AssemblyResolver Class

Resolves assemblies located in a specified directory and its subdirectories.

For a list of all members of this type, see <u>AssemblyResolver</u> Members.

System.Object AssemblyResolver

public class AssemblyResolver

Remarks

Class AssemblyResolver resolves assemblies not found by the system. An instance of this class is configured with a base directory and hooks up to the AppDomain.AssemblyResolve event. Whenever called, the instance checks the associated directory with all subdirectories for the assembly requested.

The class implements two features to speed up the search:

- AssemblyList: Before searching the file system, the assembly list is searched. Once an assembly has been found, it is added to the AssemblyList.
- SubDirectoryCache: The class caches the subdirectories of each directory once they have been determined. This avoids repeated filesystem queries for subdirectories.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>AssemblyResolver Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver Members

AssemblyResolver overview

Public Instance Constructors

<u> </u>	Constructs an instance of this
Constructor	type.

Public Instance Properties

<u> ■IncludeSubdirs</u>	Whether or not to include sub-
	directories in the searches
	which are in response to the
	AssemblyResolve event.

Public Instance Methods

- Deinstall	Deinstalls the assembly resolver.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
◆GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>••Install</u>	Installs the assembly resolver by hooking up to the AppDomain's AssemblyResolve event.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
*ResolveAssembly	Resolves the location and loads an assembly not found by the system.

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver Constructor

Constructs an instance of this type.

```
public AssemblyResolver(
   ArrayList directories
);
```

Parameters

directories

A list of directories to search for assemblies in.

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver Properties

The properties of the **AssemblyResolver** class are listed below. For a complete list of **AssemblyResolver** class members, see the <u>AssemblyResolver Members</u> topic.

Public Instance Properties

<u> </u>	Whether or not to include sub-
	directories in the searches
	which are in response to the
	AssemblyResolve event.

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver.IncludeSubdirs Property

Whether or not to include sub-directories in the searches which are in response to the AssemblyResolve event.

```
public bool IncludeSubdirs {get; set;}
```

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver Methods

The methods of the **AssemblyResolver** class are listed below. For a complete list of **AssemblyResolver** class members, see the <u>AssemblyResolver Members</u> topic.

Public Instance Methods

<u> </u>	Deinstalls the assembly resolver.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
*•Install	Installs the assembly resolver by hooking up to the AppDomain's AssemblyResolve event.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
	Resolves the location and loads

an assembly not found by the system.

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver.Deinstall Method

Deinstalls the assembly resolver.

```
public void Deinstall();
```

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver.Install Method

Installs the assembly resolver by hooking up to the AppDomain's AssemblyResolve event.

```
public void Install();
```

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblyResolver.ResolveAssembly Method

Resolves the location and loads an assembly not found by the system.

```
protected Assembly ResolveAssembly(
   object sender,
   ResolveEventArgs args
);
```

Parameters

```
sender
the sender of the event
args
event arguments
```

Return Value

the loaded assembly, null, if not found

See Also

<u>AssemblyResolver Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc Class

Represents an assembly and /doc pair.

For a list of all members of this type, see <u>AssemblySlashDoc</u> Members.

System.Object AssemblySlashDoc

public class AssemblySlashDoc

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>AssemblySlashDoc Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc Members

AssemblySlashDoc overview

Public Instance Constructors

<u></u> ◆AssemblySlashDoc	Overloaded. Initializes a new
	instance of the
	AssemblySlashDoc class.

Public Instance Properties

<u> ■AssemblyFilename</u>	Gets or sets the assembly filename in this pair.
<u>SlashDocFilename</u>	Gets or sets the /doc filename in this pair.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
••GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
	garbage collection.

MemberwiseClone (inherited from Object)

Creates a shallow copy of the current Object.

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc Constructor

Initializes a new instance of the AssemblySlashDoc class.

Overload List

Initializes a new instance of the AssemblySlashDoc class.

public AssemblySlashDoc();

Initializes a new instance of the AssemblySlashDoc class to the values indicated.

public AssemblySlashDoc(string,string);

See Also

AssemblySlashDoc Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc Constructor ()

Initializes a new instance of the AssemblySlashDoc class.

public AssemblySlashDoc();

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u> | <u>AssemblySlashDoc Constructor Overload List</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc Constructor (String, String)

Initializes a new instance of the AssemblySlashDoc class to the values indicated.

```
public AssemblySlashDoc(
    string assemblyFilename,
    string slashDocFilename
);
```

Parameters

assemblyFilename
An assembly filename.
slashDocFilename
A /doc filename.

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u> | AssemblySlashDoc Constructor Overload List

Copyright © 2000-2003 the NDoc Team

Generated from assembly NDoc.Core [1.2.1300.31138]

AssemblySlashDoc Properties

The properties of the **AssemblySlashDoc** class are listed below. For a complete list of **AssemblySlashDoc** class members, see the <u>AssemblySlashDoc Members</u> topic.

Public Instance Properties

™ AssemblyFilename	Gets or sets the assembly filename in this pair.
<u>SlashDocFilename</u>	Gets or sets the /doc filename in this pair.

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc.AssemblyFilename Property

Gets or sets the assembly filename in this pair.

```
public string AssemblyFilename {get; set;}
```

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AssemblySlashDoc.SlashDocFilename Property

Gets or sets the /doc filename in this pair.

```
public string SlashDocFilename {get; set;}
```

See Also

<u>AssemblySlashDoc Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AttributesEditor Class

Class which implements a custom UITypeEditor for attributes.

For a list of all members of this type, see <u>AttributesEditor Members</u>.

System.Object UITypeEditor

AttributesEditor

```
public class AttributesEditor :
   UITypeEditor
```

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

AttributesEditor Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesEditor Members

AttributesEditor overview

Public Instance Constructors

AttributesEditor Constructor	Initializes a new instance of the	
	AttributesEditor class.	

Public Instance Methods

<u>=•EditValue</u>	Overloaded. Handler called when editing a value.
EditValue (inherited from UITypeEditor)	Overloaded. Edits the value of the specified object using the editor style indicated by GetEditStyle.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
SetEditStyle	Overloaded. Returns the edit style for the type.
GetEditStyle (inherited from UITypeEditor)	Overloaded. Gets the editor style used by the EditValue method.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetPaintValueSupported (inherited from UITypeEditor)	Overloaded. Indicates whether the specified context supports painting a representation of an object's value within the specified context.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.

PaintValue (inherited from UITypeEditor)	Overloaded. Paints a representation of the value of an object using the specified PaintValueEventArgs.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>AttributesEditor Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AttributesEditor Constructor

Initializes a new instance of the AttributesEditor class.

public AttributesEditor();

See Also

AttributesEditor Class | NDoc.Core Namespace

Copyright $\ @$ 2000-2003 the NDoc Team

AttributesEditor Methods

The methods of the **AttributesEditor** class are listed below. For a complete list of **AttributesEditor** class members, see the <u>AttributesEditor Members</u> topic.

Public Instance Methods

<u>*•\EditValue</u>	Overloaded. Handler called when editing a value.
■◆EditValue (inherited from UITypeEditor)	Overloaded. Edits the value of the specified object using the editor style indicated by GetEditStyle.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
 ♦ <u>GetEditStyle</u>	Overloaded. Returns the edit style for the type.
	Overloaded. Gets the editor style used by the EditValue method.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetPaintValueSupported (inherited from UITypeEditor)	Overloaded. Indicates whether the specified context supports painting a representation of an object's value within the specified context.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
PaintValue (inherited from UITypeEditor)	Overloaded. Paints a representation of the value of an

	object using the specified PaintValueEventArgs .
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

AttributesEditor Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesEditor.EditValue Method

Handler called when editing a value.

Overload List

Handler called when editing a value.

<u>public override object EditValue(ITypeDescriptorContext,IServicePrc</u> Inherited from <u>UITypeEditor</u>.

public object EditValue(IServiceProvider,object);

See Also

AttributesEditor Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesEditor.EditValue Method (ITypeDescriptorContext IServiceProvider, Object)

Handler called when editing a value.

```
public override object EditValue(
    ITypeDescriptorContext context,
    IServiceProvider provider,
    object value
);
```

Parameters

context
Context
provider
Provider
value
Current Value

Return Value

New value

See Also

AttributesEditor Class | NDoc.Core Namespace | AttributesEditor.EditValue Overload List

Copyright © 2000-2003 the NDoc Team

AttributesEditor.GetEditStyle Method

Returns the edit style for the type.

Overload List

Inherited from **UITypeEditor**.

public UITypeEditorEditStyle GetEditStyle();

Returns the edit style for the type.

public override UITypeEditorEditStyle GetEditStyle(ITypeDescriptor(

See Also

AttributesEditor Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesEditor.GetEditStyle Method (ITypeDescriptorCont

Returns the edit style for the type.

```
public override UITypeEditorEditStyle GetEdit
    ITypeDescriptorContext context
);
```

Parameters

context Context

Return Value

Edit Style

See Also

<u>AttributesEditor Class | NDoc.Core Namespace | AttributesEditor.GetEditStyle Overload List</u>

Copyright © 2000-2003 the NDoc Team

AttributesForm Class

A form to build attributes filter criteria.

For a list of all members of this type, see <u>AttributesForm Members</u>.

```
System.Object MarshalByRefObject
Component
Control
ScrollableControl
ContainerControl
Form
AttributesForm
```

public class AttributesForm : Form

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

AttributesForm Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesForm Members

AttributesForm overview

Public Instance Constructors

AttributesForm Constructor	Creates and initialize a new
	AttributesForm object.

Public Instance Fields

∲ <u>Value</u>	Contains the updated value, if
	the user clicked OK.

Public Instance Properties

Form)	Gets or sets the button on the form that is clicked when the user presses the ENTER key.
<u>AccessibilityObject</u> (inherited from Control)	Gets the <u>AccessibleObject</u> assigned to the control.
AccessibleDefaultActionDescription (inherited from Control)	Gets or sets the default action description of the control for use by accessibility client applications.
**AccessibleDescription (inherited from Control)	Gets or sets the description of the control used by accessibility client applications.
Control)	Gets or sets the name of the control used by accessibility client applications.
<u>AccessibleRole</u> (inherited from Control)	Gets or sets the accessible role of the control
<u>ActiveControl</u> (inherited from	Gets or sets the active

ContainerControl)	control on the container control.
ActiveMdiChild (inherited from Form)	Gets the currently active multiple document interface (MDI) child window.
AllowDrop (inherited from Control)	Gets or sets a value indicating whether the control can accept data that the user drags onto it.
AllowTransparency (inherited from Form)	
Anchor (inherited from Control)	Gets or sets which edges of the control are anchored to the edges of its container.
AutoScale (inherited from Form)	Gets or sets a value indicating whether the form adjusts its size to fit the height of the font used on the form and scales its controls.
<u>AutoScaleBaseSize</u> (inherited from Form)	Gets or sets the base size used for autoscaling of the form.
[™] AutoScroll (inherited from Form)	Gets or sets a value indicating whether the form enables autoscrolling.
<u>AutoScrollMargin</u> (inherited from ScrollableControl)	Gets or sets the size of the auto-scroll margin.
AutoScrollMinSize (inherited from ScrollableControl)	Gets or sets the minimum size of the auto-scroll.
<u>AutoScrollPosition</u> (inherited from ScrollableControl)	Gets or sets the location of the auto-scroll position.
<u>BackColor</u> (inherited from Form)	
<u>BackgroundImage</u> (inherited from	Gets or sets the background

Control)	image displayed in the control.
<u>BindingContext</u> (inherited from ContainerControl)	
™ Bottom (inherited from Control)	Gets the distance between the bottom edge of the control and the top edge of its container's client area.
Bounds (inherited from Control)	Gets or sets the size and location of the control including its nonclient elements.
	Gets or sets the button control that is clicked when the user presses the ESC key.
	Gets a value indicating whether the control can receive focus.
	Gets a value indicating whether the control can be selected.
<u>Capture</u> (inherited from Control)	Gets or sets a value indicating whether the control has captured the mouse.
CausesValidation (inherited from Control)	Gets or sets a value indicating whether the control causes validation to be performed on any controls that require validation when it receives focus.
ClientRectangle (inherited from Control)	Gets the rectangle that represents the client area of the control.

<u>ClientSize</u> (inherited from Form)	Gets or sets the size of the client area of the form.
<u>CompanyName</u> (inherited from Control)	Gets the name of the company or creator of the application containing the control.
Container (inherited from Component)	Gets the <u>IContainer</u> that contains the <u>Component</u> .
	Gets a value indicating whether the control, or one of its child controls, currently has the input focus.
ContextMenu (inherited from Control)	Gets or sets the shortcut menu associated with the control.
<u>ControlBox</u> (inherited from Form)	Gets or sets a value indicating whether a control box is displayed in the caption bar of the form.
<u>Controls</u> (inherited from Control)	Gets the collection of controls contained within the control.
<u>Created</u> (inherited from Control)	Gets a value indicating whether the control has been created.
[™] Cursor (inherited from Control)	Gets or sets the cursor that is displayed when the mouse pointer is over the control.
<u>□ DataBindings</u> (inherited from Control)	Gets the data bindings for the control.
	Gets or sets the size and location of the form on the Windows desktop.

	Gets or sets the location of the form on the Windows desktop.
<u>DialogResult</u> (inherited from Form)	Gets or sets the dialog result for the form.
<u>□ DisplayRectangle</u> (inherited from ScrollableControl)	
	Gets a value indicating whether the control is in the process of being disposed of.
<u>□Dock</u> (inherited from Control)	Gets or sets which edge of the parent container a control is docked to.
	Gets the dock padding settings for all edges of the control.
<u>Enabled</u> (inherited from Control)	Gets or sets a value indicating whether the control can respond to user interaction.
Focused (inherited from Control)	Gets a value indicating whether the control has input focus.
Font (inherited from Control)	Gets or sets the font of the text displayed by the control.
ForeColor (inherited from Control)	Gets or sets the foreground color of the control.
FormBorderStyle (inherited from Form)	Gets or sets the border style of the form.
Handle (inherited from Control)	Gets the window handle that the control is bound to.
	Gets a value indicating whether the control contains

	one or more child controls.
<u>Height</u> (inherited from Control)	Gets or sets the height of the control.
<u>HelpButton</u> (inherited from Form)	Gets or sets a value indicating whether a Help button should be displayed in the caption box of the form.
<u>□ lcon</u> (inherited from Form)	Gets or sets the icon for the form.
<u>ImeMode</u> (inherited from Control)	Gets or sets the Input Method Editor (IME) mode of the control.
InvokeRequired (inherited from Control)	Gets a value indicating whether the caller must call an invoke method when making method calls to the control because the caller is on a different thread than the one the control was created on.
□ IsAccessible (inherited from Control)	Gets or sets a value indicating whether the control is visible to accessibility applications.
□ IsDisposed (inherited from Control)	Gets a value indicating whether the control has been disposed of.
□ IsHandleCreated (inherited from Control)	Gets a value indicating whether the control has a handle associated with it.
■IsMdiChild (inherited from Form)	Gets a value indicating whether the form is a multiple document interface (MDI) child form.

	Gets or sets a value indicating whether the form is a container for multiple document interface (MDI) child forms.
<u>IsRestrictedWindow</u> (inherited from Form)	
	Gets or sets a value indicating whether the form will receive key events before the event is passed to the control that has focus.
<u>Left</u> (inherited from Control)	Gets or sets the x-coordinate of a control's left edge in pixels.
<u>Location</u> (inherited from Control)	Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
MaximizeBox (inherited from Form)	Gets or sets a value indicating whether the maximize button is displayed in the caption bar of the form.
MaximumSize (inherited from Form)	Gets the maximum size the form can be resized to.
MdiChildren (inherited from Form)	Gets an array of forms that represent the multiple document interface (MDI) child forms that are parented to this form.
MdiParent (inherited from Form)	Gets or sets the current multiple document interface (MDI) parent form of this form.

[™] Menu (inherited from Form)	Gets or sets the MainMenu that is displayed in the form.	
<u>MergedMenu</u> (inherited from Form)	Gets the merged menu for the form.	
MinimizeBox (inherited from Form)	Gets or sets a value indicating whether the minimize button is displayed in the caption bar of the form.	
MinimumSize (inherited from Form)	Gets or sets the minimum size the form can be resized to.	
<u>Modal</u> (inherited from Form)	Gets a value indicating whether this form is displayed modally.	
<u>Name</u> (inherited from Control)	Gets or sets the name of the control.	
<u>Opacity</u> (inherited from Form)	Gets or sets the opacity level of the form.	
	Gets an array of Form objects that represent all forms that are owned by this form.	
<u>Sowner</u> (inherited from Form)	Gets or sets the form that owns this form.	
Parent (inherited from Control)	Gets or sets the parent container of the control.	
	Gets the form that the container control is assigned to.	
	Gets the product name of the assembly containing the control.	
ProductVersion (inherited from	Gets the version of the	

Control)	assembly containing the control.
RecreatingHandle (inherited from Control)	Gets a value indicating whether the control is currently re-creating its handle.
■ Region (inherited from Control)	Gets or sets the window region associated with the control.
■ Right (inherited from Control)	Gets the distance between the right edge of the control and the left edge of its container.
	Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts.
ShowInTaskbar (inherited from Form)	Gets or sets a value indicating whether the form is displayed in the Windows taskbar.
Site (inherited from Control)	Gets or sets the site of the control.
Size (inherited from Form)	Gets or sets the size of the form.
SizeGripStyle (inherited from Form)	Gets or sets the style of the size grip to display in the lower-right corner of the form.
StartPosition (inherited from Form)	Gets or sets the starting position of the form at run time.
<u>■TabIndex</u> (inherited from Form)	

<u>■ TabStop</u> (inherited from Control)	Gets or sets a value indicating whether the user can give the focus to this control using the TAB key.
<u>Tag</u> (inherited from Control)	Gets or sets the object that contains data about the control.
Text (inherited from Control)	Gets or sets the text associated with this control.
<u>™Top</u> (inherited from Control)	Gets or sets the y-coordinate of the control's top edge in pixels.
™ TopLevel (inherited from Form)	Gets or sets a value indicating whether to display the form as a top-level window.
TopLevelControl (inherited from Control)	Gets the parent control that is not parented by another Windows Forms control. Typically, this is the outermost Form that the control is contained in.
TopMost (inherited from Form)	Gets or sets a value indicating whether the form should be displayed as the top-most form of your application.
TransparencyKey (inherited from Form)	Gets or sets the color that will represent transparent areas of the form.
	Gets or sets a value indicating whether the control is displayed.
<u>™Width</u> (inherited from Control)	Gets or sets the width of the

	control.
<u>WindowState</u> (inherited from Form)	Gets or sets the form's window state.
<u>™WindowTarget</u> (inherited from Control)	

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
**BringToFront (inherited from Control)	Brings the control to the front of the z-order.
<u>○••Close</u> (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
**CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
**CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy

	used to communicate with a remote object.
Dispose (inherited from Component)	Overloaded. Releases all resources used by the Component.
DoDragDrop (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
*•GetChildAtPoint (inherited from Control)	Retrieves the child control that is located at the specified coordinates.
GetContainerControl (inherited from Control)	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
GetNextControl (inherited from Control)	Retrieves the next control forward or back in the tab order

	of child controls.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
Hide (inherited from Control)	Conceals the control from the user.
InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
Invoke (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
►• LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
PointToClient (inherited from Control)	Computes the location of the specified screen point into client coordinates.
PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
PreProcessMessage	Preprocesses input messages

(inherited from Control)	within the message loop before they are dispatched.
**RectangleToClient (inherited from Control)	Computes the size and location of the specified screen rectangle in client coordinates.
**RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
****RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.

Scale (inherited from Control)	Overloaded. Scales the control and any child controls to the specified ratio.
**ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>ToString</u> (inherited from Form)	
■ Update (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
<u>Validate</u> (inherited from	Validates the last invalidated

ContainerControl)	control and its ancestors up
	through, but not including, the
	current control.

Public Instance Events

Activated (inherited from Form)	Occurs when the form is activated in code or by the user.
BackColorChanged (inherited from Control)	Occurs when the value of the BackColor property changes.
BackgroundImageChanged (inherited from Control)	Occurs when the value of the BackgroundImage property changes.
BindingContextChanged (inherited from Control)	Occurs when the value of the BindingContext property changes.
CausesValidationChanged (inherited from Control)	Occurs when the value of the CausesValidation property changes.
ChangeUICues (inherited from Control)	Occurs when the focus or keyboard user interface (UI) cues change.
Click (inherited from Control)	Occurs when the control is clicked.
Closed (inherited from Form)	Occurs when the form is closed.
Closing (inherited from Form)	Occurs when the form is closing.
ContextMenuChanged (inherited from Control)	Occurs when the value of the ContextMenu property changes.
ControlAdded (inherited from Control)	Occurs when a new control is added to the <u>ControlCollection</u> .
ControlRemoved (inherited from Control)	Occurs when a control is removed from the ControlCollection .

CursorChanged (inherited from Control)	Occurs when the value of the Cursor property changes.
Deactivate (inherited from Form)	Occurs when the form loses focus and is not the active form.
Disposed (inherited from Component)	Adds an event handler to listen to the <u>Disposed</u> event on the component.
DockChanged (inherited from Control)	Occurs when the value of the Dock property changes.
DoubleClick (inherited from Control)	Occurs when the control is double-clicked.
DragDrop (inherited from Control)	Occurs when a drag-and-drop operation is completed.
<pre> DragEnter (inherited from Control) </pre>	Occurs when an object is dragged into the control's bounds.
DragLeave (inherited from Control)	Occurs when an object is dragged out of the control's bounds.
DragOver (inherited from Control)	Occurs when an object is dragged over the control's bounds.
EnabledChanged (inherited from Control)	Occurs when the <u>Enabled</u> property value has changed.
Fenter (inherited from Control)	Occurs when the control is entered.
FontChanged (inherited from Control)	Occurs when the <u>Font</u> property value changes.
ForeColorChanged (inherited from Control)	Occurs when the <u>ForeColor</u> property value changes.
GiveFeedback (inherited from Control)	Occurs during a drag operation.

GotFocus (inherited from Control)	Occurs when the control receives focus.
<u>HandleCreated</u> (inherited from Control)	Occurs when a handle is created for the control.
F HandleDestroyed (inherited from Control)	Occurs when the control's handle is in the process of being destroyed.
HelpRequested (inherited from Control)	Occurs when the user requests help for a control.
ImeModeChanged (inherited from Control)	Occurs when the <u>ImeMode</u> property has changed.
InputLanguageChanged (inherited from Form)	Occurs after the input language of the form has changed.
InputLanguageChanging (inherited from Form)	Occurs when the user attempts to change the input language for the form.
Invalidated (inherited from Control)	Occurs when a control's display requires redrawing.
KeyDown (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyPress (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyUp (inherited from Control)	Occurs when a key is released while the control has focus.
Layout (inherited from Control)	Occurs when a control should reposition its child controls.
Leave (inherited from Control)	Occurs when the input focus leaves the control.
Load (inherited from Form)	Occurs before a form is displayed for the first time.
LocationChanged (inherited from Control)	Occurs when the <u>Location</u> property value has changed.

LostFocus (inherited from Control)	Occurs when the control loses focus.
MaximizedBoundsChanged (inherited from Form)	Occurs when the value of the MaximizedBounds property has changed.
MaximumSizeChanged (inherited from Form)	Occurs when the value of the MaximumSize property has changed.
MdiChildActivate (inherited from Form)	Occurs when a multiple document interface (MDI) child form is activated or closed within an MDI application.
MenuComplete (inherited from Form)	Occurs when the menu of a form loses focus.
MenuStart (inherited from Form)	Occurs when the menu of a form receives focus.
MinimumSizeChanged (inherited from Form)	Occurs when the value of the MinimumSize property has changed.
MouseDown (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is pressed.
MouseEnter (inherited from Control)	Occurs when the mouse pointer enters the control.
MouseHover (inherited from Control)	Occurs when the mouse pointer hovers over the control.
MouseLeave (inherited from Control)	Occurs when the mouse pointer leaves the control.
MouseMove (inherited from Control)	Occurs when the mouse pointer is moved over the control.
MouseUp (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is released.

