javax.management.j2ee.statistics.URLStats
- javax.transaction.Status
- javax.resource.cci.Streamable
- javax.xml.stream.StreamFilter
- javax.xml.rpc. **Stub**
- javax.transaction. **Synchronization**
- javax.enterprise.deploy.spi. **Target**
- javax.enterprise.deploy.spi. **TargetModuleID**
- javax.xml.registry.infomodel. **TelephoneNumber**
- javax.ejb. **TimedObject**
- javax.ejb. **Timer**
- javax.ejb. **TimerService**
- javax.transaction. **Transaction**
- javax.transaction. **TransactionManager**
- javax.transaction. **TransactionSynchronizationRegistry**
- javax.servlet.jsp.tagext. **TryCatchFinally**
- javax.xml.rpc.encoding. **TypeMapping**
- javax.mail. **UIDFolder**
- javax.xml.bind. **Unmarshaller**
- javax.xml.registry.infomodel. **URIVariable**
  - javax.xml.registry.infomodel. **ExternalLink** (also extends javax.xml.registry.infomodel. **RegistryObject**)
  - javax.xml.registry.infomodel. **ServiceBinding** (also extends javax.xml.registry.infomodel. **RegistryObject**)
- javax.transaction. **UserTransaction**
- javax.resource.spi. **ValidatingManagedConnectionFactory**
- javax.xml.bind. **ValidationEvent**
  - javax.xml.bind. **NotIdentifiableEvent**
  - javax.xml.bind. **ParseConversionEvent**
  - javax.xml.bind. **PrintConversionEvent**
- javax.xml.bind. **ValidationEventHandler**
- javax.xml.bind. **ValidationEventLocator**
- javax.faces.component. **ValueHolder**
  - javax.faces.component. **EditableValueHolder**
- javax.servlet.jsp.el. **VariableResolver**
- javax.xml.registry.infomodel. **Versionable**
  - javax.xml.registry.infomodel. **ClassificationScheme**
  - javax.xml.registry.infomodel. **ExtrinsicObject**
  - javax.xml.registry.infomodel. **RegistryEntry** (also extends javax.xml.registry.infomodel. **RegistryObject**)
    - javax.xml.registry.infomodel. **ClassificationScheme**
    - javax.xml.registry.infomodel. **ExtrinsicObject**
- javax.xml.registry.infomodel.RegistryPackage
- javax.xml.registry.infomodel.Service
- javax.xml.registry.infomodel.RegistryPackage
- javax.xml.registry.infomodel.Service
- javax.xml.ws.WebServiceContext
- javax.resource.spi.work.WorkManager
- javax.jms.XAConnectionFactory
  - javax.jmsXAQueueConnectionFactory (also extends javax.jms.QueueConnectionFactory)
  - javax.jms.XATopicConnectionFactory (also extends javax.jms.TopicConnectionFactory)
- javax.transaction.xa.XAResource
- javax.resource.spi.XATerminator
- javax.transaction.xa.Xid
- javax.xml.stream.util.XMLEventAllocator
- javax.xml.stream.util.XMLEventConsumer
  - javax.xml.stream.XMLEventWriter
- javax.xml.stream.XMLReporter
- javax.xml.stream.XMLResolver
- javax.xml.stream.XMLStreamConstants
  - javax.xml.stream.events.Attribute
    - javax.xml.stream.events.Namespace
  - javax.xml.stream.events.Characters
  - javax.xml.stream.events.Comment
  - javax.xml.stream.events.DTD
  - javax.xml.stream.events.EndDocument
  - javax.xml.stream.events.EndElement
  - javax.xml.stream.events.EntityDeclaration
  - javax.xml.stream.events.EntityReference
  - javax.xml.stream.events.NotationDeclaration
  - javax.xml.stream.events.ProcessingInstruction
  - javax.xml.stream.events.StartDocument
  - javax.xml.stream.events.StartElement
  - javax.xml.stream.events.XMLEvent
    - javax.xml.stream.events.Attribute
      - javax.xml.stream.events.Namespace
    - javax.xml.stream.events.Characters
    - javax.xml.stream.events.Comment
- `javax.xml.stream.events/DTD`
- `javax.xml.stream.events.EndDocument`
- `javax.xml.stream.events.EndElement`
- `javax.xml.stream.events.EntityDeclaration`
- `javax.xml.stream.events.EntityReference`
- `javax.xml.stream.events.Namespace`
- `javax.xml.stream.events.NotationDeclaration`
- `javax.xml.stream.events.ProcessingInstruction`
- `javax.xml.stream.events.StartDocument`
- `javax.xml.stream.eventsStartElement`

  - `javax.xml.stream.XMLStreamReader`
  - `javax.xml.stream.XMLStreamWriter`
  - `javax.enterprise.deploy.model.XPathListener`
Annotation Type Hierarchy