MouseWheel (inherited from Control)	Occurs when the mouse wheel moves while the control has focus.
Move (inherited from Control)	Occurs when the control is moved.
Paint (inherited from Control)	Occurs when the control is redrawn.
ParentChanged (inherited from Control)	Occurs when the <u>Parent</u> property value changes.
QueryAccessibilityHelp (inherited from Control)	Occurs when <u>AccessibleObject</u> is providing help to accessibility applications.
QueryContinueDrag (inherited from Control)	Occurs during a drag-and-drop operation and allows the drag source to determine whether the drag-and-drop operation should be canceled.
Resize (inherited from Control)	Occurs when the control is resized.
RightToLeftChanged (inherited from Control)	Occurs when the RightToLeft property value changes.
SizeChanged (inherited from Control)	Occurs when the <u>Size</u> property value changes.
StyleChanged (inherited from Control)	Occurs when the control style changes.
SystemColorsChanged (inherited from Control)	Occurs when the system colors change.
TablndexChanged (inherited from Form)	
TabStopChanged (inherited from Control)	Occurs when the <u>TabStop</u> property value changes.
TextChanged (inherited from	Occurs when the <u>Text</u> property

Control)	value changes.
Validated (inherited from Control)	Occurs when the control is finished validating.
Validating (inherited from Control)	Occurs when the control is validating.
VisibleChanged (inherited from Control)	Occurs when the <u>Visible</u> property value changes.

Protected Instance Properties

	Gets the default Input Method Editor (IME) mode supported by the control.
	Gets a value that indicates whether the <u>Component</u> is currently in design mode.
Events (inherited from Component)	Gets the list of event handlers that are attached to this Component .
FontHeight (inherited from Control)	Gets or sets the height of the font of the control.
	Gets or sets a value indicating whether the horizontal scroll bar is visible.
MaximizedBounds (inherited from Form)	Gets and sets the size of the form when it is maximized.
RenderRightToLeft (inherited from Control)	
ResizeRedraw (inherited from	Gets or sets a value indicating

Control)	whether the control redraws itself when resized.
	Gets a value indicating whether the control should display focus rectangles.
ShowKeyboardCues (inherited from Control)	Gets a value indicating whether the control should display keyboard shortcuts.
✓ VScroll (inherited from ScrollableControl)	Gets or sets a value indicating whether the vertical scroll bar is visible.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
<u>ActivateMdiChild</u> (inherited from Form)	
<u>AdjustFormScrollbars</u> (inherited from Form)	
★ApplyAutoScaling (inherited from Form)	
<u>CenterToParent</u> (inherited from Form)	
<u>CenterToScreen</u> (inherited from Form)	
<u>◆CreateAccessibilityInstance</u> (inherited from Control)	Creates a new accessibility object for the control.
<u>◆CreateControlsInstance</u> (inherited from Form)	

DefWndProc (inherited from Form)	
<u>PostroyHandle</u> (inherited from Control)	Destroys the handle associate with the control.
₱ Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resource and performs other cleanup operations before the Component is reclaimed by garbage collection.
GetScrollState (inherited from ScrollableControl)	
	Returns an object that represents a service provided the Component or by its Container.
	Retrieves the value of the specified control style bit for the control.
GetTopLevel (inherited from Control)	Determines if the control is a top-level control.
InitLayout (inherited from Control)	Called after the control has be added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event fo the specified control.
InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event for the specified control.
InvokeOnClick (inherited from Control)	Raises the <u>Click</u> event for the specified control.
InvokePaint (inherited from Control)	Raises the Paint event for the specified control.
₱InvokePaintBackground (inherited)	Raises the PaintBackground

from Control)	event for the specified control.
	Determines if a character is ar input character that the contro recognizes.
	Determines whether the specified key is a regular inputey or a special key that requires preprocessing.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
◆OnActivated (inherited from Form)	Raises the <u>Activated</u> event.
OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChange</u> event.
OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> even
OnCausesValidationChanged (inherited from Control)	Raises the <u>CausesValidationChanged</u> eve
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUICues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
◆OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
◆OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
<u>◆OnContextMenuChanged</u> (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the <u>ControlAdded</u> eve

OnControlRemoved (inherited from ContainerControl)	
OnCreateControl (inherited from Form)	
OnCursorChanged (inherited from Control)	Raises the <u>CursorChanged</u> event.
[♠] OnDeactivate (inherited from Form)	Raises the <u>Deactivate</u> event.
OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> eve
OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event
OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
OnDragLeave (inherited from Control)	Raises the <u>DragLeave</u> event.
OnDragOver (inherited from Control)	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
<u>OnEnter</u> (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
OnForeColorChanged (inherited from Control)	Raises the <u>ForeColorChanged</u> event.
OnGiveFeedback (inherited from Control)	Raises the GiveFeedback eve
	Raises the <u>GotFocus</u> event.

[♣] OnHandleCreated (inherited from Form)	
[→] OnHandleDestroyed (inherited from Form)	
<u>OnHelpRequested</u> (inherited from Control)	Raises the <u>HelpRequested</u> event.
OnlmeModeChanged (inherited from Control)	Raises the <u>ImeModeChanged</u> event.
<u>→OnInputLanguageChanged</u> (inherited from Form)	Raises the InputLanguageChanged event.
OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
OnInvalidated (inherited from Control)	Raises the <u>Invalidated</u> event.
OnKeyDown (inherited from Control)	Raises the <u>KeyDown</u> event.
OnKeyPress (inherited from Control)	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
[♣] OnLayout (inherited from ScrollableControl)	
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
◆OnLoad (inherited from Form)	Raises the <u>Load</u> event.
[♣] OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
	Raises the <u>LostFocus</u> event.
	Raises the MaximizedBoundsChanged event.
⋄ OnMaximumSizeChanged	Raises the

(inherited from Form)	MaximumSizeChanged event.
OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
	Raises the MenuComplete event.
OnMenuStart (inherited from Form)	Raises the MenuStart event.
OnMinimumSizeChanged (inherited from Form)	Raises the MinimumSizeChanged event.
OnMouseDown (inherited from Control)	Raises the MouseDown event
OnMouseEnter (inherited from Control)	Raises the MouseEnter event
<u>OnMouseHover</u> (inherited from Control)	Raises the <u>MouseHover</u> even
OnMouseLeave (inherited from Control)	Raises the MouseLeave even
<u>OnMouseMove</u> (inherited from Control)	Raises the <u>MouseMove</u> event
	Raises the MouseUp event.
OnMouseWheel (inherited from ScrollableControl)	
◆OnMove (inherited from Control)	Raises the Move event.
OnNotifyMessage (inherited from Control)	Notifies the control of Window messages.
◆OnPaint (inherited from Form)	
[♠] OnPaintBackground (inherited from Control)	Paints the background of the control.
<u>OnParentBackColorChanged</u> (inherited from Control)	Raises the <u>BackColorChange</u> event when the <u>BackColor</u>

	property value of the control's container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event when the BackgroundImage property value of the control's contained changes.
OnParentBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever when the <u>BindingContext</u> property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
<u>OnParentEnabledChanged</u> (inherited from Control)	Raises the <u>EnabledChanged</u> event when the <u>Enabled</u> property value of the control's container changes.
OnParentFontChanged (inherited from Control)	Raises the <u>FontChanged</u> ever when the <u>Font</u> property value the control's container change
OnParentForeColorChanged (inherited from Control)	Raises the ForeColorChange event when the ForeColor property value of the control's container changes.
<u>OnParentRightToLeftChanged</u> (inherited from Control)	Raises the RightToLeftChange event when the RightToLeft property value of the control's container changes.
OnParentVisibleChanged (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> prope value of the control's containe changes.

OnQueryContinueDrag (inherited from Control)	Raises the <u>QueryContinueDrag</u> event.
→ OnResize (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> event
[→] OnStyleChanged (inherited from Form)	
[→] OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
[→] OnTabIndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
[→] OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
[♣] OnTextChanged (inherited from Form)	
	Raises the <u>Validated</u> event.
[♣] OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
₱ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
₱ ProcessDialogKey (inherited from Form)	
₱ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate

	control events.
ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
ProcessTabKey (inherited from Form)	
RaiseDragEvent (inherited from Control)	
RaiseKeyEvent (inherited from Control)	
	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
*RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment t the appropriate HorizontalAlignment to support right-to-left text.
*RtlTranslateContent (inherited from Control)	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
*RtlTranslateHorizontal (inherited from Control)	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
	appropriate ContentAlignment support right-to-left text. Converts the specified HorizontalAlignment to the appropriate HorizontalAlignment

	Converts the specified LeftRightAlignment to the appropriate LeftRightAlignment to support right-to-left text.
♦ Select (inherited from Form)	Overloaded.
★SetBoundsCore (inherited from Form)	
★SetClientSizeCore (inherited from Form)	
★SetDisplayRectLocation (inherited from ScrollableControl)	
★SetScrollState (inherited from ScrollableControl)	
★SetStyle (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
	Overloaded. Updates the bounds of the control with the current size and location.
<u>DupdateDefaultButton</u> (inherited from Form)	
<u>DupdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.
<u>WndProc</u> (inherited from Form)	

Protected Internal Instance Methods

<u></u> <u>ProcessKeyMessage</u>	Processes a keyboard
(inherited from Control)	message.

Explicit Interface Implementations



See Also

AttributesForm Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesForm Constructor

Creates and initialize a new AttributesForm object.

```
public AttributesForm(
   object val
);
```

Parameters

val

See Also

AttributesForm Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesForm Fields

The fields of the **AttributesForm** class are listed below. For a complete list of **AttributesForm** class members, see the <u>AttributesForm Members</u> topic.

Public Instance Fields

<u> </u>	Contains the updated value, if
	the user clicked OK.

See Also

<u>AttributesForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AttributesForm.Value Field

Contains the updated value, if the user clicked OK.

public string Value;

See Also

<u>AttributesForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AttributesForm Methods

The methods of the **AttributesForm** class are listed below. For a complete list of **AttributesForm** class members, see the <u>AttributesForm Members</u> topic.

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
**BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
■ BringToFront (inherited from Control)	Brings the control to the front of the z-order.
<u>◆Close</u> (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a

	remote object.
Dispose (inherited from Component)	Overloaded. Releases all resources used by the <u>Component</u> .
DoDragDrop (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
from Control)	Retrieves the child control that is located at the specified coordinates.
GetContainerControl (inherited from Control)	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
from Control) (inherited	Retrieves the next control forward or back in the tab order of child controls.

GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>Hide</u> (inherited from Control)	Conceals the control from the user.
**InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
••••Invalidate (inherited from Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
<u>Nove</u> (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
►•• LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
<u>PointToClient</u> (inherited from Control)	Computes the location of the specified screen point into client coordinates.
PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
**PreProcessMessage (inherited from Control)	Preprocesses input messages within the message loop before

	they are dispatched.
**RectangleToClient (inherited from Control)	Computes the size and location of the specified screen rectangle in client coordinates.
***RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.
Scale (inherited from Control)	Overloaded. Scales the control

	and any child controls to the specified ratio.
**ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>◆ToString</u> (inherited from Form)	
■ Update (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
<u>Validate</u> (inherited from ContainerControl)	Validates the last invalidated control and its ancestors up

through, but not including, the current control.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
*ActivateMdiChild (inherited from Form)	
AdjustFormScrollbars (inherited from Form)	
ApplyAutoScaling (inherited from Form)	
<u>◆CenterToParent</u> (inherited from Form)	
<u>CenterToScreen</u> (inherited from Form)	
<u>◆CreateAccessibilityInstance</u> (inherited from Control)	Creates a new accessibility object for the control.
<u>◆CreateControlsInstance</u> (inherited from Form)	
<u>◆CreateHandle</u> (inherited from Form)	
<u>▶ DefWndProc</u> (inherited from Form)	
<u>DestroyHandle</u> (inherited from Control)	Destroys the handle associated with the control.
Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resource and performs other cleanup operations before the

	Component is reclaimed by garbage collection.
	Returns an object that represents a service provided the Component or by its Container.
	Retrieves the value of the specified control style bit for t control.
	Determines if the control is a top-level control.
<u>InitLayout</u> (inherited from Control)	Called after the control has b added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event for the specified control.
№ InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event the specified control.
<u>InvokeOnClick</u> (inherited from Control)	Raises the <u>Click</u> event for the specified control.
₱InvokePaint (inherited from Control)	Raises the Paint event for the specified control.
₱InvokePaintBackground (inherited from Control)	Raises the PaintBackgroun event for the specified contro
	Determines if a character is a input character that the control recognizes.
	Determines whether the specified key is a regular inpukey or a special key that requires preprocessing.

MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
<u>→OnActivated</u> (inherited from Form)	Raises the <u>Activated</u> event.
[♠] OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChanged</u> event.
*OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the BindingContextChangedevent.
[♠] OnCausesValidationChanged (inherited from Control)	Raises the <u>CausesValidationChanged</u> ever
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUICues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
◆OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
[♠] OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
[♣] OnContextMenuChanged (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the <u>ControlAdded</u> event
OnControlRemoved (inherited from ContainerControl)	
◆OnCreateControl (inherited from Form)	
OnCursorChanged (inherited from Control)	Raises the CursorChangedevent.
	Raises the <u>Deactivate</u> event.

OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> even
◆OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event.
♠OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
◆OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
<u>OnDragLeave</u> (inherited from Control)	Raises the <u>DragLeave</u> event.
	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
OnEnter (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
MONForeColorChanged (inherited from Control)	Raises the <u>ForeColorChanged</u> event.
OnGiveFeedback (inherited from Control)	Raises the <u>GiveFeedback</u> eve
	Raises the <u>GotFocus</u> event.
Porm of the contract of	
PONHandleDestroyed (inherited from Form)	
OnHelpRequested (inherited from Control)	Raises the <u>HelpRequested</u> event.
<u>PONIMeModeChanged</u> (inherited from Control)	Raises the <u>ImeModeChanged</u> event.

OnInputLanguageChanged (inherited from Form)	Raises the InputLanguageChanged event.
◆OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
	Raises the <u>Invalidated</u> event.
	Raises the <u>KeyDown</u> event.
	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
OnLoad (inherited from Form)	Raises the <u>Load</u> event.
OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
	Raises the <u>LostFocus</u> event.
	Raises the MaximizedBoundsChanged event.
	Raises the MaximumSizeChanged event.
OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
	Raises the MenuComplete event.
	Raises the MenuStart event.
◆OnMinimumSizeChanged	Raises the

(inherited from Form)	MinimumSizeChanged event.
OnMouseDown (inherited from Control)	Raises the <u>MouseDown</u> event
OnMouseEnter (inherited from Control)	Raises the MouseEnter event.
OnMouseHover (inherited from Control)	Raises the MouseHover even
OnMouseLeave (inherited from Control)	Raises the MouseLeave even
OnMouseMove (inherited from Control)	Raises the MouseMove event
[♠] OnMouseUp (inherited from Control)	Raises the MouseUp event.
OnMouseWheel (inherited from ScrollableControl)	
◆OnMove (inherited from Control)	Raises the <u>Move</u> event.
OnNotifyMessage (inherited from Control)	Notifies the control of Window messages.
[♠] OnPaint (inherited from Form)	
OnPaintBackground (inherited from Control)	Paints the background of the control.
<u>OnParentBackColorChanged</u> (inherited from Control)	Raises the <u>BackColorChange</u> event when the <u>BackColor</u> property value of the control's container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event when the BackgroundImage property value of the control's containe changes.
♦ OnParentBindingContextChanged	Raises the

(inherited from Control)	BindingContextChanged event when the BindingContext property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
<u>OnParentEnabledChanged</u> (inherited from Control)	Raises the EnabledChanged event when the Enabled property value of the control's container changes.
<u>OnParentFontChanged</u> (inherited from Control)	Raises the <u>FontChanged</u> even when the <u>Font</u> property value of the control's container changes
	Raises the ForeColorChanged event when the ForeColor property value of the control's container changes.
*OnParentRightToLeftChanged (inherited from Control)	Raises the RightToLeftChange event when the RightToLeft property value of the control's container changes.
<u>OnParentVisibleChanged</u> (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> propert value of the control's container changes.
OnQueryContinueDrag (inherited from Control)	Raises the <u>QueryContinueDrai</u> event.
<u>OnResize</u> (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> even

OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
OnTablndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
OnTextChanged (inherited from Form)	
OnValidated (inherited from Control)	Raises the <u>Validated</u> event.
OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
ProcessDialogKey (inherited from Form)	
ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate control events.
ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
ProcessTabKey (inherited from Form)	
₱RaiseDragEvent (inherited from	

Control)	
RaiseKeyEvent (inherited from Control)	
RaiseMouseEvent (inherited from Control)	
	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment t the appropriate HorizontalAlignment to support right-to-left text.
RtlTranslateContent (inherited from Control)	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
*RtlTranslateHorizontal (inherited from Control)	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
₱RtlTranslateLeftRight (inherited from Control)	Converts the specified LeftRightAlignment to the appropriate LeftRightAlignment to support right-to-left text.
Select (inherited from Form)	Overloaded.
SetBoundsCore (inherited from Form)	

★ SetClientSizeCore (inherited from Form)	
* SetDisplayRectLocation (inherited from ScrollableControl)	
★SetScrollState (inherited from ScrollableControl)	
<u>SetStyle</u> (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
★SetVisibleCore (inherited from Form)	
<u>DupdateBounds</u> (inherited from Control)	Overloaded. Updates the bounds of the control with the current size and location.
<u>Description</u> value of the property of th	
<u>◆UpdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.

Protected Internal Instance Methods

<u></u> <u>ProcessKeyMessage</u>	Processes a keyboard
(inherited from Control)	message.

Explicit Interface Implementations

IContainerControl.ActivateControl (inherited from	
ContainerControl)	

See Also

AttributesForm Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesForm.Dispose Method

Clean up any resources being used.

Overload List

Inherited from **Component**.

public void Dispose();

Clean up any resources being used.

protected override void Dispose(bool);

See Also

AttributesForm Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributesForm.Dispose Method (Boolean)

Clean up any resources being used.

```
protected override void Dispose(
   bool disposing
);
```

See Also

<u>AttributesForm Class | NDoc.Core Namespace | AttributesForm.Dispose Overload List</u>

Copyright © 2000-2003 the NDoc Team

AttributeToShow Class

Class which holds information about an attribute.

For a list of all members of this type, see <u>AttributeToShow Members</u>.

System.Object AttributeToShow

public class AttributeToShow

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

AttributeToShow Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributeToShow Members

AttributeToShow overview

Public Instance Constructors

Creates an empty
AttributeToShow object

Public Instance Fields

▶ PropertiesToShow	List of properties
--------------------	--------------------

Public Instance Properties

<u>™Name</u>	Name attribute
--------------	----------------

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>◆ToString</u> (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited)	Creates a shallow copy of the

from	Object)
пош	Object

current Object.

See Also

<u>AttributeToShow Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

AttributeToShow Constructor

Creates an empty AttributeToShow object

public AttributeToShow();

See Also

AttributeToShow Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributeToShow Fields

The fields of the **AttributeToShow** class are listed below. For a complete list of **AttributeToShow** class members, see the <u>AttributeToShow Members</u> topic.

Public Instance Fields

◆ Properties To Show	List of properties
----------------------	--------------------

See Also

AttributeToShow Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributeToShow.PropertiesToShow Field

List of properties

public ArrayList PropertiesToShow;

See Also

AttributeToShow Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributeToShow Properties

The properties of the **AttributeToShow** class are listed below. For a complete list of **AttributeToShow** class members, see the <u>AttributeToShow Members</u> topic.

Public Instance Properties

<u>™Name</u>	Name attribute
--------------	----------------

See Also

AttributeToShow Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

AttributeToShow.Name Property

Name attribute

```
public string Name {get; set;}
```

See Also

AttributeToShow Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter Class

Provides the base class for documenters.

For a list of all members of this type, see <u>BaseDocumenter</u> Members.

System.Object BaseDocumenter

```
public abstract class BaseDocumenter :
   IDocumenter, IComparable
```

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

BaseDocumenter Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter Members

BaseDocumenter overview

Public Static Methods

LoadAssembly	Loads an assembly.
--------------	--------------------

Public Instance Properties

™ Config	See <u>IDocumenter</u> .
<u>™DevelopmentStatus</u>	The development status (alpha, beta, stable) of this documenter. Documenters should override this if they aren't stable.
<u>™MainOutputFile</u>	See <u>IDocumenter</u> .
<u>™Name</u>	Gets the display name for the documenter.

Public Instance Methods

⇒ Build	See <u>IDocumenter</u> .
→ CanBuild	Overloaded. See <u>IDocumenter</u> .
<u> </u>	See <u>IDocumenter</u> .
* <u>CompareTo</u>	Compares the currrent document to another documenter.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.

<u>Nostring</u> (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
≅• <mark>View</mark>	See <u>IDocumenter</u> .

Public Instance Events

DocBuildingProgress	See <u>IDocumenter</u> .
[₱] DocBuildingStep	See <u>IDocumenter</u> .

Protected Instance Constructors

BaseDocumenter Constructor	Initialized a new
	BaseDocumenter instance.

Protected Instance Properties

[♠] Document	Gets the XmlDocument containing the combined relected metadata and /doc comments.
	The XML string containing the combined relected metadata and /doc comments.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
* GetXPathDocument	Returns the XPathDocument to access the combined relected metadata and /doc comments.
MakeXml MakeXml	Builds an XmlDocument combining the reflected metadata with the /doc

	comments.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
<u> →OnDocBuildingProgress</u>	Raises the DocBuildingProgress event.
<u>OnDocBuildingStep</u>	Raises the DocBuildingStep event.
*SetupAssemblyResolver	Setup AssemblyResolver for case where system doesn't resolve an assembly automatically. This puts in the directories in ReferencesPath, and the directories to each assembly referenced in the project.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright $\ @$ 2000-2003 the NDoc Team

BaseDocumenter Constructor

Initialized a new BaseDocumenter instance.

```
protected BaseDocumenter(
   string name
);
```

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter Properties

The properties of the **BaseDocumenter** class are listed below. For a complete list of **BaseDocumenter** class members, see the <u>BaseDocumenter Members</u> topic.

Public Instance Properties

<u>Config</u>	See <u>IDocumenter</u> .
<u>™DevelopmentStatus</u>	The development status (alpha, beta, stable) of this documenter. Documenters should override this if they aren't stable.
<u>™MainOutputFile</u>	See <u>IDocumenter</u> .
<u>™Name</u>	Gets the display name for the documenter.

Protected Instance Properties

[♠] Document	Gets the XmlDocument containing the combined relected metadata and /doc comments.
[™] XmlBuffer	The XML string containing the combined relected metadata and /doc comments.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.Config Property

See **IDocumenter**.

```
public IDocumenterConfig Config {get; set;}
```

Implements

IDocumenter.Config

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.DevelopmentStatus Property

The development status (alpha, beta, stable) of this documenter. Documenters should override this if they aren't stable.

public virtual <u>DocumenterDevelopmentStatus</u> De

Implements

IDocumenter.DevelopmentStatus

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.Document Property

Gets the XmlDocument containing the combined relected metadata and /doc comments.

protected System.Xml.XmlDocument {ge

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.MainOutputFile Property

See **IDocumenter**.

public abstract string MainOutputFile {get;}

Implements

IDocumenter.MainOutputFile

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.Name Property

Gets the display name for the documenter.

```
public string Name {get;}
```

Implements

IDocumenter.Name

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.XmlBuffer Property

The XML string containing the combined relected metadata and /doc comments.

```
protected string XmlBuffer {get;}
```

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter Methods

The methods of the **BaseDocumenter** class are listed below. For a complete list of **BaseDocumenter** class members, see the <u>BaseDocumenter Members</u> topic.

Public Static Methods

■ S LoadAssembly	Loads an assembly.
-------------------------	--------------------

Public Instance Methods

≅• <u>Build</u>	See <u>IDocumenter</u> .
→ CanBuild	Overloaded. See <u>IDocumenter</u> .
<u> </u>	See <u>IDocumenter</u> .
* CompareTo	Compares the currrent document to another documenter.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
- ♦ View	See <u>IDocumenter</u> .

Protected Instance Methods

`	Allows an Object to attempt to
Object)	free resources and perform
	other cleanup operations before

	the Object is reclaimed by garbage collection.
* GetXPathDocument	Returns the XPathDocument to access the combined relected metadata and /doc comments.
MakeXml MakeXml MakeXm	Builds an XmlDocument combining the reflected metadata with the /doc comments.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
<u>◆OnDocBuildingProgress</u>	Raises the DocBuildingProgress event.
<u>OnDocBuildingStep</u>	Raises the DocBuildingStep event.
*SetupAssemblyResolver	Setup AssemblyResolver for case where system doesn't resolve an assembly automatically. This puts in the directories in ReferencesPath, and the directories to each assembly referenced in the project.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright $\ensuremath{\mathbb{C}}$ 2000-2003 the NDoc Team

BaseDocumenter.Build Method

See **IDocumenter**.

```
public abstract void Build(
    Project project
);
```

Implements

IDocumenter.Build

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.CanBuild Method

See **IDocumenter**.

Overload List

See **IDocumenter**.

public virtual string CanBuild(Project);

See **IDocumenter**.

public virtual string CanBuild(Project, bool);

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.CanBuild Method (Project)

See **IDocumenter**.

```
public virtual string CanBuild(
    Project project
);
```

Implements

IDocumenter.CanBuild

See Also

<u>BaseDocumenter Class</u> | <u>NDoc.Core Namespace</u> | <u>BaseDocumenter.CanBuild Overload List</u>

Copyright $\ @ \ 2000-2003$ the NDoc Team

BaseDocumenter.CanBuild Method (Project, Boolean)

See **IDocumenter**.

```
public virtual string CanBuild(
    Project project,
    bool checkInputOnly
);
```

Implements

IDocumenter.CanBuild

See Also

BaseDocumenter Class | NDoc.Core Namespace |
BaseDocumenter.CanBuild Overload List

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.Clear Method

See **IDocumenter**.

```
public abstract void Clear();
```

Implements

IDocumenter.Clear

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.CompareTo Method

Compares the currrent document to another documenter.

```
public int CompareTo(
   object obj
);
```

Implements

IComparable.CompareTo

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.GetXPathDocument Method

Returns the XPathDocument to access the combined relected metadata and /doc comments.

protected XPathDocument GetXPathDocument();

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.LoadAssembly Method

Loads an assembly.

```
public static Assembly LoadAssembly(
    string fileName
);
```

Parameters

fileName

The assembly filename.

Return Value

The assembly object.

Remarks

This method loads an assembly into memory. If you use Assembly.Load or Assembly.LoadFrom the assembly file locks. This method doesn't lock the assembly file.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.MakeXml Method

Builds an XmlDocument combining the reflected metadata with the /doc comments.

```
protected void MakeXml(
    Project project
);
```

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.OnDocBuildingProgress Method

Raises the DocBuildingProgress event.

```
protected void OnDocBuildingProgress(
  int progress
);
```

Parameters

progress
Percentage progress value

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright $\ @ \ 2000-2003$ the NDoc Team

BaseDocumenter.OnDocBuildingStep Method

Raises the DocBuildingStep event.

```
protected void OnDocBuildingStep(
   int step,
   string label
);
```

Parameters

step

The overall percent complete value.

label

A description of the work currently beeing done.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.SetupAssemblyResolver Method

Setup AssemblyResolver for case where system doesn't resolve an assembly automatically. This puts in the directories in ReferencesPath, and the directories to each assembly referenced in the project.

```
protected AssemblyResolver SetupAssemblyResol
    Project project
);
```

Parameters

project

Remarks

The case which forced this to be so thorough is when an assembly references an unmanaged (native) dll. When the assembly is loaded, the system must also find the unmanaged dll. The rules for finding the unmanaged dll are apparently just like any other application: current working directory, the path environment variable, etc.

So in order to handle that case, we have to install an AssemblyResolver that catches the resolution failure, and uses an assembly load function that cd's to the directory which hopefully contains the unmanaged dll (see LoadAssembly()). So in this case I'm assuming that the directory containing the referencing assembly also contains the unmanaged dll.

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.View Method

See **IDocumenter**.

```
public virtual void View();
```

Implements

IDocumenter.View

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter Events

The events of the **BaseDocumenter** class are listed below. For a complete list of **BaseDocumenter** class members, see the <u>BaseDocumenter Members</u> topic.

Public Instance Events

	See <u>IDocumenter</u> .
₱ DocBuildingStep ₱ DocBu	See <u>IDocumenter</u> .

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.DocBuildingProgress Event

See **IDocumenter**.

public event <u>DocBuildingEventHandler</u> DocBuilc

Implements

<u>IDocumenter.DocBuildingProgress</u>

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenter.DocBuildingStep Event

See **IDocumenter**.

public event DocBuildingEventHandler DocBuildingEventHandler

Implements

IDocumenter.DocBuildingStep

See Also

BaseDocumenter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig Class

The base documenter config class.

For a list of all members of this type, see <u>BaseDocumenterConfig</u> Members.

System.Object BaseDocumenterConfig

public abstract class BaseDocumenterConfig
 : IDocumenterConfig

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

BaseDocumenterConfig Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig Members

BaseDocumenterConfig overview

Public Instance Constructors

■ BaseDocumenterConfig	Initializes a new instance of the
Constructor	DocumenterConfig class.

Public Instance Properties

<u>AutoDocumentConstructors</u>	Gets or sets the AutoDocumentConstructors
<u>MattoPropertyBackerSummaries</u>	Gets or sets the AutoPropertyBackerSummal property.
<u>©CopyrightHref</u>	Gets or sets the CopyrightHr
<u>CopyrightText</u>	Gets or sets the CopyrightTe
<u>□ DocumentAttributes</u>	Gets or sets whether or not t document the attributes.
<u>■DocumentedAttributes</u>	Gets or sets which attributes documented.
<u>■DocumentEmptyNamespaces</u>	Gets or sets the DocumentP property.
<u> </u>	Gets or sets the DocumentIr property.
<u> </u>	Gets or sets the DocumentP property.
<u>□DocumentProtected</u>	Gets or sets the DocumentP property.
DocumentProtectedInternalAsProtected	Gets or sets the DocumentProtectedInternal/property.
<u>■EditorBrowsableFilter</u>	Specifies the level of filtering

	EditorBrowsable attribute.
GetExternalSummaries GetExternalSummaries	Load external xml files?
<u>IncludeAssemblyVersion</u>	Gets or sets the IncludeAssemblyVersion pro
[™] ReferencesPath	Gets or sets the base directory resolve directory and assem references.
<u>ShowMissingParams</u>	Gets or sets the ShowMissin property.
[™] ShowMissingRemarks	Gets or sets the ShowMissin property.
<u>ShowMissingReturns</u>	Gets or sets the ShowMissin property.
	Gets or sets the ShowMissingSummaries pro
<u>ShowMissingValues</u>	Gets or sets the ShowMissin property.
<u>ShowTypeIdInAttributes</u>	Gets or sets whether or not t TypeId property in attributes.
	Gets or sets the SkipNamespacesWithoutSur property.
<u>■UseNamespaceDocSummaries</u>	Gets or sets the UseNamespaceDocSummar property.
<u> ■UseNDocXmlFile</u>	Gets or sets the UseNDocXr property.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.

	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetProperties	Gets a list of property names.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
≅• <u>Read</u>	Reads the previously serialized state of the documenter into memory.
<u> </u>	Associates this documenter with a project;
<u> </u>	Sets the value of a property.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
™ Write	Writes the current state of the documenter to the specified XmlWrtier.

Protected Instance Methods

	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
* SetDirty	Sets the IsDirty property on the project if any is set.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team
Generated from assembly NDoc.Core [1.2.1300.31138]

BaseDocumenterConfig Constructor

Initializes a new instance of the DocumenterConfig class.

```
public BaseDocumenterConfig(
   string name
);
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig Properties

The properties of the **BaseDocumenterConfig** class are listed below. For a complete list of **BaseDocumenterConfig** class members, see the <u>BaseDocumenterConfig Members</u> topic.

Public Instance Properties

<u>AutoDocumentConstructors</u> <u>AutoPropertyBackerSummaries</u>	Gets or sets the AutoDocumentConstructors
™AutoProportyPackorSummarios	
AutorropertybackerSummanes	Gets or sets the AutoPropertyBackerSummar property.
<u>CopyrightHref</u>	Gets or sets the CopyrightHr
<u>CopyrightText</u>	Gets or sets the CopyrightTe
<u>™DocumentAttributes</u>	Gets or sets whether or not t document the attributes.
<u>□ DocumentedAttributes</u>	Gets or sets which attributes documented.
<u>DocumentEmptyNamespaces</u>	Gets or sets the DocumentP property.
<u>™DocumentInternals</u>	Gets or sets the DocumentIr property.
<u>™DocumentPrivates</u>	Gets or sets the DocumentP property.
<u>□DocumentProtected</u>	Gets or sets the DocumentP property.
DocumentProtectedInternalAsProtected	Gets or sets the DocumentProtectedInternal/property.
<u>EditorBrowsableFilter</u>	Specifies the level of filtering EditorBrowsable attribute.
<u> </u>	Load external xml files?

<u>IncludeAssemblyVersion</u>	Gets or sets the IncludeAssemblyVersion pro
<u>■ ReferencesPath</u>	Gets or sets the base director resolve directory and assem references.
<u>ShowMissingParams</u>	Gets or sets the ShowMissin property.
[™] ShowMissingRemarks	Gets or sets the ShowMissin property.
<u>ShowMissingReturns</u>	Gets or sets the ShowMissin property.
<u>ShowMissingSummaries</u>	Gets or sets the ShowMissingSummaries pro
<u>ShowMissingValues</u>	Gets or sets the ShowMissin property.
<u>ShowTypeIdInAttributes</u>	Gets or sets whether or not t TypeId property in attributes.
**SkipNamespacesWithoutSummaries	Gets or sets the SkipNamespacesWithoutSur property.
<u>■UseNamespaceDocSummaries</u>	Gets or sets the UseNamespaceDocSummar property.
<u>□ UseNDocXmlFile</u>	Gets or sets the UseNDocXr property.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright $\ @$ 2000-2003 the NDoc Team

BaseDocumenterConfig.AutoDocumentConstructors Property

Gets or sets the AutoDocumentConstructors property.

public bool AutoDocumentConstructors {get; see

Remarks

If this is true, default constructors without /doc summary comments will be automatically documented.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.AutoPropertyBackerSummaries Property

Gets or sets the AutoPropertyBackerSummaries property.

public bool AutoPropertyBackerSummaries {get;

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.CopyrightHref Property

Gets or sets the CopyrightHref property.

```
public string CopyrightHref {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.CopyrightText Property

Gets or sets the CopyrightText property.

```
public string CopyrightText {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentAttributes Property

Gets or sets whether or not to document the attributes.

```
public bool DocumentAttributes {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentedAttributes Property

Gets or sets which attributes should be documented.

public string DocumentedAttributes {get; set;

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentEmptyNamespaces Property

Gets or sets the DocumentPrivates property.

public bool DocumentEmptyNamespaces {get; set

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentInternals Property

Gets or sets the DocumentInternals property.

```
public bool DocumentInternals {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentPrivates Property

Gets or sets the DocumentPrivates property.

```
public bool DocumentPrivates {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentProtected Property

Gets or sets the DocumentProtected property.

```
public bool DocumentProtected {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.DocumentProtectedInternalAsProt Property

Gets or sets the DocumentProtectedInternalAsProtected property.

public boot DocumentProtectedInternalAsProtectedInt

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.EditorBrowsableFilter Property

Specifies the level of filtering on the EditorBrowsable attribute.

public EditorBrowsableFilterLevel EditorBrows

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.GetExternalSummaries Property

Load external xml files?

```
public bool GetExternalSummaries {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.IncludeAssemblyVersion Property

Gets or sets the IncludeAssemblyVersion property.

public bool IncludeAssemblyVersion {get; set;

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ReferencesPath Property

Gets or sets the base directory used to resolve directory and assembly references.

```
public string ReferencesPath {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowMissingParams Property

Gets or sets the ShowMissingParams property.

public bool ShowMissingParams {get; set;}

Remarks

If this is true, all members without /doc summary comments will contain the phrase "Missing Documentation" in the generated documentation.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowMissingRemarks Property

Gets or sets the ShowMissingRemarks property.

```
public bool ShowMissingRemarks {get; set;}
```

Remarks

If this is true, all members without /doc summary comments will contain the phrase "Missing Documentation" in the generated documentation.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowMissingReturns Property

Gets or sets the ShowMissingReturns property.

public bool ShowMissingReturns {get; set;}

Remarks

If this is true, all members without /doc summary comments will contain the phrase "Missing Documentation" in the generated documentation.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowMissingSummaries Property

Gets or sets the ShowMissingSummaries property.

public bool ShowMissingSummaries {get; set;}

Remarks

If this is true, all members without /doc summary comments will contain the phrase "Missing Documentation" in the generated documentation.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowMissingValues Property

Gets or sets the ShowMissingValues property.

```
public bool ShowMissingValues {get; set;}
```

Remarks

If this is true, all members without /doc summary comments will contain the phrase "Missing Documentation" in the generated documentation.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.ShowTypeIdInAttributes Property

Gets or sets whether or not to show the TypeId property in attributes.

public bool ShowTypeIdInAttributes {get; set;

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.SkipNamespacesWithoutSummarie Property

Gets or sets the SkipNamespacesWithoutSummaries property.

public bool SkipNamespacesWithoutSummaries {

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.UseNamespaceDocSummaries Property

Gets or sets the UseNamespaceDocSummaries property.

public bool UseNamespaceDocSummaries {get; se

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.UseNDocXmlFile Property

Gets or sets the UseNDocXmlFile property.

```
public string UseNDocXmlFile {get; set;}
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig Methods

The methods of the **BaseDocumenterConfig** class are listed below. For a complete list of **BaseDocumenterConfig** class members, see the <u>BaseDocumenterConfig Members</u> topic.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
● GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
<u> </u>	Gets a list of property names.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
Read	Reads the previously serialized state of the documenter into memory.
** SetProject	Associates this documenter with a project;
<u>*••SetValue</u>	Sets the value of a property.
◆ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
••• Write	Writes the current state of the documenter to the specified XmlWrtier.

Protected Instance Methods

₱Finalize (inherited from)	Allows an Object to attempt to
Object)	free resources and perform
	other cleanup operations before the Object is reclaimed by

	garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
* SetDirty	Sets the IsDirty property on the project if any is set.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright $\ @$ 2000-2003 the NDoc Team

BaseDocumenterConfig.GetProperties Method

Gets a list of property names.

```
public <u>IEnumerable</u> GetProperties();
```

Implements

<u>IDocumenterConfig.GetProperties</u>

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.Read Method

Reads the previously serialized state of the documenter into memory.

```
public void Read(
   XmlReader reader
);
```

Parameters

reader

An XmlReader positioned on a documenter element.

Implements

IDocumenterConfig.Read

Remarks

This method uses reflection to set all of the public properties in the documenter.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.SetDirty Method

Sets the IsDirty property on the project if any is set.

```
protected void SetDirty();
```

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.SetProject Method

Associates this documenter with a project;

```
public void SetProject(
    Project project
);
```

Implements

<u>IDocumenterConfig.SetProject</u>

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.SetValue Method

Sets the value of a property.

```
public void SetValue(
   string name,
   string value
);
```

Implements

IDocumenterConfig.SetValue

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

BaseDocumenterConfig.Write Method

Writes the current state of the documenter to the specified XmlWrtier.

```
public void Write(
   XmlWriter writer
);
```

Parameters

writer

An XmlWriter.

Implements

IDocumenterConfig.Write

Remarks

This method uses reflection to serialize all of the public properties in the documenter.

See Also

BaseDocumenterConfig Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

CouldNotLoadAllAssembliesException Class

This exception is thrown when one or more assemblies can not be loaded.

For a list of all members of this type, see <u>CouldNotLoadAllAssembliesException Members</u>.

System.Object Exception
ApplicationException

CouldNotLoadAllAssembliesException

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>CouldNotLoadAllAssembliesException Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

${\bf CouldNotLoadAllAssemblies Exception\ Members}$

<u>CouldNotLoadAllAssembliesException overview</u>

Public Instance Constructors

CouldNotLoadAllAssembliesException	Overloaded. Initializes a new i of the CouldNotLoadAllAssembliesE class.
------------------------------------	--

Public Instance Properties

<u>►HelpLink</u> (inherited from Exception)	Gets or sets a link to the help file associated with this exception.
InnerException (inherited from Exception)	Gets the Exception instance that caused the current exception.
<u>Message</u> (inherited from Exception)	Gets a message that describes the current exception.
Source (inherited from Exception)	Gets or sets the name of the application or the object that causes the error.
StackTrace (inherited from Exception)	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
<u>TargetSite</u> (inherited from Exception)	Gets the method that throws the current exception.

Public Instance Methods

	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
GetBaseException (inherited	When overridden in a derived

from Exception)	class, returns the <u>Exception</u> that is the root cause of one or more subsequent exceptions.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetObjectData (inherited from Exception)	When overridden in a derived class, sets the <u>SerializationInfo</u> with information about the exception.
GetType (inherited from Object)	Gets the Type of the current instance.
<u>ToString</u> (inherited from Exception)	Creates and returns a string representation of the current exception.

Protected Instance Constructors

CouldNotLoadAllAssembliesException	Overloaded. Initializes a new i of the CouldNotLoadAllAssembliesE class.
------------------------------------	--

Protected Instance Properties

₱ HResult (inherited from	Gets or sets HRESULT, a coded
Exception)	numerical value that is assigned
	to a specific exception.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.

MemberwiseClone (inherited from Object)

Creates a shallow copy of the current Object.

See Also

<u>CouldNotLoadAllAssembliesException Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

CouldNotLoadAllAssembliesException Constructor

Initializes a new instance of the CouldNotLoadAllAssembliesException class.

Overload List

Initializes a new instance of the CouldNotLoadAllAssembliesException class.

public CouldNotLoadAllAssembliesException();

Missing <summary> documentation for

M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

protected CouldNotLoadAllAssembliesException(SerializationInfo,Si

Missing <summary> documentation for

M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

public CouldNotLoadAllAssembliesException(string);

Missing <summary> documentation for

M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

public CouldNotLoadAllAssembliesException(string,Exception);

See Also

<u>CouldNotLoadAllAssembliesException Class</u> | <u>NDoc.Core</u> <u>Namespace</u>

Copyright © 2000-2003 the NDoc Team

CouldNotLoadAllAssembliesException Constructor ()

Initializes a new instance of the CouldNotLoadAllAssembliesException class.

public CouldNotLoadAllAssembliesException();

See Also

CouldNotLoadAllAssembliesException Class | NDoc.Core Namespace | CouldNotLoadAllAssembliesException Constructor Overload List

Copyright © 2000-2003 the NDoc Team

Generated from assembly NDoc.Core [1.2.1300.31138]

CouldNotLoadAllAssembliesException Constructor (String)

Missing <summary> documentation for M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

```
public CouldNotLoadAllAssembliesException(
    string message
);
```

See Also

CouldNotLoadAllAssembliesException Class | NDoc.Core Namespace | CouldNotLoadAllAssembliesException Constructor Overload List

Copyright © 2000-2003 the NDoc Team

CouldNotLoadAllAssembliesException Constructor (String Exception)

Missing <summary> documentation for M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

```
public CouldNotLoadAllAssembliesException(
    string message,
    Exception inner
);
```

See Also

CouldNotLoadAllAssembliesException Class | NDoc.Core Namespace | CouldNotLoadAllAssembliesException Constructor Overload List

Copyright © 2000-2003 the NDoc Team

Generated from assembly NDoc.Core [1.2.1300.31138]

CouldNotLoadAllAssembliesException Constructor (Seriali StreamingContext)

Missing <summary> documentation for M:NDoc.Core.CouldNotLoadAllAssembliesException.#ctor(System

```
protected CouldNotLoadAllAssembliesException
    SerializationInfo info,
    StreamingContext context
);
```

See Also

CouldNotLoadAllAssembliesException Class | NDoc.Core Namespace | CouldNotLoadAllAssembliesException Constructor Overload List

Copyright © 2000-2003 the NDoc Team

Generated from assembly NDoc.Core [1.2.1300.31138]

DocBuildingEventHandler Delegate

Used by IDocumenter's events to notify doc building progress.

Parameters

sender

The source of the event.

е

An ProgressArgs that contains the event data.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

DocumenterDevelopmentStatus Enumeration

This is used to indicate the development status of each documenter.

public enum DocumenterDevelopmentStatus

Members

Member Name	Description
Alpha	Still in development, not really ready for anyone else to use except to provide feedback.
Beta	Ready for users to try out, with the understanding that bugs are likely.
Stable	Ready for use, or at least as stable as free software gets!

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

DocumenterException Class

Represents the exceptions that are thrown when attempting to build documentation.

For a list of all members of this type, see <u>DocumenterException</u> <u>Members</u>.

System.Object Exception
ApplicationException
DocumenterException

public class DocumenterException :
 ApplicationException

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>DocumenterException Members | NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

DocumenterException Members

<u>DocumenterException overview</u>

Public Instance Constructors

<u>■ DocumenterException</u>	Overloaded. Initializes a new
	instance of the
	DocumenterException class.

Public Instance Properties

HelpLink (inherited from Exception)	Gets or sets a link to the help file associated with this exception.
<u>InnerException</u> (inherited from Exception)	Gets the Exception instance that caused the current exception.
<u>Message</u> (inherited from Exception)	Gets a message that describes the current exception.
Source (inherited from Exception)	Gets or sets the name of the application or the object that causes the error.
<u>StackTrace</u> (inherited from Exception)	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
TargetSite (inherited from Exception)	Gets the method that throws the current exception.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
<u>◆GetBaseException</u> (inherited from Exception)	When overridden in a derived class, returns the Exception that

	is the root cause of one or more subsequent exceptions.
● GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetObjectData (inherited from Exception)	When overridden in a derived class, sets the <u>SerializationInfo</u> with information about the exception.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>NoString</u> (inherited from Exception)	Creates and returns a string representation of the current exception.

Protected Instance Properties

HResult (inherited from Exception)	Gets or sets HRESULT, a coded numerical value that is assigned
	to a specific exception.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>DocumenterException Class | NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

DocumenterException Constructor

Initializes a new instance of the DocumenterException class with the specified message.

Overload List

Initializes a new instance of the DocumenterException class with the specified message.

public DocumenterException(string);

Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the root cause of this exception.

public DocumenterException(string,Exception);

See Also

<u>DocumenterException Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

DocumenterException Constructor (String)

Initializes a new instance of the DocumenterException class with the specified message.

```
public DocumenterException(
   string message
);
```

Parameters

message

The message to display when the exception is thrown.

See Also

<u>DocumenterException Class | NDoc.Core Namespace | DocumenterException Constructor Overload List</u>

Copyright © 2000-2003 the NDoc Team

DocumenterException Constructor (String, Exception)

Initializes a new instance of the Exception class with a specified error message and a reference to the inner exception that is the root cause of this exception.

```
public DocumenterException(
    string message,
    Exception inner
);
```

Parameters

message

The error message that explains the reason for the exception.

inner

An instance of Exception that is the cause of the current Exception. If inner is non-null, then the current Exception is raised in a catch block handling inner.

See Also

<u>DocumenterException Class | NDoc.Core Namespace |</u> <u>DocumenterException Constructor Overload List</u>

Copyright © 2000-2003 the NDoc Team

EditorBrowsableFilterLevel Enumeration

Define the levels of filtering on the EditorBrowsable attribute.

public enum EditorBrowsableFilterLevel

Members

Member Name	Description
Off	No filtering.
HideNever	Hide members flagged with EditorBrowsableState.Never.
HideAdvanced	Hide members flagged with EditorBrowsableState.Never or EditorBrowsableState.Advanced.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources Class

Utilties to help reading and writing embedded resources.

For a list of all members of this type, see <u>EmbeddedResources</u> Members.

System.Object EmbeddedResources

public class EmbeddedResources

Remarks

This is used to access the stylesheets.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

EmbeddedResources Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources Members

EmbeddedResources overview

Public Static Methods

[™] S WriteEmbeddedResource	Writes an embedded resource to disk.
™ S WriteEmbeddedResources	Writes all the embedded resources with the specified prefix to disk.

Public Instance Constructors

<u> </u>	Initializes a new instance of the
Constructor	EmbeddedResources class.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
	garbage concentri

MemberwiseClone (inherited from Object)

Creates a shallow copy of the current Object.

See Also

EmbeddedResources Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources Constructor

Initializes a new instance of the EmbeddedResources class.

public EmbeddedResources();

See Also

EmbeddedResources Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources Methods

The methods of the **EmbeddedResources** class are listed below. For a complete list of **EmbeddedResources** class members, see the <u>EmbeddedResources Members</u> topic.