- javax.ejb.**ActivationConfigProperty** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**ApplicationException** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**EJB** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**EJBs** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Init** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Local** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**LocalHome** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**MessageDriven** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**PostActivate** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**PrePassivate** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Remote** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**RemoteHome** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Remove** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Stateful** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Stateless** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**Timeout** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**TransactionAttribute** (implements java.lang.annotation.**Annotation**)
- javax.ejb.**TransactionManagement** (implements java.lang.annotation.**Annotation**)
- javax.interceptor.**AroundInvoke** (implements java.lang.annotation.**Annotation**)
- javax.interceptor.**ExcludeClassInterceptors** (implements java.lang.annotation.**Annotation**)
- javax.interceptor.**ExcludeDefaultInterceptors** (implements java.lang.annotation.**Annotation**)
- javax.interceptor.**Interceptors** (implements java.lang.annotation.**Annotation**)
- javax.annotation.**Generated** (implements java.lang.annotation.**Annotation**
java.lang.annotation.Annotation

- javax.annotation.PostConstruct (implements java.lang.annotation.Annotation)
- javax.annotation.PreDestroy (implements java.lang.annotation.Annotation)
- javax.annotation.Resource (implements java.lang.annotation.Annotation)
- javax.annotation.Resources (implements java.lang.annotation.Annotation)
- javax.annotation.security.DeclareRoles (implements java.lang.annotation.Annotation)
- javax.annotation.security.DenyAll (implements java.lang.annotation.Annotation)
- javax.annotation.security.PermitAll (implements java.lang.annotation.Annotation)
- javax.annotation.security.RolesAllowed (implements java.lang.annotation.Annotation)
- javax.annotation.security.RunAs (implements java.lang.annotation.Annotation)
- javax.xml.ws.BindingType (implements java.lang.annotation.Annotation)
- javax.xml.ws.RequestWrapper (implements java.lang.annotation.Annotation)
- javax.xml.ws.ResponseWrapper (implements java.lang.annotation.Annotation)
- javax.xml.ws.ServiceMode (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebEndpoint (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebFault (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebServiceClient (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebServiceProvider (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebServiceRef (implements java.lang.annotation.Annotation)
- javax.xml.ws.WebServiceRefs (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlAccessorType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlAnyAttribute (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlAnyElement (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlAttachmentRef (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlAttribute (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElement (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElementDecl (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElementRef (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElementRefs (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElementWrapper (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlElements (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlEnum (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlEnumValue (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlID (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlIDREF (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlInlineBinaryData (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlMimeType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlMixed (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlName (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlNamespaceAware (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlNamespaceRef (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlNamespaces (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaValue (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlValue (implements java.lang.annotation.Annotation)
java.lang.annotation.Annotation
- javax.xml.bind.annotation.XmlNs (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlRegistry (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlRootElement (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchema (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlSchemaTypes (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlTransient (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlType (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.XmlValue (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.adapters.XmlJavaTypeAdapter (implements java.lang.annotation.Annotation)
- javax.xml.bind.annotation.adapters.XmlJavaTypeAdapters (implements java.lang.annotation.Annotation)
- javax.jws.HandlerChain (implements java.lang.annotation.Annotation)
- javax.jws.Oneway (implements java.lang.annotation.Annotation)
- javax.jws.WebMethod (implements java.lang.annotation.Annotation)
- javax.jws.WebParam (implements java.lang.annotation.Annotation)
- javax.jws.WebResult (implements java.lang.annotation.Annotation)
- javax.jws.WebService (implements java.lang.annotation.Annotation)
- javax.jws.soap.InitParam (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPBinding (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPMessageHandler (implements java.lang.annotation.Annotation)
- javax.jws.soap.SOAPMessageHandlers (implements java.lang.annotation.Annotation)
- javax.persistence.AssociationOverride (implements java.lang.annotation.Annotation)
java.lang.annotation.Annotation
- javax.persistence.AssociationOverrides (implements java.lang.annotation.Annotation)
- javax.persistence.AttributeOverride (implements java.lang.annotation.Annotation)
- javax.persistence.AttributeOverrides (implements java.lang.annotation.Annotation)
- javax.persistence.Basic (implements java.lang.annotation.Annotation)
- javax.persistence.Column (implements java.lang.annotation.Annotation)
- javax.persistence.ColumnResult (implements java.lang.annotation.Annotation)
- javax.persistence.DiscriminatorColumn (implements java.lang.annotation.Annotation)
- javax.persistence.DiscriminatorValue (implements java.lang.annotation.Annotation)
- javax.persistence.Embeddable (implements java.lang.annotation.Annotation)
- javax.persistence.Embedded (implements java.lang.annotation.Annotation)
- javax.persistence.EmbeddedId (implements java.lang.annotation.Annotation)
- javax.persistence.Entity (implements java.lang.annotation.Annotation)
- javax.persistence.EntityListeners (implements java.lang.annotation.Annotation)
- javax.persistence.EntityResult (implements java.lang.annotation.Annotation)
- javax.persistence.Enumerated (implements java.lang.annotation.Annotation)
- javax.persistence.ExcludeDefaultListeners (implements java.lang.annotation.Annotation)
- javax.persistence.ExcludeSuperclassListeners (implements java.lang.annotation.Annotation)
- javax.persistence.FieldResult (implements java.lang.annotation.Annotation)
- javax.persistence.GeneratedValue (implements java.lang.annotation.Annotation)
- `javax.persistence.Id` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.IdClass` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Inheritance` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.JoinColumn` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.JoinColumns` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.JoinTable` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Lob` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.ManyToMany` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.ManyToOne` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.MapKey` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.MappedSuperclass` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.NamedNativeQueries` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.NamedNativeQuery` (implement `java.lang.annotation.Annotation`)
- `javax.persistence.NamedQueries` (implements `java.lang.annotation.Annotation`)
- `javax.persistenceNamedQuery` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.OneToMany` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.OneToOne` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.OrderBy` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceContext` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceContexts` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.PersistenceProperty` (implements...
java.lang.annotation.Annotation
- javax.persistence.PersistenceUnit (implements java.lang.annotation.Annotation)
- javax.persistence.PersistenceUnits (implements java.lang.annotation.Annotation)
- javax.persistence.PostLoad (implements java.lang.annotation.Annotation)
- javax.persistence.PostPersist (implements java.lang.annotation.Annotation)
- javax.persistence.PostRemove (implements java.lang.annotation.Annotation)
- javax.persistence.PostUpdate (implements java.lang.annotation.Annotation)
- javax.persistence.PrePersist (implements java.lang.annotation.Annotation)
- javax.persistence.PreRemove (implements java.lang.annotation.Annotation)
- javax.persistence.PreUpdate (implements java.lang.annotation.Annotation)
- javax.persistence.PrimaryKeyJoinColumn (implements java.lang.annotation.Annotation)
- javax.persistence.PrimaryKeyJoinColumns (implements java.lang.annotation.Annotation)
- javax.persistence.QueryHint (implements java.lang.annotation.Annotation)
- javax.persistence.SecondaryTable (implements java.lang.annotation.Annotation)
- javax.persistence.SecondaryTables (implements java.lang.annotation.Annotation)
- javax.persistence.SequenceGenerator (implements java.lang.annotation.Annotation)
- javax.persistence.SqlResultSetMapping (implements java.lang.annotation.Annotation)
- javax.persistence.SqlResultSetMappings (implements java.lang.annotation.Annotation)
- javax.persistence.Table (implements java.lang.annotation.Annotation)
- javax.persistence.TableGenerator (implements java.lang.annotation.Annotation)
- `javax.persistence.Temporal` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Transient` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.UniqueConstraint` (implements `java.lang.annotation.Annotation`)
- `javax.persistence.Version` (implements `java.lang.annotation.Annotation`
Enum Hierarchy