Public Static Methods

™ S WriteEmbeddedResource	Writes an embedded resource to disk.
™ S WriteEmbeddedResources	Writes all the embedded resources with the specified prefix to disk.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
••GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

EmbeddedResources Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources.WriteEmbeddedResource Method

Writes an embedded resource to disk.

```
public static void WriteEmbeddedResource(
    Assembly assembly,
    string name,
    string directory,
    string filename
);
```

Parameters

assembly

The assembly containing the embedded resource.

name

The name of the embedded resource.

directory

The directory to write the resource to.

filename

The filename of the resource on disk.

See Also

EmbeddedResources Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

EmbeddedResources.WriteEmbeddedResources Method

Writes all the embedded resources with the specified prefix to disk.

```
public static void WriteEmbeddedResources(
    Assembly assembly,
    string prefix,
    string directory
);
```

Parameters

```
assembly
The assembly containing the embedded resources.

prefix
The prefix to search for.

directory
```

The directory to write the resources to.

See Also

EmbeddedResources Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp Class

HTML Help file utilities.

For a list of all members of this type, see

Html/Help Members">Html/Help Members.

System.Object HtmlHelp

public class HtmlHelp

Remarks

This class is used by the MsdnHelp documenter to create the files needed by the HTML Help compiler.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

HtmlHelp Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp Members

HtmlHelp overview

Public Instance Constructors

<u>MtmlHelp Constructor</u>	Initializes a new instance of the
	HtmlHelp class.

Public Instance Properties

<u>□ DefaultTopic</u>	Gets or sets the DefaultTopic property.
<u>□ DirectoryName</u>	Gets the directory name containing the HTML Help files.
<u>■HtmlHelpCompiler</u>	Gets or sets the path to the HTML Help Compiler.
<u>IncludeFavorites</u>	Gets or sets the IncludeFavorites property.
<u>[®]ProjectName</u>	Gets the HTML Help project name.

Public Instance Methods

<u> </u>	Adds a file to the contents file.
**AddFileToProject	Adds a file to the HTML Help project file.
**CloseBookInContents	Closes the last opened "book" in the contents file.
<u> </u>	Closes the contents file.
<u> </u>	Closes the HTML Help project file.
<u> </u>	Compiles the HTML Help project.
Equals (inherited from Object)	Determines whether the

	specified <u>Object</u> is equal to the current <u>Object</u> .
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
■ GetPathToCompiledHtmlFile	Gets the path the the CHM file.
<u> </u>	Gets the path the the HHC file.
<u> </u>	Gets the path the the HHK file.
<u> </u>	Gets the path the the LOG file.
<u> </u>	Gets the path the the HHP file.
	Gets the <u>Type</u> of the current instance.
**OpenBookInContents	Creates a new "book" in the HTML Help contents file.
••OpenContentsFile	Opens a HTML Help contents file for writing.
◆OpenProjectFile	Opens an HTML Help project file for writing.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
WriteEmptyIndexFile	Writes an empty index file.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp Constructor

Initializes a new instance of the HtmlHelp class.

```
public HtmlHelp(
    string directoryName,
    string projectName,
    string defaultTopic,
    string htmlHelpCompiler,
    bool generateTocOnly
);
```

Parameters

directoryName

The directory to write the HTML Help files to.

projectName

The name of the HTML Help project.

defaultTopic

The default topic for the compiled HTML Help file.

htmlHelpCompiler

The path to the HTML Help compiler.

generateTocOnly

When true, HtmlHelp only outputs the HHC file and does not compile the CHM.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp Properties

The properties of the **HtmlHelp** class are listed below. For a complete list of **HtmlHelp** class members, see the <u>HtmlHelp</u> <u>Members</u> topic.

Public Instance Properties

™ DefaultTopic	Gets or sets the DefaultTopic property.
<u>™DirectoryName</u>	Gets the directory name containing the HTML Help files.
<u>™HtmlHelpCompiler</u>	Gets or sets the path to the HTML Help Compiler.
<u>IncludeFavorites</u>	Gets or sets the IncludeFavorites property.
[™] ProjectName	Gets the HTML Help project name.

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.DefaultTopic Property

Gets or sets the DefaultTopic property.

```
public string DefaultTopic {get; set;}
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.DirectoryName Property

Gets the directory name containing the HTML Help files.

```
public string DirectoryName {get;}
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.HtmlHelpCompiler Property

Gets or sets the path to the HTML Help Compiler.

```
public string HtmlHelpCompiler {get; set;}
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.IncludeFavorites Property

Gets or sets the IncludeFavorites property.

```
public bool IncludeFavorites {get; set;}
```

Remarks

Setting this to true will include the "favorites" tab in the compiled HTML Help file.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.ProjectName Property

Gets the HTML Help project name.

```
public string ProjectName {get;}
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp Methods

The methods of the **HtmlHelp** class are listed below. For a complete list of **HtmlHelp** class members, see the <u>HtmlHelp Members</u> topic.

Public Instance Methods

*AddFileToContents	Adds a file to the contents file.
**AddFileToProject	Adds a file to the HTML Help project file.
**CloseBookInContents	Closes the last opened "book" in the contents file.
<u> </u>	Closes the contents file.
<u> </u>	Closes the HTML Help project file.
**CompileProject	Compiles the HTML Help project.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
Cat look Cada (inhavitad fuers	
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
·	particular type, suitable for use in hashing algorithms and data
Object)	particular type, suitable for use in hashing algorithms and data structures like a hash table.
Object) ••GetPathToCompiledHtmlFile	particular type, suitable for use in hashing algorithms and data structures like a hash table. Gets the path the the CHM file.
Object) ••GetPathToCompiledHtmlFile ••GetPathToContentsFile	particular type, suitable for use in hashing algorithms and data structures like a hash table. Gets the path the the CHM file. Gets the path the the HHC file.
Object) GetPathToCompiledHtmlFile GetPathToContentsFile GetPathToIndexFile	particular type, suitable for use in hashing algorithms and data structures like a hash table. Gets the path the the CHM file. Gets the path the the HHC file. Gets the path the the HHK file.
Object) GetPathToCompiledHtmlFile GetPathToContentsFile GetPathToIndexFile GetPathToLogFile	particular type, suitable for use in hashing algorithms and data structures like a hash table. Gets the path the the CHM file. Gets the path the the HHC file. Gets the path the the HHK file. Gets the path the the LOG file.

**OpenContentsFile	Opens a HTML Help contents file for writing.
<u> </u>	Opens an HTML Help project file for writing.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
<u> </u>	Writes an empty index file.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited from Object)	Creates a shallow copy of the current Object.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.AddFileToContents Method

Adds a file to the contents file.

```
public void AddFileToContents(
   string headingName,
   string htmlFilename
);
```

Parameters

headingName

The name as it should appear in the contents.

htmlFilename

The filename for this entry.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.AddFileToProject Method

Adds a file to the HTML Help project file.

```
public void AddFileToProject(
   string filename
);
```

Parameters

filename

The filename to add.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.CloseBookInContents Method

Closes the last opened "book" in the contents file.

```
public void CloseBookInContents();
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.CloseContentsFile Method

Closes the contents file.

```
public void CloseContentsFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.CloseProjectFile Method

Closes the HTML Help project file.

```
public void CloseProjectFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.CompileProject Method

Compiles the HTML Help project.

```
public void CompileProject();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.GetPathToCompiledHtmlFile Method

Gets the path the the CHM file.

public string GetPathToCompiledHtmlFile();

Return Value

The path to the CHM file.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.GetPathToContentsFile Method

Gets the path the the HHC file.

```
public string GetPathToContentsFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.GetPathToIndexFile Method

Gets the path the the HHK file.

```
public string GetPathToIndexFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.GetPathToLogFile Method

Gets the path the the LOG file.

```
public string GetPathToLogFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.GetPathToProjectFile Method

Gets the path the the HHP file.

```
public string GetPathToProjectFile();
```

See Also

<u>HtmlHelp Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

HtmlHelp.OpenBookInContents Method

Creates a new "book" in the HTML Help contents file.

```
public void OpenBookInContents();
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.OpenContentsFile Method

Opens a HTML Help contents file for writing.

```
public void OpenContentsFile(
    string tocName,
    bool isDefault
);
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.OpenProjectFile Method

Opens an HTML Help project file for writing.

```
public void OpenProjectFile();
```

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

HtmlHelp.WriteEmptyIndexFile Method

Writes an empty index file.

Remarks

The HTML Help Compiler will complain if this file doesn't exist.

See Also

HtmlHelp Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

IDocumenter Interface

Represents a class capable of generating documentation from a given assembly and /doc file.

For a list of all members of this type, see **IDocumenter Members**.

public interface IDocumenter

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>IDocumenter Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter Members

IDocumenter overview

Public Instance Properties

™ Config	Gets or sets the documenter's config object.
<u>■DevelopmentStatus</u>	The development status (alpha, beta, stable) of this documenter.
<u>™MainOutputFile</u>	Returns the documenter's main output file path.
[™] Name	Gets the display name for the documenter.

Public Instance Methods

■ •Build	Builds the documentation.
<u>◆CanBuild</u>	Overloaded. Checks if the documentation output file(s) exist.
<u> </u>	Resets the documenter to a clean state.
□ View	Spawns a new process to view the generated documentation.

Public Instance Events

Raised to update the current step's percent complete value.
Raised to update the overall percent complete value and the current step name.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team Generated from assembly NDoc.Core [1.2.1300.31138]

IDocumenter Properties

The properties of the **IDocumenter** interface are listed below. For a complete list of **IDocumenter** interface members, see the <u>IDocumenter Members</u> topic.

Public Instance Properties

<u>Config</u>	Gets or sets the documenter's config object.
<u>™DevelopmentStatus</u>	The development status (alpha, beta, stable) of this documenter.
<u>™MainOutputFile</u>	Returns the documenter's main output file path.
[™] Name	Gets the display name for the documenter.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.Config Property

Gets or sets the documenter's config object.

```
IDocumenterConfig Config {get; set;}
```

Remarks

This can be put inside a PropertyGrid for editing by the user.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.DevelopmentStatus Property

The development status (alpha, beta, stable) of this documenter.

<u>DocumenterDevelopmentStatus</u> **DevelopmentStatus**

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.MainOutputFile Property

Returns the documenter's main output file path.

```
string MainOutputFile {get;}
```

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.Name Property

Gets the display name for the documenter.

```
string Name {get;}
```

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter Methods

The methods of the **IDocumenter** interface are listed below. For a complete list of **IDocumenter** interface members, see the <u>IDocumenter Members</u> topic.

Public Instance Methods

■ •Build	Builds the documentation.
<u>◆CanBuild</u>	Overloaded. Checks if the documentation output file(s) exist.
<u> </u>	Resets the documenter to a clean state.
≅• <mark>View</mark>	Spawns a new process to view the generated documentation.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.Build Method

Builds the documentation.

```
void Build(
   Project project
);
```

Remarks

The compiler does not currently allow namespaces to documented.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.CanBuild Method

Checks if the documentation output file(s) exist.

Overload List

Checks if the documentation output file(s) exist.

string CanBuild(Project);

Checks to make sure the documenter can perform a build.

string CanBuild(Project, bool);

See Also

IDocumenter Interface | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

IDocumenter.CanBuild Method (Project)

Checks if the documentation output file(s) exist.

```
string CanBuild(
    Project project
);
```

Parameters

project

The project that generated the documentation.

Return Value

True if the documentation can be viewed.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u> | IDocumenter.CanBuild Overload List

Copyright © 2000-2003 the NDoc Team

IDocumenter.CanBuild Method (Project, Boolean)

Checks to make sure the documenter can perform a build.

```
string CanBuild(
    Project project,
    bool checkInputOnly
);
```

Parameters

```
project
The project that would be built.

checkInputOnly
When true, don't check for output file locking.
```

Return Value

null if the documenter can build; otherwise a message describing why it can't build

Remarks

This is for people who like to leave their CHMs open.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u> | <u>IDocumenter.CanBuild Overload List</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.Clear Method

Resets the documenter to a clean state.

```
void Clear();
```

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.View Method

Spawns a new process to view the generated documentation.

```
void View();
```

Exceptions

Exception Type	Condition
· ·	Thrown if the main output file does not exist.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter Events

The events of the **IDocumenter** interface are listed below. For a complete list of **IDocumenter** interface members, see the <u>IDocumenter Members</u> topic.

Public Instance Events

[₱] <u>DocBuildingProgress</u>	Raised to update the current step's percent complete value.
[₱] DocBuildingStep	Raised to update the overall percent complete value and the current step name.

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.DocBuildingProgress Event

Raised to update the current step's percent complete value.

event <u>DocBuildingEventHandler</u> <u>DocBuildingProg</u>

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenter.DocBuildingStep Event

Raised to update the overall percent complete value and the current step name.

event <u>DocBuildingEventHandler</u> DocBuildingSter

See Also

<u>IDocumenter Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig Interface

Specifies the methods that are common to all documenter configs.

For a list of all members of this type, see <u>IDocumenterConfig</u> Members.

public interface IDocumenterConfig

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>IDocumenterConfig Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig Members

IDocumenterConfig overview

Public Instance Methods

<u> </u>	Gets a list of property names.
≅ ◆ Read	Reads the previously serialized state of the documenter into memory.
<u> </u>	Associates the config with a project.
<u> </u>	Sets the value of a property.
™ Write	Writes the current state of the documenter to the specified XmlWrtier.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig Methods

The methods of the **IDocumenterConfig** interface are listed below. For a complete list of **IDocumenterConfig** interface members, see the <u>IDocumenterConfig Members</u> topic.

Public Instance Methods

<u> </u>	Gets a list of property names.
≅ • Read	Reads the previously serialized state of the documenter into memory.
<u> </u>	Associates the config with a project.
<u> </u>	Sets the value of a property.
Write	Writes the current state of the documenter to the specified XmlWrtier.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig.GetProperties Method

Gets a list of property names.

```
IEnumerable GetProperties();
```

Return Value

An enumerable list of property names.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig.Read Method

Reads the previously serialized state of the documenter into memory.

```
void Read(
    XmlReader reader
);
```

Parameters

reader

An XmlReader positioned on a documenter element.

Remarks

This method uses reflection to set all of the public properties in the documenter.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig.SetProject Method

Associates the config with a project.

```
void SetProject(
    Project project
);
```

Remarks

Changes to the config will notify the project so that it can be marked as modified and saved.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig.SetValue Method

Sets the value of a property.

```
void SetValue(
   string name,
   string value
);
```

Parameters

```
name
The name of the property.

value
The value of the property.
```

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

IDocumenterConfig.Write Method

Writes the current state of the documenter to the specified XmlWrtier.

```
void Write(
    XmlWriter writer
);
```

Parameters

writer

An XmlWriter.

Remarks

This method uses reflection to serialize all of the public properties in the documenter.

See Also

<u>IDocumenterConfig Interface</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

ProgressArgs Class

Custom event arguments' class used by DocBuildingEventHandler.

For a list of all members of this type, see ProgressArgs Members.

System.Object EventArgs

ProgressArgs

public class ProgressArgs : EventArgs

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

ProgressArgs Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

ProgressArgs Members

ProgressArgs overview

Public Instance Constructors

ProgressArgs Constructor	ProgressArgs default
	constructor.

Public Instance Properties

<u>Progress</u>	Gets the percentage value.
<u>Status</u>	Gets the current work label.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
••GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

ProgressArgs Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

ProgressArgs Constructor

ProgressArgs default constructor.

```
public ProgressArgs(
    int progress,
    string status
);
```

Parameters

progress

Percentage value for a progress bar.

status

The label describing the current work beeing done.

See Also

ProgressArgs Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

ProgressArgs Properties

The properties of the **ProgressArgs** class are listed below. For a complete list of **ProgressArgs** class members, see the <u>ProgressArgs Members</u> topic.

Public Instance Properties

<u>Progress</u>	Gets the percentage value.
<u>Status</u>	Gets the current work label.

See Also

ProgressArgs Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

ProgressArgs.Progress Property

Gets the percentage value.

Property Value

A number between 0 and 100 corresponding to the percentage of work completed.

See Also

ProgressArgs Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

ProgressArgs.Status Property

Gets the current work label.

Property Value

A short description of the current work beeing done.

See Also

ProgressArgs Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Class

Represents an NDoc project.

For a list of all members of this type, see **Project Members**.

System.Object Project

public class Project

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

Project Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Members

Project overview

Public Static Methods

S FindDocumenters	Searches the module directory
	for assemblies containing
	classes the implement
	IDocumenter.

Public Instance Constructors

Project Constructor	Initializes a new instance of the
	Project class.

Public Instance Properties

**AssemblySlashDocCount	Gets the number of assembly/doc pairs in the project.
<u>™Documenters</u>	Gets or sets the Documenters property.
<u>IsDirty</u>	Gets the IsDirty property.
<u>™NamespaceCount</u>	The number of namespaces in the project.

Public Instance Methods

**AddAssemblySlashDoc	Adds an assembly/doc pair to the project.
* <u>Clear</u>	Clears the project.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
FindAssemblySlashDocByName	Returns the index of the assembly/doc based on an

	assembly name.
→ <u>GetAssemblySlashDoc</u>	Gets an assembly/doc pair.
GetAssemblySlashDocs	Gets an enumerable list of assembly/doc pairs.
**GetDocumenter	Retrieves a documenter by name.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
**GetNamespaces	Gets an enumerable list of namespace names.
◆ GetNamespaceSummary	Gets the summary for a namespace.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
◆ <u>Read</u>	Reads an NDoc project file.
**ReadNamespaceSummaries	Loads namespace summaries from an XML document.
**RemoveAssemblySlashDoc	Removes an assembly/doc pair from the project.
SetNamespaceSummary	Sets a namespace summary.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
₩rite	Writes an NDoc project file.

Public Instance Events

Modified	Raised by projects when they're dirty state changes from false to
	true.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited from Object)	Creates a shallow copy of the current Object.

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Constructor

Initializes a new instance of the Project class.

```
public Project();
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Properties

The properties of the **Project** class are listed below. For a complete list of **Project** class members, see the <u>Project Members</u> topic.

Public Instance Properties

**AssemblySlashDocCount	Gets the number of assembly/doc pairs in the project.
<u>■Documenters</u>	Gets or sets the Documenters property.
™ IsDirty	Gets the IsDirty property.
<u>™NamespaceCount</u>	The number of namespaces in the project.

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.AssemblySlashDocCount Property

Gets the number of assembly/doc pairs in the project.

public int AssemblySlashDocCount {get;}

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.Documenters Property

Gets or sets the Documenters property.

public System.Collections.ArrayList Documente

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.IsDirty Property

Gets the IsDirty property.

```
public bool IsDirty {get; set;}
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.NamespaceCount Property

The number of namespaces in the project.

```
public int NamespaceCount {get;}
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Methods

The methods of the **Project** class are listed below. For a complete list of **Project** class members, see the <u>Project Members</u> topic.

Public Static Methods

S FindDocumenters	Searches the module directory
	for assemblies containing
	classes the implement
	IDocumenter.

Public Instance Methods

*AddAssemblySlashDoc	Adds an assembly/doc pair to the project.
<u> </u>	Clears the project.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
FindAssemblySlashDocByName	Returns the index of the assembly/doc based on an assembly name.
→ GetAssemblySlashDoc	Gets an assembly/doc pair.
GetAssemblySlashDocs	Gets an enumerable list of assembly/doc pairs.
<u> </u>	Retrieves a documenter by name.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
<u> </u>	Gets an enumerable list of namespace names.
⇔ GetNamespaceSummary	Gets the summary for a

	namespace.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u> </u>	Reads an NDoc project file.
**ReadNamespaceSummaries	Loads namespace summaries from an XML document.
**RemoveAssemblySlashDoc	Removes an assembly/doc pair from the project.
◆SetNamespaceSummary	Sets a namespace summary.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
* <u>Write</u>	Writes an NDoc project file.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

Project Class | NDoc.Core Namespace

Copyright $\ @ \ 2000-2003$ the NDoc Team

Project.AddAssemblySlashDoc Method

Adds an assembly/doc pair to the project.

```
public bool AddAssemblySlashDoc(
    AssemblySlashDoc assemblySlashDoc
);
```

Return Value

bool - true for doc added, false or exception otherwise

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.Clear Method

Clears the project.

```
public void Clear();
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.FindAssemblySlashDocByName Method

Returns the index of the assembly/doc based on an assembly name.

```
public int FindAssemblySlashDocByName(
    string assemblyName
);
```

Parameters

assemblyName
The assembly to search for.

Return Value

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.FindDocumenters Method

Searches the module directory for assemblies containing classes the implement IDocumenter.

public static ArrayList FindDocumenters();

Return Value

An ArrayList containing new instances of all the found documenters.

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.GetAssemblySlashDoc Method

Gets an assembly/doc pair.

```
public AssemblySlashDoc GetAssemblySlashDoc(
   int index
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.GetAssemblySlashDocs Method

Gets an enumerable list of assembly/doc pairs.

```
public <u>IEnumerable</u> GetAssemblySlashDocs();
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.GetDocumenter Method

Retrieves a documenter by name.

```
public IDocumenter GetDocumenter(
   string name
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.GetNamespaces Method

Gets an enumerable list of namespace names.

```
public <u>IEnumerable</u> GetNamespaces();
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.GetNamespaceSummary Method

Gets the summary for a namespace.

```
public string GetNamespaceSummary(
   string namespaceName
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.Read Method

Reads an NDoc project file.

```
public void Read(
   string filename
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.ReadNamespaceSummaries Method

Loads namespace summaries from an XML document.

```
public void ReadNamespaceSummaries(
    XmlReader reader
);
```

Parameters

reader

An open XmlReader positioned before the namespace elements.

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.RemoveAssemblySlashDoc Method

Removes an assembly/doc pair from the project.

```
public void RemoveAssemblySlashDoc(
   int index
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.SetNamespaceSummary Method

Sets a namespace summary.

```
public void SetNamespaceSummary(
   string namespaceName,
   string summary
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.Write Method

Writes an NDoc project file.

```
public void Write(
   string filename
);
```

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project Events

The events of the **Project** class are listed below. For a complete list of **Project** class members, see the <u>Project Members</u> topic.

Public Instance Events

Raised by projects when they're dirty state changes from false to
true.

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.Modified Event

Raised by projects when they're dirty state changes from false to true.

public event ProjectModifiedEventHandler Mod:

See Also

Project Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

Project.AssemblyAlreadyExistsException Class

A custom exception to detect if a duplicate assembly is beeing added.

For a list of all members of this type, see Project.AssemblyAlreadyExistsException Members.

System.Object Exception
ApplicationException

Project.AssemblyAlreadyExistsException

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>Project.AssemblyAlreadyExistsException Members | NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

Project.AssemblyAlreadyExistsException Members

Project.AssemblyAlreadyExistsException overview

Public Instance Constructors

Project.AssemblyAlreadyExistsException	Initializes a new instance of AssemblyAlreadyExistsExc class with a specified error message.
--	--

Public Instance Properties

HelpLink (inherited from Exception)	Gets or sets a link to the help file associated with this exception.
<u>InnerException</u> (inherited from Exception)	Gets the Exception instance that caused the current exception.
<u>Message</u> (inherited from Exception)	Gets a message that describes the current exception.
Source (inherited from Exception)	Gets or sets the name of the application or the object that causes the error.
StackTrace (inherited from Exception)	Gets a string representation of the frames on the call stack at the time the current exception was thrown.
<u>TargetSite</u> (inherited from Exception)	Gets the method that throws the current exception.

Public Instance Methods

	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
GetBaseException (inherited	When overridden in a derived

from Exception)	class, returns the <u>Exception</u> that is the root cause of one or more subsequent exceptions.
● GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetObjectData (inherited from Exception)	When overridden in a derived class, sets the <u>SerializationInfo</u> with information about the exception.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>Nostring</u> (inherited from Exception)	Creates and returns a string representation of the current exception.

Protected Instance Properties

	Gets or sets HRESULT, a coded
Exception)	numerical value that is assigned
	to a specific exception.

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>Project.AssemblyAlreadyExistsException Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team
Generated from assembly NDoc.Core [1.2.1300.31138]

Project.AssemblyAlreadyExistsException Constructor

Initializes a new instance of the AssemblyAlreadyExistsException class with a specified error message.

```
public Project.AssemblyAlreadyExistsException
    string message
);
```

See Also

<u>Project.AssemblyAlreadyExistsException Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

ProjectModifiedEventHandler Delegate

Handles ProjectModified events.

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit Class

This is a dialog box to prompt for a name.

For a list of all members of this type, see <u>SimpleEdit Members</u>.