- java.lang.**Object**
  - java.lang.**Enum**<E> (implements java.lang.**Comparable**<T>, java.io.**Serializable**)
    - javax.ejb.**TransactionAttributeType**
    - javax.ejb.**TransactionManagementType**
    - javax.annotation.**Resource.AuthenticationType**
    - javax.xml.ws.**Service.Mode**
    - javax.xml.ws.handler.**MessageContext.Scope**
    - javax.xml.bind.annotation.**XmlAccessOrder**
    - javax.xml.bind.annotation.**XmlAccessType**
    - javax.xml.bind.annotation.**XmlNsForm**
    - javax.jws.**WebParam.Mode**
    - javax.jws.soap.**SOAPBinding.Style**
    - javax.jws.soap.**SOAPBinding.Use**
    - javax.jws.soap.**SOAPBinding.ParameterStyle**
    - javax.persistence.**CascadeType**
    - javax.persistence.**DiscriminatorType**
    - javax.persistence.**EnumType**
    - javax.persistence.**FetchType**
    - javax.persistence.**FlushModeType**
    - javax.persistence.**GenerationType**
    - javax.persistence.**InheritanceType**
    - javax.persistence.**LockModeType**
    - javax.persistence.**PersistenceContextType**
    - javax.persistence.**TemporalType**
    - javax.persistence.spi.**PersistenceUnitTransactionType**

---

**Overview**  Package  Class  **Deprecated**  Index  Help

**Submit a bug or feature**

Copyright 2007 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](https://www.oracle.com/licenses/index.html).