```
System.Object MarshalByRefObject
Component
Control
ScrollableControl
ContainerControl
Form
SimpleEdit
```

```
public class SimpleEdit : Form
```

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

SimpleEdit Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit Members

SimpleEdit overview

Public Instance Constructors

SimpleEdit Constructor	Creates a SimpleEdit form.
-------------------------------	----------------------------

Public Instance Fields

Value	Contains the value of the text
	box.

Public Instance Properties

AcceptButton (inherited from Form)	Gets or sets the button on the form that is clicked when the user presses the ENTER key.
**AccessibilityObject (inherited from Control)	Gets the AccessibleObject assigned to the control.
AccessibleDefaultActionDescription (inherited from Control)	Gets or sets the default action description of the control for use by accessibility client applications.
**AccessibleDescription (inherited from Control)	Gets or sets the description of the control used by accessibility client applications.
Control)	Gets or sets the name of the control used by accessibility client applications.
<u>AccessibleRole</u> (inherited from Control)	Gets or sets the accessible role of the control
	Gets or sets the active control on the container

	control.
Form)	Gets the currently active multiple document interface (MDI) child window.
	Gets or sets a value indicating whether the control can accept data that the user drags onto it.
**MilowTransparency (inherited from Form)	
Anchor (inherited from Control)	Gets or sets which edges of the control are anchored to the edges of its container.
AutoScale (inherited from Form)	Gets or sets a value indicating whether the form adjusts its size to fit the height of the font used on the form and scales its controls.
from Form)	Gets or sets the base size used for autoscaling of the form.
AutoScroll (inherited from Form)	Gets or sets a value indicating whether the form enables autoscrolling.
<u>AutoScrollMargin</u> (inherited from ScrollableControl)	Gets or sets the size of the auto-scroll margin.
AutoScrollMinSize (inherited from ScrollableControl)	Gets or sets the minimum size of the auto-scroll.
AutoScrollPosition (inherited from ScrollableControl)	Gets or sets the location of the auto-scroll position.
[™] BackColor (inherited from Form)	
	Gets or sets the background image displayed in the

	control.
BindingContext (inherited from ContainerControl)	
■Bottom (inherited from Control)	Gets the distance between the bottom edge of the control and the top edge of its container's client area.
Bounds (inherited from Control)	Gets or sets the size and location of the control including its nonclient elements.
	Gets or sets the button control that is clicked when the user presses the ESC key.
CanFocus (inherited from Control)	Gets a value indicating whether the control can receive focus.
<u>CanSelect</u> (inherited from Control)	Gets a value indicating whether the control can be selected.
<u>Capture</u> (inherited from Control)	Gets or sets a value indicating whether the control has captured the mouse.
CausesValidation (inherited from Control)	Gets or sets a value indicating whether the control causes validation to be performed on any controls that require validation when it receives focus.
Control)	Gets the rectangle that represents the client area of the control.
<u>ClientSize</u> (inherited from Form)	Gets or sets the size of the

	client area of the form.
CompanyName (inherited from Control)	Gets the name of the company or creator of the application containing the control.
Container (inherited from Component)	Gets the <u>IContainer</u> that contains the <u>Component</u> .
ContainsFocus (inherited from Control)	Gets a value indicating whether the control, or one of its child controls, currently has the input focus.
ContextMenu (inherited from Control)	Gets or sets the shortcut menu associated with the control.
<u>ControlBox</u> (inherited from Form)	Gets or sets a value indicating whether a control box is displayed in the caption bar of the form.
<u>Controls</u> (inherited from Control)	Gets the collection of controls contained within the control.
<u>Created</u> (inherited from Control)	Gets a value indicating whether the control has been created.
<u>Cursor</u> (inherited from Control)	Gets or sets the cursor that is displayed when the mouse pointer is over the control.
<u>□ DataBindings</u> (inherited from Control)	Gets the data bindings for the control.
	Gets or sets the size and location of the form on the Windows desktop.
<u>□DesktopLocation</u> (inherited from	Gets or sets the location of

Form)	the form on the Windows desktop.
<u>DialogResult</u> (inherited from Form)	Gets or sets the dialog result for the form.
<u>□ DisplayRectangle</u> (inherited from ScrollableControl)	
□ Disposing (inherited from Control)	Gets a value indicating whether the control is in the process of being disposed of.
[™] Dock (inherited from Control)	Gets or sets which edge of the parent container a control is docked to.
<u>DockPadding</u> (inherited from ScrollableControl)	Gets the dock padding settings for all edges of the control.
<u>Enabled</u> (inherited from Control)	Gets or sets a value indicating whether the control can respond to user interaction.
[™] Focused (inherited from Control)	Gets a value indicating whether the control has input focus.
Font (inherited from Control)	Gets or sets the font of the text displayed by the control.
<u>ForeColor</u> (inherited from Control)	Gets or sets the foreground color of the control.
FormBorderStyle (inherited from Form)	Gets or sets the border style of the form.
<u>Handle</u> (inherited from Control)	Gets the window handle that the control is bound to.
HasChildren (inherited from Control)	Gets a value indicating whether the control contains one or more child controls.

Height (inherited from Control)	Gets or sets the height of the control.
HelpButton (inherited from Form)	Gets or sets a value indicating whether a Help button should be displayed in the caption box of the form.
<u>Icon</u> (inherited from Form)	Gets or sets the icon for the form.
<u>ImeMode</u> (inherited from Control)	Gets or sets the Input Method Editor (IME) mode of the control.
InvokeRequired (inherited from Control)	Gets a value indicating whether the caller must call an invoke method when making method calls to the control because the caller is on a different thread than the one the control was created on.
■ IsAccessible (inherited from Control)	Gets or sets a value indicating whether the control is visible to accessibility applications.
	Gets a value indicating whether the control has been disposed of.
	Gets a value indicating whether the control has a handle associated with it.
<u>IsMdiChild</u> (inherited from Form)	Gets a value indicating whether the form is a multiple document interface (MDI) child form.
<u>IsMdiContainer</u> (inherited from	Gets or sets a value

Form)	indicating whether the form is a container for multiple document interface (MDI) child forms.
IsRestrictedWindow (inherited from Form)	
Form)	Gets or sets a value indicating whether the form will receive key events before the event is passed to the control that has focus.
<u>Left</u> (inherited from Control)	Gets or sets the x-coordinate of a control's left edge in pixels.
Location (inherited from Control)	Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
MaximizeBox (inherited from Form) Form Fo	Gets or sets a value indicating whether the maximize button is displayed in the caption bar of the form.
MaximumSize (inherited from Form)	Gets the maximum size the form can be resized to.
MdiChildren (inherited from Form)	Gets an array of forms that represent the multiple document interface (MDI) child forms that are parented to this form.
MdiParent (inherited from Form)	Gets or sets the current multiple document interface (MDI) parent form of this form.
[™] Menu (inherited from Form)	Gets or sets the MainMenu

MergedMenu (inherited from	that is displayed in the form. Gets the merged menu for
Form) MinimizeBox (inherited from Form)	the form. Gets or sets a value indicating whether the minimize button is displayed in the caption bar of the form.
MinimumSize (inherited from Form)	Gets or sets the minimum size the form can be resized to.
[™] Modal (inherited from Form)	Gets a value indicating whether this form is displayed modally.
<u>Name</u> (inherited from Control)	Gets or sets the name of the control.
[™] Opacity (inherited from Form)	Gets or sets the opacity level of the form.
[™] OwnedForms (inherited from Form)	Gets an array of Form objects that represent all forms that are owned by this form.
<u>Owner</u> (inherited from Form)	Gets or sets the form that owns this form.
Parent (inherited from Control)	Gets or sets the parent container of the control.
ParentForm (inherited from ContainerControl)	Gets the form that the container control is assigned to.
<u>ProductName</u> (inherited from Control)	Gets the product name of the assembly containing the control.
ProductVersion (inherited from Control)	Gets the version of the assembly containing the

displayed in the Windows taskbar. Gets or sets the site of the control. Gets or sets the size of the form. Gets or sets the size of the form. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.		control.
region associated with the control. Gets the distance between the right edge of the control and the left edge of its container. Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts. ShowInTaskbar (inherited from Form) Gets or sets a value indicating whether the form is displayed in the Windows taskbar. Gets or sets the site of the control. Gets or sets the size of the form. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	·	whether the control is currently re-creating its
the right edge of the control and the left edge of its container. Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts. ShowInTaskbar (inherited from Form) Gets or sets a value indicating whether the form is displayed in the Windows taskbar. Gets or sets the site of the control. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	Region (inherited from Control)	region associated with the
indicating whether control's elements are aligned to support locales using right-to-left fonts. Gets or sets a value indicating whether the form is displayed in the Windows taskbar. Gets or sets the site of the control. Gets or sets the site of the control. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	□ Right (inherited from Control)	the right edge of the control and the left edge of its
indicating whether the form is displayed in the Windows taskbar. Site (inherited from Control) Gets or sets the site of the control. Gets or sets the size of the form. Gets or sets the size of the form. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.		indicating whether control's elements are aligned to support locales using right-
control. Gets or sets the size of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	· ·	indicating whether the form is displayed in the Windows
form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the style of the size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	Site (inherited from Control)	
size grip to display in the lower-right corner of the form. Gets or sets the starting position of the form at run time.	Size (inherited from Form)	
Form) position of the form at run time.		size grip to display in the lower-right corner of the
	•	position of the form at run
^{control of the control of the con}	<u>™TabIndex</u> (inherited from Form)	

TabStop (inherited from Control)	Gets or sets a value indicating whether the user can give the focus to this control using the TAB key.
<u>Tag</u> (inherited from Control)	Gets or sets the object that contains data about the control.
Text (inherited from Control)	Gets or sets the text associated with this control.
<u>™Top</u> (inherited from Control)	Gets or sets the y-coordinate of the control's top edge in pixels.
TopLevel (inherited from Form)	Gets or sets a value indicating whether to display the form as a top-level window.
TopLevelControl (inherited from Control)	Gets the parent control that is not parented by another Windows Forms control. Typically, this is the outermost Form that the control is contained in.
TopMost (inherited from Form)	Gets or sets a value indicating whether the form should be displayed as the top-most form of your application.
TransparencyKey (inherited from Form)	Gets or sets the color that will represent transparent areas of the form.
	Gets or sets a value indicating whether the control is displayed.
<u>™Width</u> (inherited from Control)	Gets or sets the width of the

	control.
<u>WindowState</u> (inherited from Form)	Gets or sets the form's window state.
<u>™WindowTarget</u> (inherited from Control)	

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
**BringToFront (inherited from Control)	Brings the control to the front of the z-order.
<u>○••Close</u> (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
**CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
**CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy

	used to communicate with a remote object.
Dispose (inherited from Component)	Overloaded. Releases all resources used by the Component.
<u>DoDragDrop</u> (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
*•GetChildAtPoint (inherited from Control)	Retrieves the child control that is located at the specified coordinates.
GetContainerControl (inherited from Control)	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
For MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
GetNextControl (inherited from Control)	Retrieves the next control forward or back in the tab order

	of child controls.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
Hide (inherited from Control)	Conceals the control from the user.
InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
Invoke (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
►• LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
PointToClient (inherited from Control)	Computes the location of the specified screen point into client coordinates.
PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
PreProcessMessage	Preprocesses input messages

(inherited from Control)	within the message loop before they are dispatched.
**RectangleToClient (inherited from Control)	Computes the size and location of the specified screen rectangle in client coordinates.
**RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
****RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.

Scale (inherited from Control)	Overloaded. Scales the control and any child controls to the specified ratio.
**ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>ToString</u> (inherited from Form)	
■ Update (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
<u>Validate</u> (inherited from	Validates the last invalidated

ContainerControl)	control and its ancestors up
	through, but not including, the
	current control.

Public Instance Events

Activated (inherited from Form)	Occurs when the form is activated in code or by the user.
BackColorChanged (inherited from Control)	Occurs when the value of the BackColor property changes.
BackgroundImageChanged (inherited from Control)	Occurs when the value of the BackgroundImage property changes.
BindingContextChanged (inherited from Control)	Occurs when the value of the BindingContext property changes.
CausesValidationChanged (inherited from Control)	Occurs when the value of the CausesValidation property changes.
ChangeUICues (inherited from Control)	Occurs when the focus or keyboard user interface (UI) cues change.
Click (inherited from Control)	Occurs when the control is clicked.
Closed (inherited from Form)	Occurs when the form is closed.
Closing (inherited from Form)	Occurs when the form is closing.
ContextMenuChanged (inherited from Control)	Occurs when the value of the ContextMenu property changes.
ControlAdded (inherited from Control)	Occurs when a new control is added to the <u>ControlCollection</u> .
ControlRemoved (inherited from Control)	Occurs when a control is removed from the ControlCollection .

CursorChanged (inherited from Control)	Occurs when the value of the Cursor property changes.
Deactivate (inherited from Form)	Occurs when the form loses focus and is not the active form.
Disposed (inherited from Component)	Adds an event handler to listen to the <u>Disposed</u> event on the component.
DockChanged (inherited from Control)	Occurs when the value of the Dock property changes.
DoubleClick (inherited from Control)	Occurs when the control is double-clicked.
DragDrop (inherited from Control)	Occurs when a drag-and-drop operation is completed.
<pre> DragEnter (inherited from Control) </pre>	Occurs when an object is dragged into the control's bounds.
DragLeave (inherited from Control)	Occurs when an object is dragged out of the control's bounds.
DragOver (inherited from Control)	Occurs when an object is dragged over the control's bounds.
EnabledChanged (inherited from Control)	Occurs when the <u>Enabled</u> property value has changed.
Fenter (inherited from Control)	Occurs when the control is entered.
FontChanged (inherited from Control)	Occurs when the <u>Font</u> property value changes.
ForeColorChanged (inherited from Control)	Occurs when the <u>ForeColor</u> property value changes.
GiveFeedback (inherited from Control)	Occurs during a drag operation.

GotFocus (inherited from Control)	Occurs when the control receives focus.
<u>HandleCreated</u> (inherited from Control)	Occurs when a handle is created for the control.
F HandleDestroyed (inherited from Control)	Occurs when the control's handle is in the process of being destroyed.
HelpRequested (inherited from Control)	Occurs when the user requests help for a control.
ImeModeChanged (inherited from Control)	Occurs when the <u>ImeMode</u> property has changed.
InputLanguageChanged (inherited from Form)	Occurs after the input language of the form has changed.
InputLanguageChanging (inherited from Form)	Occurs when the user attempts to change the input language for the form.
Invalidated (inherited from Control)	Occurs when a control's display requires redrawing.
KeyDown (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyPress (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyUp (inherited from Control)	Occurs when a key is released while the control has focus.
Layout (inherited from Control)	Occurs when a control should reposition its child controls.
Leave (inherited from Control)	Occurs when the input focus leaves the control.
Load (inherited from Form)	Occurs before a form is displayed for the first time.
LocationChanged (inherited from Control)	Occurs when the <u>Location</u> property value has changed.

LostFocus (inherited from Control)	Occurs when the control loses focus.
MaximizedBoundsChanged (inherited from Form)	Occurs when the value of the MaximizedBounds property has changed.
MaximumSizeChanged (inherited from Form)	Occurs when the value of the MaximumSize property has changed.
MdiChildActivate (inherited from Form)	Occurs when a multiple document interface (MDI) child form is activated or closed within an MDI application.
MenuComplete (inherited from Form)	Occurs when the menu of a form loses focus.
MenuStart (inherited from Form)	Occurs when the menu of a form receives focus.
MinimumSizeChanged (inherited from Form)	Occurs when the value of the MinimumSize property has changed.
MouseDown (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is pressed.
MouseEnter (inherited from Control)	Occurs when the mouse pointer enters the control.
MouseHover (inherited from Control)	Occurs when the mouse pointer hovers over the control.
MouseLeave (inherited from Control)	Occurs when the mouse pointer leaves the control.
MouseMove (inherited from Control)	Occurs when the mouse pointer is moved over the control.
MouseUp (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is released.

MouseWheel (inherited from Control)	Occurs when the mouse wheel moves while the control has focus.
Move (inherited from Control)	Occurs when the control is moved.
Paint (inherited from Control)	Occurs when the control is redrawn.
ParentChanged (inherited from Control)	Occurs when the <u>Parent</u> property value changes.
QueryAccessibilityHelp (inherited from Control)	Occurs when <u>AccessibleObject</u> is providing help to accessibility applications.
QueryContinueDrag (inherited from Control)	Occurs during a drag-and-drop operation and allows the drag source to determine whether the drag-and-drop operation should be canceled.
Resize (inherited from Control)	Occurs when the control is resized.
RightToLeftChanged (inherited from Control)	Occurs when the RightToLeft property value changes.
SizeChanged (inherited from Control)	Occurs when the <u>Size</u> property value changes.
StyleChanged (inherited from Control)	Occurs when the control style changes.
SystemColorsChanged (inherited from Control)	Occurs when the system colors change.
TablndexChanged (inherited from Form)	
TabStopChanged (inherited from Control)	Occurs when the <u>TabStop</u> property value changes.
TextChanged (inherited from	Occurs when the <u>Text</u> property

Control)	value changes.
Validated (inherited from Control)	Occurs when the control is finished validating.
Validating (inherited from Control)	Occurs when the control is validating.
VisibleChanged (inherited from Control)	Occurs when the <u>Visible</u> property value changes.

Protected Instance Properties

	Gets the default Input Method Editor (IME) mode supported by the control.
	Gets a value that indicates whether the <u>Component</u> is currently in design mode.
Events (inherited from Component)	Gets the list of event handlers that are attached to this Component .
FontHeight (inherited from Control)	Gets or sets the height of the font of the control.
	Gets or sets a value indicating whether the horizontal scroll bar is visible.
MaximizedBounds (inherited from Form)	Gets and sets the size of the form when it is maximized.
RenderRightToLeft (inherited from Control)	
ResizeRedraw (inherited from	Gets or sets a value indicating

Control)	whether the control redraws itself when resized.
	Gets a value indicating whether the control should display focus rectangles.
ShowKeyboardCues (inherited from Control)	Gets a value indicating whether the control should display keyboard shortcuts.
✓ VScroll (inherited from ScrollableControl)	Gets or sets a value indicating whether the vertical scroll bar is visible.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
<u>ActivateMdiChild</u> (inherited from Form)	
<u>AdjustFormScrollbars</u> (inherited from Form)	
★ApplyAutoScaling (inherited from Form)	
<u>CenterToParent</u> (inherited from Form)	
<u>CenterToScreen</u> (inherited from Form)	
<u>◆CreateAccessibilityInstance</u> (inherited from Control)	Creates a new accessibility object for the control.
<u>◆CreateControlsInstance</u> (inherited from Form)	

DefWndProc (inherited from Form)	
<u>PostroyHandle</u> (inherited from Control)	Destroys the handle associate with the control.
₱ Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resource and performs other cleanup operations before the Component is reclaimed by garbage collection.
GetScrollState (inherited from ScrollableControl)	
	Returns an object that represents a service provided the Component or by its Container.
	Retrieves the value of the specified control style bit for the control.
GetTopLevel (inherited from Control)	Determines if the control is a top-level control.
InitLayout (inherited from Control)	Called after the control has be added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event fo the specified control.
InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event for the specified control.
InvokeOnClick (inherited from Control)	Raises the <u>Click</u> event for the specified control.
InvokePaint (inherited from Control)	Raises the Paint event for the specified control.
₱InvokePaintBackground (inherited)	Raises the PaintBackground

from Control)	event for the specified control.
	Determines if a character is ar input character that the contro recognizes.
	Determines whether the specified key is a regular inputey or a special key that requires preprocessing.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
◆OnActivated (inherited from Form)	Raises the <u>Activated</u> event.
OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChange</u> event.
OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> even
OnCausesValidationChanged (inherited from Control)	Raises the <u>CausesValidationChanged</u> eve
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUICues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
◆OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
◆OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
<u>◆OnContextMenuChanged</u> (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the <u>ControlAdded</u> eve

OnControlRemoved (inherited from ContainerControl)	
OnCreateControl (inherited from Form)	
OnCursorChanged (inherited from Control)	Raises the <u>CursorChanged</u> event.
[♠] OnDeactivate (inherited from Form)	Raises the <u>Deactivate</u> event.
OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> eve
OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event
OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
OnDragLeave (inherited from Control)	Raises the <u>DragLeave</u> event.
OnDragOver (inherited from Control)	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
<u>OnEnter</u> (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
OnForeColorChanged (inherited from Control)	Raises the <u>ForeColorChanged</u> event.
OnGiveFeedback (inherited from Control)	Raises the GiveFeedback eve
	Raises the <u>GotFocus</u> event.

[♣] OnHandleCreated (inherited from Form)	
[→] OnHandleDestroyed (inherited from Form)	
<u>OnHelpRequested</u> (inherited from Control)	Raises the <u>HelpRequested</u> event.
OnlmeModeChanged (inherited from Control)	Raises the <u>ImeModeChanged</u> event.
<u>→OnInputLanguageChanged</u> (inherited from Form)	Raises the InputLanguageChanged event.
OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
OnInvalidated (inherited from Control)	Raises the <u>Invalidated</u> event.
[♣] OnKeyDown (inherited from Control)	Raises the <u>KeyDown</u> event.
OnKeyPress (inherited from Control)	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
[♣] OnLayout (inherited from ScrollableControl)	
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
◆OnLoad (inherited from Form)	Raises the <u>Load</u> event.
[♣] OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
	Raises the <u>LostFocus</u> event.
	Raises the MaximizedBoundsChanged event.
⋄ OnMaximumSizeChanged	Raises the

(inherited from Form)	MaximumSizeChanged event.
OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
	Raises the MenuComplete event.
OnMenuStart (inherited from Form)	Raises the MenuStart event.
OnMinimumSizeChanged (inherited from Form)	Raises the MinimumSizeChanged event.
OnMouseDown (inherited from Control)	Raises the MouseDown event
OnMouseEnter (inherited from Control)	Raises the MouseEnter event
<u>OnMouseHover</u> (inherited from Control)	Raises the <u>MouseHover</u> even
OnMouseLeave (inherited from Control)	Raises the MouseLeave even
<u>OnMouseMove</u> (inherited from Control)	Raises the <u>MouseMove</u> event
	Raises the MouseUp event.
OnMouseWheel (inherited from ScrollableControl)	
◆OnMove (inherited from Control)	Raises the Move event.
OnNotifyMessage (inherited from Control)	Notifies the control of Window messages.
◆OnPaint (inherited from Form)	
[♠] OnPaintBackground (inherited from Control)	Paints the background of the control.
<u>OnParentBackColorChanged</u> (inherited from Control)	Raises the <u>BackColorChange</u> event when the <u>BackColor</u>

	property value of the control's container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event when the BackgroundImage property value of the control's contained changes.
OnParentBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever when the <u>BindingContext</u> property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
<u>OnParentEnabledChanged</u> (inherited from Control)	Raises the <u>EnabledChanged</u> event when the <u>Enabled</u> property value of the control's container changes.
OnParentFontChanged (inherited from Control)	Raises the <u>FontChanged</u> ever when the <u>Font</u> property value the control's container change
OnParentForeColorChanged (inherited from Control)	Raises the ForeColorChange event when the ForeColor property value of the control's container changes.
<u>OnParentRightToLeftChanged</u> (inherited from Control)	Raises the RightToLeftChange event when the RightToLeft property value of the control's container changes.
OnParentVisibleChanged (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> prope value of the control's containe changes.

OnQueryContinueDrag (inherited from Control)	Raises the <u>QueryContinueDrag</u> event.
→ OnResize (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> event
[→] OnStyleChanged (inherited from Form)	
[→] OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
[→] OnTabIndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
[→] OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
[♣] OnTextChanged (inherited from Form)	
	Raises the <u>Validated</u> event.
[♣] OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
₱ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
₱ProcessDialogKey (inherited from Form)	
₱ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate

	control events.
ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
ProcessTabKey (inherited from Form)	
RaiseDragEvent (inherited from Control)	
RaiseKeyEvent (inherited from Control)	
	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
*RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment t the appropriate HorizontalAlignment to support right-to-left text.
*RtlTranslateContent (inherited from Control)	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
	appropriate ContentAlignment support right-to-left text. Converts the specified HorizontalAlignment to the appropriate HorizontalAlignment

	Converts the specified LeftRightAlignment to the appropriate LeftRightAlignment to support right-to-left text.
◆ScaleCore (inherited from Form)	
	Overloaded.
★SetBoundsCore (inherited from Form)	
★SetDisplayRectLocation (inherited from ScrollableControl)	
★SetScrollState (inherited from ScrollableControl)	
★SetStyle (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
	Overloaded. Updates the bounds of the control with the current size and location.
<u>DupdateDefaultButton</u> (inherited from Form)	
<u>DupdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.
<u>WndProc</u> (inherited from Form)	

Protected Internal Instance Methods

<u></u> <u>ProcessKeyMessage</u>	Processes a keyboard
(inherited from Control)	message.

Explicit Interface Implementations



See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit Constructor

Creates a SimpleEdit form.

```
public SimpleEdit();
```

See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit Fields

The fields of the **SimpleEdit** class are listed below. For a complete list of **SimpleEdit** class members, see the <u>SimpleEdit Members</u> topic.

Public Instance Fields

♦ <u>Value</u>	Contains the value of the text
	box.

See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit.Value Field

Contains the value of the text box.

See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit Methods

The methods of the **SimpleEdit** class are listed below. For a complete list of **SimpleEdit** class members, see the <u>SimpleEdit Members</u> topic.

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
**BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
**BringToFront (inherited from Control)	Brings the control to the front of the z-order.
◆Close (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
*CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
**CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a

	remote object.
Dispose (inherited from Component)	Overloaded. Releases all resources used by the <u>Component</u> .
DoDragDrop (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
from Control)	Retrieves the child control that is located at the specified coordinates.
GetContainerControl (inherited from Control)	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
from Control) (inherited	Retrieves the next control forward or back in the tab order of child controls.

GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
Hide (inherited from Control)	Conceals the control from the user.
InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
Invalidate (inherited from Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
Invoke (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
►• LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
PointToClient (inherited from Control)	Computes the location of the specified screen point into client coordinates.
**PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
**PreProcessMessage (inherited from Control)	Preprocesses input messages within the message loop before
ı	1

	they are dispatched.
**RectangleToClient (inherited from Control)	Computes the size and location of the specified screen rectangle in client coordinates.
***RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.
Scale (inherited from Control)	Overloaded. Scales the control

	and any child controls to the specified ratio.
**ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>◆ToString</u> (inherited from Form)	
■ Update (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
<u>Validate</u> (inherited from ContainerControl)	Validates the last invalidated control and its ancestors up

through, but not including, the current control.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
ActivateMdiChild (inherited from Form)	
*AdjustFormScrollbars (inherited from Form)	
ApplyAutoScaling (inherited from Form)	
<u>CenterToScreen</u> (inherited from Form)	
CreateAccessibilityInstance (inherited from Control)	Creates a new accessibility object for the control.
CreateControlsInstance (inherited from Form)	
<u>→ DefWndProc</u> (inherited from Form)	
DestroyHandle (inherited from Control)	Destroys the handle associated with the control.
* Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resource and performs other cleanup operations before the

	Component is reclaimed by garbage collection.
	Returns an object that represents a service provided the Component or by its Container.
	Retrieves the value of the specified control style bit for t control.
	Determines if the control is a top-level control.
<u>InitLayout</u> (inherited from Control)	Called after the control has b added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event for the specified control.
№ InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event the specified control.
	Raises the <u>Click</u> event for the specified control.
₱InvokePaint (inherited from Control)	Raises the Paint event for the specified control.
₱InvokePaintBackground (inherited from Control)	Raises the PaintBackgroun event for the specified contro
◆IsInputChar (inherited from Control)	Determines if a character is a input character that the contrrecognizes.
	Determines whether the specified key is a regular inpukey or a special key that requires preprocessing.

MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
◆OnActivated (inherited from Form)	Raises the <u>Activated</u> event.
OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChange</u> event.
OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever
OnCausesValidationChanged (inherited from Control)	Raises the CausesValidationChangedev
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUlCues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
◆OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
◆OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
OnContextMenuChanged (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the ControlAdded eve
OnControlRemoved (inherited from ContainerControl)	
<u>◆OnCreateControl</u> (inherited from Form)	
◆OnCursorChanged (inherited from Control)	Raises the CursorChangedevent.
<u>OnDeactivate</u> (inherited from Form)	Raises the <u>Deactivate</u> event.

OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> even
OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event.
OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
OnDragLeave (inherited from Control)	Raises the <u>DragLeave</u> event.
OnDragOver (inherited from Control)	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
OnEnter (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
OnForeColorChanged (inherited from Control)	Raises the <u>ForeColorChanged</u> event.
OnGiveFeedback (inherited from Control)	Raises the <u>GiveFeedback</u> eve
OnGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event.
[♠] OnHandleCreated (inherited from Form)	
№OnHandleDestroyed (inherited from Form)	
OnHelpRequested (inherited from Control)	Raises the <u>HelpRequested</u> event.
OnlmeModeChanged (inherited from Control)	Raises the <u>ImeModeChanged</u> event.

OnInputLanguageChanged (inherited from Form)	Raises the InputLanguageChanged event.
◆OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
	Raises the <u>Invalidated</u> event.
	Raises the <u>KeyDown</u> event.
	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
OnLoad (inherited from Form)	Raises the <u>Load</u> event.
OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
	Raises the <u>LostFocus</u> event.
	Raises the MaximizedBoundsChanged event.
	Raises the MaximumSizeChanged event.
OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
	Raises the MenuComplete event.
	Raises the MenuStart event.
◆OnMinimumSizeChanged	Raises the

(inherited from Form)	MinimumSizeChanged event.
OnMouseDown (inherited from Control)	Raises the <u>MouseDown</u> event
OnMouseEnter (inherited from Control)	Raises the MouseEnter event
OnMouseHover (inherited from Control)	Raises the MouseHover even
OnMouseLeave (inherited from Control)	Raises the MouseLeave even
OnMouseMove (inherited from Control)	Raises the MouseMove event
[♠] OnMouseUp (inherited from Control)	Raises the MouseUp event.
OnMouseWheel (inherited from ScrollableControl)	
◆OnMove (inherited from Control)	Raises the <u>Move</u> event.
OnNotifyMessage (inherited from Control)	Notifies the control of Window messages.
[♠] OnPaint (inherited from Form)	
OnPaintBackground (inherited from Control)	Paints the background of the control.
<u>OnParentBackColorChanged</u> (inherited from Control)	Raises the <u>BackColorChange</u> event when the <u>BackColor</u> property value of the control's container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event when the BackgroundImage property value of the control's containe changes.
♦ OnParentBindingContextChanged	Raises the

(inherited from Control)	BindingContextChanged event when the BindingContext property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
<u>OnParentEnabledChanged</u> (inherited from Control)	Raises the EnabledChanged event when the Enabled property value of the control's container changes.
<u>OnParentFontChanged</u> (inherited from Control)	Raises the <u>FontChanged</u> even when the <u>Font</u> property value of the control's container changes
	Raises the ForeColorChanged event when the ForeColor property value of the control's container changes.
*OnParentRightToLeftChanged (inherited from Control)	Raises the RightToLeftChange event when the RightToLeft property value of the control's container changes.
<u>OnParentVisibleChanged</u> (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> propert value of the control's container changes.
OnQueryContinueDrag (inherited from Control)	Raises the <u>QueryContinueDrai</u> event.
<u>OnResize</u> (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> even

OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
OnTablndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
OnTextChanged (inherited from Form)	
OnValidated (inherited from Control)	Raises the <u>Validated</u> event.
OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
ProcessDialogKey (inherited from Form)	
ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate control events.
ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
ProcessTabKey (inherited from Form)	
₱RaiseDragEvent (inherited from	

Control)	
RaiseKeyEvent (inherited from Control)	
RaiseMouseEvent (inherited from Control)	
	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment t the appropriate HorizontalAlignment to support right-to-left text.
RtlTranslateContent (inherited from Control)	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
*RtlTranslateHorizontal (inherited from Control)	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
₱RtlTranslateLeftRight (inherited from Control)	Converts the specified LeftRightAlignment to the appropriate LeftRightAlignment to support right-to-left text.
Select (inherited from Form)	Overloaded.
SetBoundsCore (inherited from Form)	

★SetClientSizeCore (inherited from Form)	
* SetDisplayRectLocation (inherited from ScrollableControl)	
★SetScrollState (inherited from ScrollableControl)	
<u>SetStyle</u> (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
★SetVisibleCore (inherited from Form)	
<u>DupdateBounds</u> (inherited from Control)	Overloaded. Updates the bounds of the control with the current size and location.
<u>Description</u> value of the property of th	
<u>◆UpdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.

Protected Internal Instance Methods

<u></u> <u>ProcessKeyMessage</u>	Processes a keyboard
(inherited from Control)	message.

Explicit Interface Implementations

IContainerControl.ActivateControl (inherited from	
ContainerControl)	

See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit.Dispose Method

Clean up any resources being used.

Overload List

Inherited from **Component**.

public void Dispose();

Clean up any resources being used.

protected override void Dispose(bool);

See Also

SimpleEdit Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

SimpleEdit.Dispose Method (Boolean)

Clean up any resources being used.

```
protected override void Dispose(
   bool disposing
);
```

See Also

<u>SimpleEdit Class</u> | <u>NDoc.Core Namespace</u> | <u>SimpleEdit.Dispose</u> <u>Overload List</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter Class

Stream writer that parses a template file to write a new file.

For a list of all members of this type, see <u>TemplateWriter Members</u>.

<u>System.Object</u> <u>MarshalByRefObject</u> <u>TextWriter</u> <u>StreamWriter</u>

TemplateWriter

public class TemplateWriter : StreamWriter

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

TemplateWriter Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TemplateWriter Members

TemplateWriter overview

Public Instance Constructors

Overloaded. Initializes a new
instance of the TemplateWriter
class.

Public Instance Properties

**AutoFlush (inherited from StreamWriter)	Gets or sets a value indicating whether the <u>StreamWriter</u> will flush its buffer to the underlying stream after every call to <u>Write</u> .
■BaseStream (inherited from StreamWriter)	Gets the underlying stream that interfaces with a backing store.
<u>Encoding</u> (inherited from StreamWriter)	Gets the <u>Encoding</u> in which the output is written.
FormatProvider (inherited from TextWriter)	Gets an object that controls formatting.
<u>NewLine</u> (inherited from TextWriter)	Gets or sets the line terminator string used by the current TextWriter .

Public Instance Methods

- Close	Closes the current StreamWriter and StreamReader.
**CopyToEnd	Copies the text lines form the template to the output stream until the end of the template stream.
* CopyToLine	Copies the text lines form the template to the output stream until a specific line is found.

**CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
Flush (inherited from StreamWriter)	Clears all buffers for the current writer and causes any buffered data to be written to the underlying stream.
● GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
Write (inherited from TextWriter)	Overloaded. Writes out a formatted string, using the same semantics as Format.
**Write (inherited from StreamWriter)	Overloaded. Writes a string to the stream.
** WriteLine (inherited from TextWriter)	Overloaded. Writes out a formatted string and a new line, using the same semantics as

E	-ormat.
---	---------

Protected Instance Fields

<u>CoreNewLine</u> (inherited from	Stores the new line characters
TextWriter)	used for this TextWriter .

Protected Instance Methods

Dispose Dispose	Releases the unmanaged resources used by the TemplateWriter and optionally releases the managed resources.
★Finalize (inherited from StreamWriter)	Frees the resources of the current <u>StreamWriter</u> before it is reclaimed by the garbage collector.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

Explicit Interface Implementations

<u>a</u> <u>NDisposable.Dispose</u> (inherited)	
from TextWriter)	

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter Constructor

Initializes a new instance of the TemplateWriter class for the specified file on the specified path, using the specified template stream and using the default encoding and buffer size.

Overload List

Initializes a new instance of the TemplateWriter class for the specified stream, using the specified template stream and using the default encoding and buffer size.

public TemplateWriter(Stream, TextReader);

Initializes a new instance of the TemplateWriter class for the specified file on the specified path, using the specified template stream and using the default encoding and buffer size.

public TemplateWriter(string,TextReader);

See Also

TemplateWriter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TemplateWriter Constructor (String, TextReader)

Initializes a new instance of the TemplateWriter class for the specified file on the specified path, using the specified template stream and using the default encoding and buffer size.

```
public TemplateWriter(
    string outPath,
    TextReader template
);
```

Parameters

outPath

The complete file path to write to.

template

The template's stream reader.

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u> | <u>TemplateWriter</u> <u>Constructor Overload List</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter Constructor (Stream, TextReader)

Initializes a new instance of the TemplateWriter class for the specified stream, using the specified template stream and using the default encoding and buffer size.

```
public TemplateWriter(
    Stream stream,
    TextReader template
);
```

Parameters

stream

The stream to write to.

template

The template's stream reader.

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u> | <u>TemplateWriter</u> <u>Constructor Overload List</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter Methods

The methods of the **TemplateWriter** class are listed below. For a complete list of **TemplateWriter** class members, see the <u>TemplateWriter Members</u> topic.

Public Instance Methods

Closes the current StreamWriter and StreamReader.
Copies the text lines form the emplate to the output stream until the end of the template stream.
Copies the text lines form the emplate to the output stream until a specific line is found.
Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.
Determines whether the specified Object is equal to the current Object.
Clears all buffers for the current writer and causes any buffered data to be written to the underlying stream.
Serves as a hash function for a particular type, suitable for use n hashing algorithms and data structures like a hash table.
Retrieves the current lifetime service object that controls the

● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
◆InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .
<u>Write</u> (inherited from TextWriter)	Overloaded. Writes out a formatted string, using the same semantics as Format.
◇ Write (inherited from StreamWriter)	Overloaded. Writes a string to the stream.
**WriteLine (inherited from TextWriter)	Overloaded. Writes out a formatted string and a new line, using the same semantics as Format.

Protected Instance Methods

Dispose Dispose	Releases the unmanaged resources used by the TemplateWriter and optionally releases the managed resources.
Finalize (inherited from StreamWriter)	Frees the resources of the current StreamWriter before it is reclaimed by the garbage collector.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

Explicit Interface Implementations

ă <mark>⁰IDisposable.Dispose</mark> (inherited	
from TextWriter)	

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter.Close Method

Closes the current StreamWriter and StreamReader.

```
public override void Close();
```

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter.CopyToEnd Method

Copies the text lines form the template to the output stream until the end of the template stream.

```
public void CopyToEnd();
```

See Also

<u>TemplateWriter Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TemplateWriter.CopyToLine Method

Copies the text lines form the template to the output stream until a specific line is found.

```
public bool CopyToLine(
   string toLine
);
```

Parameters

toLine

The line text to search for. Must match exactly.

Return Value

true if the line was found, **false** if the end of the template stream was reached.

See Also

TemplateWriter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TemplateWriter.Dispose Method

Releases the unmanaged resources used by the TemplateWriter and optionally releases the managed resources.

```
protected override void Dispose(
   bool disposing
);
```

Parameters

disposing

true to release both managed and unmanaged resources; **false** to release only unmanaged resources.

See Also

TemplateWriter Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TextEditor Class

Provides editing facilities for large blocks of text in the PropertyGrid.

For a list of all members of this type, see <u>TextEditor Members</u>.

System.Object UITypeEditor

TextEditor

public class TextEditor : UITypeEditor

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

<u>TextEditor Members</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TextEditor Members

TextEditor overview

Public Instance Constructors

◆ TextEditor Constructor	Creates a new instance of the
	<u>TextEditor</u> class.

Public Instance Methods

<u>EditValue</u>	Overloaded. Edits the specified object's value using the editor style indicated by GetEditStyle.
EditValue (inherited from UITypeEditor)	Overloaded. Edits the value of the specified object using the editor style indicated by GetEditStyle.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
GetEditStyle	Overloaded. Gets the editor style used by the EditValue method.
GetEditStyle (inherited from UITypeEditor)	Overloaded. Gets the editor style used by the EditValue method.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetPaintValueSupported (inherited from UITypeEditor)	Overloaded. Indicates whether the specified context supports painting a representation of an object's value within the specified context.

● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
PaintValue (inherited from UITypeEditor)	Overloaded. Paints a representation of the value of an object using the specified PaintValueEventArgs.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>TextEditor Class</u> | <u>NDoc.Core Namespace</u>

Copyright $\ @ \ 2000-2003$ the NDoc Team

TextEditor Constructor

Creates a new instance of the **TextEditor** class.

```
public TextEditor();
```

See Also

<u>TextEditor Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TextEditor Methods

The methods of the **TextEditor** class are listed below. For a complete list of **TextEditor** class members, see the <u>TextEditor</u> <u>Members</u> topic.

Public Instance Methods

EditValue	Overloaded. Edits the specified object's value using the editor style indicated by GetEditStyle.
<u>EditValue</u> (inherited from UITypeEditor)	Overloaded. Edits the value of the specified object using the editor style indicated by GetEditStyle.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
*•GetEditStyle	Overloaded. Gets the editor style used by the EditValue method.
GetEditStyle (inherited from UITypeEditor)	Overloaded. Gets the editor style used by the EditValue method.
	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetPaintValueSupported (inherited from UITypeEditor)	Overloaded. Indicates whether the specified context supports painting a representation of an object's value within the specified context.
◆GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.

PaintValue (inherited from UITypeEditor)	Overloaded. Paints a representation of the value of an object using the specified PaintValueEventArgs.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

<u>TextEditor Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TextEditor.EditValue Method

Edits the specified object's value using the editor style indicated by GetEditStyle.

Overload List

Edits the specified object's value using the editor style indicated by GetEditStyle.

<u>public override object EditValue(ITypeDescriptorContext,IServicePrc Inherited from UITypeEditor.</u>

public object EditValue(IServiceProvider,object);

See Also

TextEditor Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TextEditor.EditValue Method (ITypeDescriptorContext, IServiceProvider, Object)

Edits the specified object's value using the editor style indicated by GetEditStyle.

```
public override object EditValue(
    ITypeDescriptorContext context,
    IServiceProvider provider,
    object value
);
```

Parameters

context

An <u>ITypeDescriptorContext</u> that can be used to gain additional context information.

provider

An IServiceProvider that this editor can use to obtain services.

value

The object to edit.

Return Value

The new value of the object.

See Also

<u>TextEditor Class</u> | <u>NDoc.Core Namespace</u> | <u>TextEditor.EditValue</u> Overload List

Copyright © 2000-2003 the NDoc Team

TextEditor.GetEditStyle Method

Gets the editor style used by the EditValue method.

Overload List

Inherited from **UITypeEditor**.

public UITypeEditorEditStyle GetEditStyle();

Gets the editor style used by the EditValue method.

public override UITypeEditorEditStyle GetEditStyle(ITypeDescriptor(

See Also

TextEditor Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TextEditor.GetEditStyle Method (ITypeDescriptorContext)

Gets the editor style used by the **EditValue** method.

```
public override UITypeEditorEditStyle GetEdit
    ITypeDescriptorContext context
);
```

Parameters

context

An <u>ITypeDescriptorContext</u> that can be used to gain additional context information.

Return Value

A <u>UITypeEditorEditStyle</u> value that indicates the style of editor used by <u>EditValue</u>.

See Also

<u>TextEditor Class</u> | <u>NDoc.Core Namespace</u> | <u>TextEditor.GetEditStyle</u> Overload List

Copyright © 2000-2003 the NDoc Team

TextEditorForm Class

Used in the conjunction with the <u>TextEditor</u>, this form provides the user a larger interface with which to edit text.

For a list of all members of this type, see <u>TextEditorForm Members</u>.

```
System.Object MarshalByRefObject
Component
Control
ScrollableControl
ContainerControl
Form
TextEditorForm
```

public class TextEditorForm : Form

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

TextEditorForm Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TextEditorForm Members

TextEditorForm overview

Public Instance Constructors

TextEditorForm Constructor	Creates a new instance of the
	<u>TextEditorForm</u> class.

Public Instance Properties

[™] AcceptButton (inherited from Form)	Gets or sets the button on the form that is clicked when the user presses the ENTER key.
AccessibilityObject (inherited from Control)	Gets the <u>AccessibleObject</u> assigned to the control.
AccessibleDefaultActionDescription (inherited from Control)	Gets or sets the default action description of the control for use by accessibility client applications.
**AccessibleDescription (inherited from Control)	Gets or sets the description of the control used by accessibility client applications.
	Gets or sets the name of the control used by accessibility client applications.
<u>AccessibleRole</u> (inherited from Control)	Gets or sets the accessible role of the control
	Gets or sets the active control on the container control.
ActiveMdiChild (inherited from Form)	Gets the currently active multiple document interface

	(MDI) child window.
	Gets or sets a value indicating whether the control can accept data that the user drags onto it.
<u>AllowTransparency</u> (inherited from Form)	
<u>Anchor</u> (inherited from Control)	Gets or sets which edges of the control are anchored to the edges of its container.
[™] AutoScale (inherited from Form)	Gets or sets a value indicating whether the form adjusts its size to fit the height of the font used on the form and scales its controls.
<u>AutoScaleBaseSize</u> (inherited from Form)	Gets or sets the base size used for autoscaling of the form.
[™] AutoScroll (inherited from Form)	Gets or sets a value indicating whether the form enables autoscrolling.
<u>AutoScrollMargin</u> (inherited from ScrollableControl)	Gets or sets the size of the auto-scroll margin.
ScrollableControl)	Gets or sets the minimum size of the auto-scroll.
AutoScrollPosition (inherited from ScrollableControl)	Gets or sets the location of the auto-scroll position.
<u>■BackColor</u> (inherited from Form)	
	Gets or sets the background image displayed in the control.
<u>BindingContext</u> (inherited from ContainerControl)	

Bottom (inherited from Control)	Gets the distance between the bottom edge of the control and the top edge of its container's client area.
<u>Bounds</u> (inherited from Control)	Gets or sets the size and location of the control including its nonclient elements.
© CancelButton (inherited from Form)	Gets or sets the button control that is clicked when the user presses the ESC key.
CanFocus (inherited from Control)	Gets a value indicating whether the control can receive focus.
CanSelect (inherited from Control)	Gets a value indicating whether the control can be selected.
<u>Capture</u> (inherited from Control)	Gets or sets a value indicating whether the control has captured the mouse.
Causes Validation (inherited from Control)	Gets or sets a value indicating whether the control causes validation to be performed on any controls that require validation when it receives focus.
ClientRectangle (inherited from Control)	Gets the rectangle that represents the client area of the control.
<u>ClientSize</u> (inherited from Form)	Gets or sets the size of the client area of the form.
CompanyName (inherited from Control)	Gets the name of the company or creator of the

	application containing the control.
<u>Container</u> (inherited from Component)	Gets the <u>IContainer</u> that contains the <u>Component</u> .
ContainsFocus (inherited from Control)	Gets a value indicating whether the control, or one of its child controls, currently has the input focus.
ContextMenu (inherited from Control)	Gets or sets the shortcut menu associated with the control.
© ControlBox (inherited from Form)	Gets or sets a value indicating whether a control box is displayed in the caption bar of the form.
Controls (inherited from Control)	Gets the collection of controls contained within the control.
<u>Created</u> (inherited from Control)	Gets a value indicating whether the control has been created.
<u>Cursor</u> (inherited from Control)	Gets or sets the cursor that is displayed when the mouse pointer is over the control.
<u>□ DataBindings</u> (inherited from Control)	Gets the data bindings for the control.
<u>□ DesktopBounds</u> (inherited from Form)	Gets or sets the size and location of the form on the Windows desktop.
<u>□ DesktopLocation</u> (inherited from Form)	Gets or sets the location of the form on the Windows desktop.
<u>□DialogResult</u> (inherited from	Gets or sets the dialog result

Form)	for the form.
<u>□ DisplayRectangle</u> (inherited from ScrollableControl)	
	Gets a value indicating whether the control is in the process of being disposed of.
<u>Dock</u> (inherited from Control)	Gets or sets which edge of the parent container a control is docked to.
<u>DockPadding</u> (inherited from ScrollableControl)	Gets the dock padding settings for all edges of the control.
<u>Enabled</u> (inherited from Control)	Gets or sets a value indicating whether the control can respond to user interaction.
Focused (inherited from Control)	Gets a value indicating whether the control has input focus.
Font (inherited from Control)	Gets or sets the font of the text displayed by the control.
ForeColor (inherited from Control)	Gets or sets the foreground color of the control.
FormBorderStyle (inherited from Form)	Gets or sets the border style of the form.
Handle (inherited from Control)	Gets the window handle that the control is bound to.
HasChildren (inherited from Control)	Gets a value indicating whether the control contains one or more child controls.
Height (inherited from Control)	Gets or sets the height of the control.

HelpButton (inherited from Form)	Gets or sets a value indicating whether a Help button should be displayed in the caption box of the form.
<u>Icon</u> (inherited from Form)	Gets or sets the icon for the form.
<u>ImeMode</u> (inherited from Control)	Gets or sets the Input Method Editor (IME) mode of the control.
	Gets a value indicating whether the caller must call an invoke method when making method calls to the control because the caller is on a different thread than the one the control was created on.
	Gets or sets a value indicating whether the control is visible to accessibility applications.
<u>IsDisposed</u> (inherited from Control)	Gets a value indicating whether the control has been disposed of.
□ IsHandleCreated (inherited from Control)	Gets a value indicating whether the control has a handle associated with it.
[™] IsMdiChild (inherited from Form)	Gets a value indicating whether the form is a multiple document interface (MDI) child form.
	Gets or sets a value indicating whether the form is a container for multiple document interface (MDI)

	child forms.
<u>IsRestrictedWindow</u> (inherited from Form)	
★ KeyPreview (inherited from Form)	Gets or sets a value indicating whether the form will receive key events before the event is passed to the control that has focus.
<u>Left</u> (inherited from Control)	Gets or sets the x-coordinate of a control's left edge in pixels.
<u>Location</u> (inherited from Control)	Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
MaximizeBox (inherited from Form)	Gets or sets a value indicating whether the maximize button is displayed in the caption bar of the form.
<u>MaximumSize</u> (inherited from Form)	Gets the maximum size the form can be resized to.
MdiChildren (inherited from Form)	Gets an array of forms that represent the multiple document interface (MDI) child forms that are parented to this form.
MdiParent (inherited from Form)	Gets or sets the current multiple document interface (MDI) parent form of this form.
<u>Menu</u> (inherited from Form)	Gets or sets the MainMenu that is displayed in the form.
MergedMenu (inherited from Form)	Gets the merged menu for the form.

MinimizeBox (inherited from Form)	Gets or sets a value indicating whether the minimize button is displayed in the caption bar of the form.
MinimumSize (inherited from Form)	Gets or sets the minimum size the form can be resized to.
<u>Modal</u> (inherited from Form)	Gets a value indicating whether this form is displayed modally.
<u>Name</u> (inherited from Control)	Gets or sets the name of the control.
[™] Opacity (inherited from Form)	Gets or sets the opacity level of the form.
<u>SownedForms</u> (inherited from Form)	Gets an array of Form objects that represent all forms that are owned by this form.
<u>Owner</u> (inherited from Form)	Gets or sets the form that owns this form.
Parent (inherited from Control)	Gets or sets the parent container of the control.
	Gets the form that the container control is assigned to.
	Gets the product name of the assembly containing the control.
ProductVersion (inherited from Control)	Gets the version of the assembly containing the control.
RecreatingHandle (inherited from Control)	Gets a value indicating whether the control is

	currently re-creating its handle.
Region (inherited from Control)	Gets or sets the window region associated with the control.
[™] Right (inherited from Control)	Gets the distance between the right edge of the control and the left edge of its container.
■ RightToLeft (inherited from Control)	Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts.
ShowInTaskbar (inherited from Form)	Gets or sets a value indicating whether the form is displayed in the Windows taskbar.
Site (inherited from Control)	Gets or sets the site of the control.
Size (inherited from Form)	Gets or sets the size of the form.
SizeGripStyle (inherited from Form)	Gets or sets the style of the size grip to display in the lower-right corner of the form.
StartPosition (inherited from Form)	Gets or sets the starting position of the form at run time.
<u>■TabIndex</u> (inherited from Form)	
<u>TabStop</u> (inherited from Control)	Gets or sets a value indicating whether the user can give the focus to this control using the TAB key.

<u>■ Tag</u> (inherited from Control)	Gets or sets the object that contains data about the control.
Text (inherited from Control)	Gets or sets the text associated with this control.
<u>™Top</u> (inherited from Control)	Gets or sets the y-coordinate of the control's top edge in pixels.
TopLevel (inherited from Form)	Gets or sets a value indicating whether to display the form as a top-level window.
TopLevelControl (inherited from Control)	Gets the parent control that is not parented by another Windows Forms control. Typically, this is the outermost Form that the control is contained in.
TopMost (inherited from Form)	Gets or sets a value indicating whether the form should be displayed as the top-most form of your application.
TransparencyKey (inherited from Form)	Gets or sets the color that will represent transparent areas of the form.
<u> </u>	Gets or sets the text to edit.
<u>Sible</u> (inherited from Control)	Gets or sets a value indicating whether the control is displayed.
	Gets or sets the width of the control.
<u>™WindowState</u> (inherited from Form)	Gets or sets the form's window state.

<u>™WindowTarget</u> (inherited from	
Control)	

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
**BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
**BringToFront (inherited from Control)	Brings the control to the front of the z-order.
○ Close (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
**CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
**CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a remote object.
Dispose (inherited from	Overloaded. Releases all

Component)	resources used by the Component.
<u>DoDragDrop</u> (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
*◆GetChildAtPoint (inherited from Control)	Retrieves the child control that is located at the specified coordinates.
	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
**GetLifetimeService (inherited from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
from Control (inherited	Retrieves the next control forward or back in the tab order of child controls.
GetType (inherited from Object)	Gets the Type of the current instance.

Hide (inherited from Control)	Conceals the control from the user.
◆InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
Invalidate (inherited from Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
Invoke (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
► LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
PointToClient (inherited from Control)	Computes the location of the specified screen point into client coordinates.
PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
*PreProcessMessage (inherited from Control)	Preprocesses input messages within the message loop before they are dispatched.
**RectangleToClient (inherited	Computes the size and location

from Control)	of the specified screen rectangle in client coordinates.
**RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
***RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
***ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
***ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.
Scale (inherited from Control)	Overloaded. Scales the control and any child controls to the specified ratio.

**ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>◆ToString</u> (inherited from Form)	
■ Update (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
Validate (inherited from ContainerControl)	Validates the last invalidated control and its ancestors up through, but not including, the current control.

Public Instance Events

Activated (inherited from Form)	Occurs when the form is activated in code or by the user.
BackColorChanged (inherited from Control)	Occurs when the value of the BackColor property changes.
BackgroundImageChanged (inherited from Control)	Occurs when the value of the BackgroundImage property changes.
BindingContextChanged (inherited from Control)	Occurs when the value of the BindingContext property changes.
CausesValidationChanged (inherited from Control)	Occurs when the value of the CausesValidation property changes.
ChangeUICues (inherited from Control)	Occurs when the focus or keyboard user interface (UI) cues change.
Click (inherited from Control)	Occurs when the control is clicked.
Click (inherited from Control) Closed (inherited from Form)	
,	clicked.
Closed (inherited from Form)	clicked. Occurs when the form is closed. Occurs when the form is
Closed (inherited from Form) Closing (inherited from Form) ContextMenuChanged	clicked. Occurs when the form is closed. Occurs when the form is closing. Occurs when the value of the
Closed (inherited from Form) Closing (inherited from Form) ContextMenuChanged (inherited from Control) ControlAdded (inherited from	clicked. Occurs when the form is closed. Occurs when the form is closing. Occurs when the value of the ContextMenu property changes. Occurs when a new control is
Closed (inherited from Form) Closing (inherited from Form) ContextMenuChanged (inherited from Control) ControlAdded (inherited from Control) ControlRemoved (inherited	clicked. Occurs when the form is closed. Occurs when the form is closing. Occurs when the value of the ContextMenu property changes. Occurs when a new control is added to the ControlCollection. Occurs when a control is removed from the

Form)	focus and is not the active form.
Disposed (inherited from Component)	Adds an event handler to listen to the <u>Disposed</u> event on the component.
DockChanged (inherited from Control)	Occurs when the value of the Dock property changes.
DoubleClick (inherited from Control)	Occurs when the control is double-clicked.
DragDrop (inherited from Control)	Occurs when a drag-and-drop operation is completed.
DragEnter (inherited from Control)	Occurs when an object is dragged into the control's bounds.
DragLeave (inherited from Control)	Occurs when an object is dragged out of the control's bounds.
DragOver (inherited from Control)	Occurs when an object is dragged over the control's bounds.
FinabledChanged (inherited from Control)	Occurs when the <u>Enabled</u> property value has changed.
Enter (inherited from Control)	Occurs when the control is entered.
FontChanged (inherited from Control)	Occurs when the <u>Font</u> property value changes.
ForeColorChanged (inherited from Control)	Occurs when the <u>ForeColor</u> property value changes.
✓ GiveFeedback (inherited from Control)	Occurs during a drag operation.
	Occurs when the control receives focus.
HandleCreated (inherited from	Occurs when a handle is

Control)	created for the control.
F HandleDestroyed (inherited from Control)	Occurs when the control's handle is in the process of being destroyed.
HelpRequested (inherited from Control)	Occurs when the user requests help for a control.
ImeModeChanged (inherited from Control)	Occurs when the <u>ImeMode</u> property has changed.
InputLanguageChanged (inherited from Form)	Occurs after the input language of the form has changed.
InputLanguageChanging (inherited from Form)	Occurs when the user attempts to change the input language for the form.
Invalidated (inherited from Control)	Occurs when a control's display requires redrawing.
KeyDown (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyPress (inherited from Control)	Occurs when a key is pressed while the control has focus.
KeyUp (inherited from Control)	Occurs when a key is released while the control has focus.
Layout (inherited from Control)	Occurs when a control should reposition its child controls.
Leave (inherited from Control)	Occurs when the input focus leaves the control.
Load (inherited from Form)	Occurs before a form is displayed for the first time.
LocationChanged (inherited from Control)	Occurs when the <u>Location</u> property value has changed.
LostFocus (inherited from Control)	Occurs when the control loses focus.

MaximizedBoundsChanged (inherited from Form)	Occurs when the value of the MaximizedBounds property has changed.
MaximumSizeChanged (inherited from Form)	Occurs when the value of the MaximumSize property has changed.
MdiChildActivate (inherited from Form)	Occurs when a multiple document interface (MDI) child form is activated or closed within an MDI application.
MenuComplete (inherited from Form)	Occurs when the menu of a form loses focus.
MenuStart (inherited from Form)	Occurs when the menu of a form receives focus.
MinimumSizeChanged (inherited from Form)	Occurs when the value of the MinimumSize property has changed.
MouseDown (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is pressed.
MouseEnter (inherited from Control)	Occurs when the mouse pointer enters the control.
MouseHover (inherited from Control)	Occurs when the mouse pointer hovers over the control.
MouseLeave (inherited from Control)	Occurs when the mouse pointer leaves the control.
MouseMove (inherited from Control)	Occurs when the mouse pointer is moved over the control.
MouseUp (inherited from Control)	Occurs when the mouse pointer is over the control and a mouse button is released.
MouseWheel (inherited from Control)	Occurs when the mouse wheel moves while the control has

	focus.
Move (inherited from Control)	Occurs when the control is moved.
Paint (inherited from Control)	Occurs when the control is redrawn.
ParentChanged (inherited from Control)	Occurs when the <u>Parent</u> property value changes.
QueryAccessibilityHelp (inherited from Control)	Occurs when <u>AccessibleObject</u> is providing help to accessibility applications.
QueryContinueDrag (inherited from Control)	Occurs during a drag-and-drop operation and allows the drag source to determine whether the drag-and-drop operation should be canceled.
Resize (inherited from Control)	Occurs when the control is resized.
RightToLeftChanged (inherited from Control)	Occurs when the RightToLeft property value changes.
SizeChanged (inherited from Control)	Occurs when the <u>Size</u> property value changes.
StyleChanged (inherited from Control)	Occurs when the control style changes.
SystemColorsChanged (inherited from Control)	Occurs when the system colors change.
TablndexChanged (inherited from Form)	
TabStopChanged (inherited from Control)	Occurs when the <u>TabStop</u> property value changes.
<u>TextChanged</u> (inherited from Control)	Occurs when the <u>Text</u> property value changes.
✓ Validated (inherited from	Occurs when the control is

Control)	finished validating.
Validating (inherited from Control)	Occurs when the control is validating.
VisibleChanged (inherited from Control)	Occurs when the <u>Visible</u> property value changes.

Protected Instance Properties

	Gets the default Input Method Editor (IME) mode supported by the control.
	Gets a value that indicates whether the <u>Component</u> is currently in design mode.
Events (inherited from Component)	Gets the list of event handlers that are attached to this Component .
FontHeight (inherited from Control)	Gets or sets the height of the font of the control.
	Gets or sets a value indicating whether the horizontal scroll bar is visible.
MaximizedBounds (inherited from Form)	Gets and sets the size of the form when it is maximized.
RenderRightToLeft (inherited from Control)	
ResizeRedraw (inherited from Control)	Gets or sets a value indicating whether the control redraws itself when resized.

FishowFocusCues (inherited from Control)	Gets a value indicating whether the control should display focus rectangles.
ShowKeyboardCues (inherited from Control)	Gets a value indicating whether the control should display keyboard shortcuts.
✓ VScroll (inherited from ScrollableControl)	Gets or sets a value indicating whether the vertical scroll bar is visible.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
*AdjustFormScrollbars (inherited from Form)	
ApplyAutoScaling (inherited from Form)	
CreateAccessibilityInstance (inherited from Control)	Creates a new accessibility object for the control.
CreateControlsInstance (inherited from Form)	
<u>▶ DefWndProc</u> (inherited from Form)	

DestroyHandle (inherited from Control)	Destroys the handle association with the control.
Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resou and performs other cleanup operations before the Component is reclaimed by garbage collection.
	Returns an object that represents a service provide the <u>Component</u> or by its <u>Container</u> .
	Retrieves the value of the specified control style bit for control.
GetTopLevel (inherited from Control)	Determines if the control is a top-level control.
₱ <u>InitLayout</u> (inherited from Control)	Called after the control has be added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event f the specified control.
InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event the specified control.
InvokeOnClick (inherited from Control)	Raises the <u>Click</u> event for the specified control.
<u>InvokePaint</u> (inherited from Control)	Raises the Paint event for th specified control.
InvokePaintBackground (inherited from Control)	Raises the PaintBackgroun event for the specified control

	Determines if a character is a input character that the contro recognizes.
	Determines whether the specified key is a regular inpukey or a special key that requires preprocessing.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
<u>OnActivated</u> (inherited from Form)	Raises the <u>Activated</u> event.
OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChange</u> event.
OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever
OnCausesValidationChanged (inherited from Control)	Raises the CausesValidationChangedev
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUICues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
[♠] OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
[♠] OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
<u>OnContextMenuChanged</u> (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the ControlAdded eve
◆OnControlRemoved (inherited from	

ContainerControl)	
OnCreateControl (inherited from Form)	
OnCursorChanged (inherited from Control)	Raises the CursorChangedevent.
	Raises the <u>Deactivate</u> event.
OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> event
OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event.
OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
OnDragLeave (inherited from Control)	Raises the <u>DragLeave</u> event.
OnDragOver (inherited from Control)	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
◆OnEnter (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
OnForeColorChanged (inherited from Control)	Raises the ForeColorChangedevent.
OnGiveFeedback (inherited from Control)	Raises the <u>GiveFeedback</u> even
OnGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event.
◆OnHandleCreated (inherited from	

Form)	
◆OnHandleDestroyed (inherited from Form)	
OnHelpRequested (inherited from Control)	Raises the <u>HelpRequested</u> event.
OnlmeModeChanged (inherited from Control)	Raises the <u>ImeModeChanged</u> event.
OnInputLanguageChanged (inherited from Form)	Raises the InputLanguageChanged event.
OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
OnInvalidated (inherited from Control)	Raises the <u>Invalidated</u> event.
	Raises the <u>KeyDown</u> event.
OnKeyPress (inherited from Control)	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
[♠] OnLayout (inherited from ScrollableControl)	
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
◆OnLoad (inherited from Form)	Raises the <u>Load</u> event.
OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
OnLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event.
OnMaximizedBoundsChanged (inherited from Form)	Raises the MaximizedBoundsChanged event.
<u>OnMaximumSizeChanged</u> (inherited from Form)	Raises the MaximumSizeChanged event.

OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
◆OnMenuComplete (inherited from Form)	Raises the MenuComplete event.
	Raises the MenuStart event.
<u>OnMinimumSizeChanged</u> (inherited from Form)	Raises the MinimumSizeChanged event.
<u>OnMouseDown</u> (inherited from Control)	Raises the <u>MouseDown</u> event.
◆OnMouseEnter (inherited from Control)	Raises the MouseEnter event.
	Raises the <u>MouseHover</u> event.
	Raises the MouseLeave event.
	Raises the <u>MouseMove</u> event.
	Raises the MouseUp event.
<u>OnMove</u> (inherited from Control)	Raises the <u>Move</u> event.
OnNotifyMessage (inherited from Control)	Notifies the control of Windows messages.
<u>OnPaint</u> (inherited from Form)	
<u>OnPaintBackground</u> (inherited from Control)	Paints the background of the control.
	Raises the <u>BackColorChanged</u> event when the <u>BackColor</u> property value of the control's

	container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the <u>BackgroundImageChanged</u> event when the <u>BackgroundImage</u> property value of the control's containe changes.
OnParentBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever when the <u>BindingContext</u> property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
◆OnParentEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event when the <u>Enabled</u> property value of the control's container changes.
<u>OnParentFontChanged</u> (inherited from Control)	Raises the <u>FontChanged</u> ever when the <u>Font</u> property value the control's container change
<u>OnParentForeColorChanged</u> (inherited from Control)	Raises the ForeColorChange event when the ForeColor property value of the control's container changes.
<u>OnParentRightToLeftChanged</u> (inherited from Control)	Raises the RightToLeftChang event when the RightToLeft property value of the control's container changes.
<u>OnParentVisibleChanged</u> (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> proper value of the control's containe changes.
◆OnQueryContinueDrag (inherited)	Raises the QueryContinueDra

from Control)	event.
◆OnResize (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> event
OnStyleChanged (inherited from Form)	
OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
OnTabIndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
OnTextChanged (inherited from Form)	
OnValidated (inherited from Control)	Raises the <u>Validated</u> event.
OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
₱ProcessDialogKey (inherited from Form)	
ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate control events.
	Control events.

ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
₱ProcessTabKey (inherited from Form)	
RaiseDragEvent (inherited from Control)	
RaiseKeyEvent (inherited from Control)	
RaiseMouseEvent (inherited from Control)	
RaisePaintEvent (inherited from Control)	
	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
*RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment the appropriate HorizontalAlignment to support right-to-left text.
	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
*RtlTranslateHorizontal (inherited from Control)	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
₱ RtlTranslateLeftRight (inherited from Control)	Converts the specified LeftRightAlignment to the

	appropriate <u>LeftRightAlignment</u> to support right-to-left text.
◆ScaleCore (inherited from Form)	
◆ Select (inherited from Form)	Overloaded.
★SetBoundsCore (inherited from Form)	
★SetClientSizeCore (inherited from Form)	
★SetDisplayRectLocation (inherited from ScrollableControl)	
* SetScrollState (inherited from ScrollableControl)	
<u>SetStyle</u> (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
★SetVisibleCore (inherited from Form)	
<u>DupdateBounds</u> (inherited from Control)	Overloaded. Updates the bounds of the control with the current size and location.
<u>Description</u> (inherited from Form) ### Index in the inherited from Form in the inherited from Form) ### Index in the inherited from Form in the inhe	
<u>DupdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.

Protected Internal Instance Methods

^ĕ ProcessKeyMessage	Processes a keyboard

message.

Explicit Interface Implementations



See Also

<u>TextEditorForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

NDoc Documentation

TextEditorForm Constructor

Creates a new instance of the **TextEditorForm** class.

public TextEditorForm();

See Also

<u>TextEditorForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright $\ @$ 2000-2003 the NDoc Team

NDoc Documentation

TextEditorForm Properties

The properties of the **TextEditorForm** class are listed below. For a complete list of **TextEditorForm** class members, see the <u>TextEditorForm Members</u> topic.

Public Instance Properties

	Gets or sets the button on the form that is clicked when the user presses the ENTER key.
AccessibilityObject (inherited from Control)	Gets the <u>AccessibleObject</u> assigned to the control.
AccessibleDefaultActionDescription (inherited from Control)	Gets or sets the default action description of the control for use by accessibility client applications.
**AccessibleDescription (inherited from Control)	Gets or sets the description of the control used by accessibility client applications.
	Gets or sets the name of the control used by accessibility client applications.
	Gets or sets the accessible role of the control
ContainerControl (inherited from	Gets or sets the active control on the container control.
Form)	Gets the currently active multiple document interface (MDI) child window.
[™] AllowDrop (inherited from	Gets or sets a value

Control)	indicating whether the control can accept data that the user drags onto it.
<u>AllowTransparency</u> (inherited from Form)	
Anchor (inherited from Control)	Gets or sets which edges of the control are anchored to the edges of its container.
Matto Scale (inherited from Form)	Gets or sets a value indicating whether the form adjusts its size to fit the height of the font used on the form and scales its controls.
from Form)	Gets or sets the base size used for autoscaling of the form.
[™] AutoScroll (inherited from Form)	Gets or sets a value indicating whether the form enables autoscrolling.
<u>AutoScrollMargin</u> (inherited from ScrollableControl)	Gets or sets the size of the auto-scroll margin.
AutoScrollMinSize (inherited from ScrollableControl)	Gets or sets the minimum size of the auto-scroll.
AutoScrollPosition (inherited from ScrollableControl)	Gets or sets the location of the auto-scroll position.
[™] BackColor (inherited from Form)	
	Gets or sets the background image displayed in the control.
<u>BindingContext</u> (inherited from ContainerControl)	
Bottom (inherited from Control)	Gets the distance between the bottom edge of the

	control and the top edge of its container's client area.
[™] Bounds (inherited from Control)	Gets or sets the size and location of the control including its nonclient elements.
<u>CancelButton</u> (inherited from Form)	Gets or sets the button control that is clicked when the user presses the ESC key.
CanFocus (inherited from Control)	Gets a value indicating whether the control can receive focus.
	Gets a value indicating whether the control can be selected.
<u>Capture</u> (inherited from Control)	Gets or sets a value indicating whether the control has captured the mouse.
Causes Validation (inherited from Control)	Gets or sets a value indicating whether the control causes validation to be performed on any controls that require validation when it receives focus.
ClientRectangle (inherited from Control)	Gets the rectangle that represents the client area of the control.
<u>ClientSize</u> (inherited from Form)	Gets or sets the size of the client area of the form.
CompanyName (inherited from Control)	Gets the name of the company or creator of the application containing the control.

Container (inherited from Component)	Gets the <u>IContainer</u> that contains the <u>Component</u> .
ContainsFocus (inherited from Control)	Gets a value indicating whether the control, or one of its child controls, currently has the input focus.
ContextMenu (inherited from Control)	Gets or sets the shortcut menu associated with the control.
S ControlBox (inherited from Form)	Gets or sets a value indicating whether a control box is displayed in the caption bar of the form.
<u>Controls</u> (inherited from Control)	Gets the collection of controls contained within the control.
<u>Created</u> (inherited from Control)	Gets a value indicating whether the control has been created.
<u>Cursor</u> (inherited from Control)	Gets or sets the cursor that is displayed when the mouse pointer is over the control.
<u>□ DataBindings</u> (inherited from Control)	Gets the data bindings for the control.
<u>□ DesktopBounds</u> (inherited from Form)	Gets or sets the size and location of the form on the Windows desktop.
DesktopLocation (inherited from Form)	Gets or sets the location of the form on the Windows desktop.
<u>DialogResult</u> (inherited from Form)	Gets or sets the dialog result for the form.
<u>□DisplayRectangle</u> (inherited from	

ScrollableControl)	
	Gets a value indicating whether the control is in the process of being disposed of.
<u>Dock</u> (inherited from Control)	Gets or sets which edge of the parent container a control is docked to.
	Gets the dock padding settings for all edges of the control.
Enabled (inherited from Control)	Gets or sets a value indicating whether the control can respond to user interaction.
Focused (inherited from Control)	Gets a value indicating whether the control has input focus.
Font (inherited from Control)	Gets or sets the font of the text displayed by the control.
ForeColor (inherited from Control)	Gets or sets the foreground color of the control.
FormBorderStyle (inherited from Form)	Gets or sets the border style of the form.
<u>Handle</u> (inherited from Control)	Gets the window handle that the control is bound to.
HasChildren (inherited from Control)	Gets a value indicating whether the control contains one or more child controls.
<u>Height</u> (inherited from Control)	Gets or sets the height of the control.
HelpButton (inherited from Form)	Gets or sets a value indicating whether a Help button should be displayed in

	the caption box of the form.
<u>Icon</u> (inherited from Form)	Gets or sets the icon for the form.
<u>ImeMode</u> (inherited from Control)	Gets or sets the Input Method Editor (IME) mode of the control.
InvokeRequired (inherited from Control)	Gets a value indicating whether the caller must call an invoke method when making method calls to the control because the caller is on a different thread than the one the control was created on.
	Gets or sets a value indicating whether the control is visible to accessibility applications.
□ IsDisposed (inherited from Control)	Gets a value indicating whether the control has been disposed of.
□ IsHandleCreated (inherited from Control)	Gets a value indicating whether the control has a handle associated with it.
	Gets a value indicating whether the form is a multiple document interface (MDI) child form.
■ IsMdiContainer (inherited from Form)	Gets or sets a value indicating whether the form is a container for multiple document interface (MDI) child forms.
<u>IsRestrictedWindow</u> (inherited)	

from Form)	
Form)	Gets or sets a value indicating whether the form will receive key events before the event is passed to the control that has focus.
<u>Left</u> (inherited from Control)	Gets or sets the x-coordinate of a control's left edge in pixels.
<u>Location</u> (inherited from Control)	Gets or sets the coordinates of the upper-left corner of the control relative to the upper-left corner of its container.
MaximizeBox (inherited from Form)	Gets or sets a value indicating whether the maximize button is displayed in the caption bar of the form.
<u>MaximumSize</u> (inherited from Form)	Gets the maximum size the form can be resized to.
MdiChildren (inherited from Form)	Gets an array of forms that represent the multiple document interface (MDI) child forms that are parented to this form.
<u>MdiParent</u> (inherited from Form)	Gets or sets the current multiple document interface (MDI) parent form of this form.
Menu (inherited from Form)	Gets or sets the MainMenu that is displayed in the form.
<u>MergedMenu</u> (inherited from Form)	Gets the merged menu for the form.
<u>MinimizeBox</u> (inherited from Form)	Gets or sets a value indicating whether the

	minimize button is displayed in the caption bar of the form.
MinimumSize (inherited from Form)	Gets or sets the minimum size the form can be resized to.
[™] Modal (inherited from Form)	Gets a value indicating whether this form is displayed modally.
<u>Name</u> (inherited from Control)	Gets or sets the name of the control.
[™] Opacity (inherited from Form)	Gets or sets the opacity level of the form.
<u>OwnedForms</u> (inherited from Form)	Gets an array of Form objects that represent all forms that are owned by this form.
[™] Owner (inherited from Form)	Gets or sets the form that owns this form.
Parent (inherited from Control)	Gets or sets the parent container of the control.
□ ParentForm (inherited from ContainerControl)	Gets the form that the container control is assigned to.
<u>ProductName</u> (inherited from Control)	Gets the product name of the assembly containing the control.
ProductVersion (inherited from Control)	Gets the version of the assembly containing the control.
RecreatingHandle (inherited from Control)	Gets a value indicating whether the control is currently re-creating its handle.

Region (inherited from Control)	Gets or sets the window region associated with the control.
□ Right (inherited from Control)	Gets the distance between the right edge of the control and the left edge of its container.
■ RightToLeft (inherited from Control)	Gets or sets a value indicating whether control's elements are aligned to support locales using right-to-left fonts.
ShowInTaskbar (inherited from Form)	Gets or sets a value indicating whether the form is displayed in the Windows taskbar.
Site (inherited from Control)	Gets or sets the site of the control.
Size (inherited from Form)	Gets or sets the size of the form.
SizeGripStyle (inherited from Form)	Gets or sets the style of the size grip to display in the lower-right corner of the form.
StartPosition (inherited from Form)	Gets or sets the starting position of the form at run time.
<u>■TabIndex</u> (inherited from Form)	
<u>■TabStop</u> (inherited from Control)	Gets or sets a value indicating whether the user can give the focus to this control using the TAB key.
<u>■ Tag</u> (inherited from Control)	Gets or sets the object that contains data about the

	control.
<u>Text</u> (inherited from Control)	Gets or sets the text associated with this control.
Top (inherited from Control)	Gets or sets the y-coordinate of the control's top edge in pixels.
TopLevel (inherited from Form)	Gets or sets a value indicating whether to display the form as a top-level window.
TopLevelControl (inherited from Control)	Gets the parent control that is not parented by another Windows Forms control. Typically, this is the outermost Form that the control is contained in.
TopMost (inherited from Form)	Gets or sets a value indicating whether the form should be displayed as the top-most form of your application.
TransparencyKey (inherited from Form)	Gets or sets the color that will represent transparent areas of the form.
<u>™Value</u>	Gets or sets the text to edit.
	Gets or sets a value indicating whether the control is displayed.
<u>™Width</u> (inherited from Control)	Gets or sets the width of the control.
<u>WindowState</u> (inherited from Form)	Gets or sets the form's window state.
<u>™WindowTarget</u> (inherited from	

Control)

Protected Instance Properties

	Gets the default Input Method Editor (IME) mode supported by the control.
	Gets a value that indicates whether the <u>Component</u> is currently in design mode.
Events (inherited from Component)	Gets the list of event handlers that are attached to this Component .
FontHeight (inherited from Control)	Gets or sets the height of the font of the control.
	Gets or sets a value indicating whether the horizontal scroll bar is visible.
MaximizedBounds (inherited from Form)	Gets and sets the size of the form when it is maximized.
RenderRightToLeft (inherited from Control)	
ResizeRedraw (inherited from Control)	Gets or sets a value indicating whether the control redraws itself when resized.
ShowFocusCues (inherited from Control)	Gets a value indicating whether the control should display focus rectangles.
ShowKeyboardCues (inherited from Control)	Gets a value indicating whether the control should display

keyboard shortcuts.
Gets or sets a value indicating whether the vertical scroll bar is visible.

See Also

<u>TextEditorForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright $\ @ \ 2000-2003 \ the \ NDoc \ Team$

NDoc Documentation

TextEditorForm.Value Property

Gets or sets the text to edit.

```
public string Value {get; set;}
```

Property Value

The text to edit.

See Also

<u>TextEditorForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TextEditorForm Methods

The methods of the **TextEditorForm** class are listed below. For a complete list of **TextEditorForm** class members, see the <u>TextEditorForm Members</u> topic.

Public Instance Methods

Activate (inherited from Form)	Activates the form and gives it focus.
**AddOwnedForm (inherited from Form)	Adds an owned form to this form.
**BeginInvoke (inherited from Control)	Overloaded. Executes the specified delegate asynchronously with the specified arguments, on the thread that the control's underlying handle was created on.
**BringToFront (inherited from Control)	Brings the control to the front of the z-order.
◆Close (inherited from Form)	Closes the form.
Contains (inherited from Control)	Retrieves a value indicating whether the specified control is a child of the control.
CreateControl (inherited from Control)	Forces the creation of the control, including the creation of the handle and any child controls.
*CreateGraphics (inherited from Control)	Creates the <u>Graphics</u> object for the control.
CreateObjRef (inherited from MarshalByRefObject)	Creates an object that contains all the relevant information required to generate a proxy used to communicate with a

	remote object.
Dispose (inherited from Component)	Overloaded. Releases all resources used by the <u>Component</u> .
DoDragDrop (inherited from Control)	Begins a drag-and-drop operation.
EndInvoke (inherited from Control)	Retrieves the return value of the asynchronous operation represented by the IAsyncResult object passed.
Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
FindForm (inherited from Control)	Retrieves the form that the control is on.
Focus (inherited from Control)	Sets input focus to the control.
from Control)	Retrieves the child control that is located at the specified coordinates.
GetContainerControl (inherited from Control)	Returns the next ContainerControl up the control's chain of parent controls.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
from MarshalByRefObject)	Retrieves the current lifetime service object that controls the lifetime policy for this instance.
from Control) (inherited	Retrieves the next control forward or back in the tab order of child controls.

GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
<u>Hide</u> (inherited from Control)	Conceals the control from the user.
**InitializeLifetimeService (inherited from MarshalByRefObject)	Obtains a lifetime service object to control the lifetime policy for this instance.
••••Invalidate (inherited from Control)	Overloaded. Invalidates the specified region of the control (adds it to the control's update region, which is the area that will be repainted at the next paint operation), and causes a paint message to be sent to the control.
<u>Nove</u> (inherited from Control)	Overloaded. Executes the specified delegate, on the thread that owns the control's underlying window handle, with the specified list of arguments.
►•• LayoutMdi (inherited from Form)	Arranges the multiple document interface (MDI) child forms within the MDI parent form.
PerformLayout (inherited from Control)	Overloaded. Forces the control to apply layout logic to all its child controls.
<u>PointToClient</u> (inherited from Control)	Computes the location of the specified screen point into client coordinates.
PointToScreen (inherited from Control)	Computes the location of the specified client point into screen coordinates.
**PreProcessMessage (inherited from Control)	Preprocesses input messages within the message loop before

	they are dispatched.
**RectangleToClient (inherited from Control)	Computes the size and location of the specified screen rectangle in client coordinates.
***RectangleToScreen (inherited from Control)	Computes the size and location of the specified client rectangle in screen coordinates.
Refresh (inherited from Control)	Forces the control to invalidate its client area and immediately redraw itself and any child controls.
RemoveOwnedForm (inherited from Form)	Removes an owned form from this form.
ResetBackColor (inherited from Control)	Resets the <u>BackColor</u> property to its default value.
ResetBindings (inherited from Control)	Resets the <u>DataBindings</u> property to its default value.
ResetCursor (inherited from Control)	Resets the <u>Cursor</u> property to its default value.
ResetFont (inherited from Control)	Resets the <u>Font</u> property to its default value.
ResetForeColor (inherited from Control)	Resets the <u>ForeColor</u> property to its default value.
ResetImeMode (inherited from Control)	Resets the <u>ImeMode</u> property to its default value.
ResetRightToLeft (inherited from Control)	Resets the RightToLeft property to its default value.
ResetText (inherited from Control)	Resets the <u>Text</u> property to its default value.
ResumeLayout (inherited from Control)	Overloaded. Resumes normal layout logic.
Scale (inherited from Control)	Overloaded. Scales the control

	and any child controls to the specified ratio.
*ScrollControlIntoView (inherited from ScrollableControl)	
Select (inherited from Control)	Overloaded. Activates the control.
**SelectNextControl (inherited from Control)	Activates the next control.
SendToBack (inherited from Control)	Sends the control to the back of the z-order.
**SetAutoScrollMargin (inherited from ScrollableControl)	Sets the size of the auto-scroll margins.
SetBounds (inherited from Control)	Overloaded. Sets the bounds of the control to the specified location and size.
**SetDesktopBounds (inherited from Form)	Sets the bounds of the form in desktop coordinates.
**SetDesktopLocation (inherited from Form)	Sets the location of the form in desktop coordinates.
<u>◆Show</u> (inherited from Control)	Displays the control to the user.
ShowDialog (inherited from Form)	Overloaded. Shows the form as a modal dialog box with no owner window.
**SuspendLayout (inherited from Control)	Temporarily suspends the layout logic for the control.
<u>NoString</u> (inherited from Form)	
<u>Update</u> (inherited from Control)	Causes the control to redraw the invalidated regions within its client area.
<u>Validate</u> (inherited from ContainerControl)	Validates the last invalidated control and its ancestors up

through, but not including, the current control.

Protected Instance Methods

*AccessibilityNotifyClients (inherited from Control)	Notifies the accessibility client applications of the specified AccessibleEvents for the specified child control.
*ActivateMdiChild (inherited from Form)	
AdjustFormScrollbars (inherited from Form)	
ApplyAutoScaling (inherited from Form)	
<u>◆CenterToParent</u> (inherited from Form)	
<u>CenterToScreen</u> (inherited from Form)	
<u>◆CreateAccessibilityInstance</u> (inherited from Control)	Creates a new accessibility object for the control.
<u>◆CreateControlsInstance</u> (inherited from Form)	
*CreateHandle (inherited from Form)	
<u>▶ DefWndProc</u> (inherited from Form)	
<u>DestroyHandle</u> (inherited from Control)	Destroys the handle associated with the control.
Dispose	Overloaded. Clean up any resources being used.
Finalize (inherited from Component)	Releases unmanaged resource and performs other cleanup operations before the

	Component is reclaimed by garbage collection.
	Returns an object that represents a service provided the Component or by its Container.
	Retrieves the value of the specified control style bit for t control.
	Determines if the control is a top-level control.
<u>InitLayout</u> (inherited from Control)	Called after the control has b added to another container.
InvokeGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event for the specified control.
№ InvokeLostFocus (inherited from Control)	Raises the <u>LostFocus</u> event the specified control.
	Raises the <u>Click</u> event for the specified control.
₱InvokePaint (inherited from Control)	Raises the Paint event for the specified control.
₱InvokePaintBackground (inherited from Control)	Raises the PaintBackgroun event for the specified contro
◆IsInputChar (inherited from Control)	Determines if a character is a input character that the contrrecognizes.
	Determines whether the specified key is a regular inpukey or a special key that requires preprocessing.

MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.
NotifyInvalidate (inherited from Control)	
<u>◆OnActivated</u> (inherited from Form)	Raises the <u>Activated</u> event.
OnBackColorChanged (inherited from Control)	Raises the <u>BackColorChange</u> event.
OnBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event.
OnBindingContextChanged (inherited from Control)	Raises the <u>BindingContextChanged</u> ever
OnCausesValidationChanged (inherited from Control)	Raises the CausesValidationChangedev
OnChangeUICues (inherited from Control)	Raises the <u>ChangeUlCues</u> event.
◆OnClick (inherited from Control)	Raises the <u>Click</u> event.
◆OnClosed (inherited from Form)	Raises the <u>Closed</u> event.
◆OnClosing (inherited from Form)	Raises the <u>Closing</u> event.
OnContextMenuChanged (inherited from Control)	Raises the <u>ContextMenuChanged</u> event.
OnControlAdded (inherited from Control)	Raises the ControlAdded eve
OnControlRemoved (inherited from ContainerControl)	
<u>◆OnCreateControl</u> (inherited from Form)	
◆OnCursorChanged (inherited from Control)	Raises the CursorChangedevent.
<u>OnDeactivate</u> (inherited from Form)	Raises the <u>Deactivate</u> event.

OnDockChanged (inherited from Control)	Raises the <u>DockChanged</u> even
OnDoubleClick (inherited from Control)	Raises the <u>DoubleClick</u> event.
OnDragDrop (inherited from Control)	Raises the <u>DragDrop</u> event.
OnDragEnter (inherited from Control)	Raises the <u>DragEnter</u> event.
OnDragLeave (inherited from Control)	Raises the <u>DragLeave</u> event.
OnDragOver (inherited from Control)	Raises the <u>DragOver</u> event.
OnEnabledChanged (inherited from Control)	Raises the <u>EnabledChanged</u> event.
OnEnter (inherited from Control)	Raises the <u>Enter</u> event.
OnFontChanged (inherited from Form)	
OnForeColorChanged (inherited from Control)	Raises the <u>ForeColorChanged</u> event.
OnGiveFeedback (inherited from Control)	Raises the <u>GiveFeedback</u> eve
OnGotFocus (inherited from Control)	Raises the <u>GotFocus</u> event.
[♠] OnHandleCreated (inherited from Form)	
MONHandleDestroyed (inherited from Form)	
OnHelpRequested (inherited from Control)	Raises the <u>HelpRequested</u> event.
OnlmeModeChanged (inherited from Control)	Raises the <u>ImeModeChanged</u> event.

OnInputLanguageChanged (inherited from Form)	Raises the InputLanguageChanged event.
◆OnInputLanguageChanging (inherited from Form)	Raises the InputLanguageChanging event
	Raises the <u>Invalidated</u> event.
	Raises the <u>KeyDown</u> event.
	Raises the <u>KeyPress</u> event.
◆OnKeyUp (inherited from Control)	Raises the <u>KeyUp</u> event.
◆OnLeave (inherited from Control)	Raises the <u>Leave</u> event.
OnLoad (inherited from Form)	Raises the <u>Load</u> event.
OnLocationChanged (inherited from Control)	Raises the <u>LocationChanged</u> event.
	Raises the <u>LostFocus</u> event.
	Raises the MaximizedBoundsChanged event.
	Raises the MaximumSizeChanged event.
OnMdiChildActivate (inherited from Form)	Raises the MdiChildActivate event.
	Raises the MenuComplete event.
	Raises the MenuStart event.
◆OnMinimumSizeChanged	Raises the

(inherited from Form)	MinimumSizeChanged event.
OnMouseDown (inherited from Control)	Raises the <u>MouseDown</u> event
<u>OnMouseEnter</u> (inherited from Control)	Raises the MouseEnter event
OnMouseHover (inherited from Control)	Raises the MouseHover even
OnMouseLeave (inherited from Control)	Raises the MouseLeave even
OnMouseMove (inherited from Control)	Raises the MouseMove event
[♠] OnMouseUp (inherited from Control)	Raises the MouseUp event.
OnMouseWheel (inherited from ScrollableControl)	
◆OnMove (inherited from Control)	Raises the <u>Move</u> event.
OnNotifyMessage (inherited from Control)	Notifies the control of Window messages.
[♠] OnPaint (inherited from Form)	
OnPaintBackground (inherited from Control)	Paints the background of the control.
<u>OnParentBackColorChanged</u> (inherited from Control)	Raises the <u>BackColorChange</u> event when the <u>BackColor</u> property value of the control's container changes.
OnParentBackgroundImageChanged (inherited from Control)	Raises the BackgroundImageChanged event when the BackgroundImage property value of the control's containe changes.
♦ OnParentBindingContextChanged	Raises the

(inherited from Control)	BindingContextChanged event when the BindingContext property value of the control's container changes.
OnParentChanged (inherited from Control)	Raises the <u>ParentChanged</u> event.
<u>OnParentEnabledChanged</u> (inherited from Control)	Raises the EnabledChanged event when the Enabled property value of the control's container changes.
	Raises the <u>FontChanged</u> even when the <u>Font</u> property value of the control's container changes
	Raises the ForeColorChanged event when the ForeColor property value of the control's container changes.
*OnParentRightToLeftChanged (inherited from Control)	Raises the RightToLeftChange event when the RightToLeft property value of the control's container changes.
<u>OnParentVisibleChanged</u> (inherited from Control)	Raises the <u>VisibleChanged</u> event when the <u>Visible</u> propert value of the control's container changes.
OnQueryContinueDrag (inherited from Control)	Raises the <u>QueryContinueDrai</u> event.
<u>OnResize</u> (inherited from Form)	
OnRightToLeftChanged (inherited from Control)	Raises the RightToLeftChangedevent.
OnSizeChanged (inherited from Control)	Raises the <u>SizeChanged</u> even

OnSystemColorsChanged (inherited from Control)	Raises the SystemColorsChanged event.
OnTablndexChanged (inherited from Control)	Raises the <u>TabIndexChanged</u> event.
OnTabStopChanged (inherited from Control)	Raises the <u>TabStopChanged</u> event.
OnTextChanged (inherited from Form)	
OnValidated (inherited from Control)	Raises the <u>Validated</u> event.
OnValidating (inherited from Control)	Raises the <u>Validating</u> event.
OnVisibleChanged (inherited from Form)	
ProcessCmdKey (inherited from Form)	
ProcessDialogChar (inherited from ContainerControl)	
ProcessDialogKey (inherited from Form)	
ProcessKeyEventArgs (inherited from Control)	Processes a key message and generates the appropriate control events.
ProcessKeyPreview (inherited from Form)	
ProcessMnemonic (inherited from ContainerControl)	
ProcessTabKey (inherited from Form)	
₱RaiseDragEvent (inherited from	

Control)	
RaiseMouseEvent (inherited from Control)	
RaisePaintEvent (inherited from Control)	
RecreateHandle (inherited from Control)	Forces the re-creation of the handle for the control.
ResetMouseEventArgs (inherited from Control)	
RtlTranslateAlignment (inherited from Control)	Overloaded. Converts the specified HorizontalAlignment t the appropriate HorizontalAlignment to support right-to-left text.
RtlTranslateContent (inherited from Control)	Converts the specified ContentAlignment to the appropriate ContentAlignment t support right-to-left text.
*RtlTranslateHorizontal (inherited from Control)	Converts the specified HorizontalAlignment to the appropriate HorizontalAlignmer to support right-to-left text.
₱RtlTranslateLeftRight (inherited from Control)	Converts the specified LeftRightAlignment to the appropriate LeftRightAlignment to support right-to-left text.
Select (inherited from Form)	Overloaded.
SetBoundsCore (inherited from Form)	

★SetClientSizeCore (inherited from Form)	
* SetDisplayRectLocation (inherited from ScrollableControl)	
★SetScrollState (inherited from ScrollableControl)	
<u>SetStyle</u> (inherited from Control)	Sets the specified style bit to th specified value.
★SetTopLevel (inherited from Control)	Sets the control as the top-leve control.
★SetVisibleCore (inherited from Form)	
<u>DupdateBounds</u> (inherited from Control)	Overloaded. Updates the bounds of the control with the current size and location.
<u>Description</u> value of the property of th	
<u>◆UpdateStyles</u> (inherited from Control)	Forces the assigned styles to b reapplied to the control.
<u>DupdateZOrder</u> (inherited from Control)	Updates the control in its parent's z-order.

Protected Internal Instance Methods

<u></u> <u>ProcessKeyMessage</u>	Processes a keyboard
(inherited from Control)	message.

Explicit Interface Implementations

IContainerControl.ActivateControl (inherited from	
ContainerControl)	

See Also

<u>TextEditorForm Class</u> | <u>NDoc.Core Namespace</u>

Copyright © 2000-2003 the NDoc Team

TextEditorForm.Dispose Method

Clean up any resources being used.

Overload List

Inherited from **Component**.

public void Dispose();

Clean up any resources being used.

protected override void Dispose(bool);

See Also

TextEditorForm Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

TextEditorForm.Dispose Method (Boolean)

Clean up any resources being used.

```
protected override void Dispose(
   bool disposing
);
```

See Also

<u>TextEditorForm Class | NDoc.Core Namespace | TextEditorForm.Dispose Overload List</u>

Copyright © 2000-2003 the NDoc Team

XmlDocumentationCache Class

Caches XML Documentation files and summaries.

For a list of all members of this type, see <u>XmlDocumentationCache</u> Members.

System.Object XmlDocumentationCache

public class XmlDocumentationCache

Requirements

Namespace: NDoc.Core

Assembly: NDoc.Core (in NDoc.Core.dll)

See Also

XmlDocumentationCache Members | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

XmlDocumentationCache Members

XmlDocumentationCache overview

Public Instance Constructors

XmlDocumentationCache	Initializes a new instance of the
Constructor	XmlDocumentationCache class.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified <u>Object</u> is equal to the current <u>Object</u> .
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetSummary	Returns the original summary for a member inherited from a specified type.
GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
SetXmlFor	Gets the xml documentation for the assembly of the specified type.
ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
<u>MemberwiseClone</u> (inherited	Creates a shallow copy of the

•	_	• • • •
from	/ 1h	100t)
11()11		
11 0111	\mathbf{O}	

current Object.

See Also

XmlDocumentationCache Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

XmlDocumentationCache Constructor

Initializes a new instance of the XmlDocumentationCache class.

public XmlDocumentationCache();

See Also

XmlDocumentationCache Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

XmlDocumentationCache Methods

The methods of the **XmlDocumentationCache** class are listed below. For a complete list of **XmlDocumentationCache** class members, see the <u>XmlDocumentationCache Members</u> topic.

Public Instance Methods

Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object.
GetHashCode (inherited from Object)	Serves as a hash function for a particular type, suitable for use in hashing algorithms and data structures like a hash table.
GetSummary	Returns the original summary for a member inherited from a specified type.
● GetType (inherited from Object)	Gets the <u>Type</u> of the current instance.
* ◆GetXmlFor	Gets the xml documentation for the assembly of the specified type.
◆ ToString (inherited from Object)	Returns a <u>String</u> that represents the current <u>Object</u> .

Protected Instance Methods

Finalize (inherited from Object)	Allows an Object to attempt to free resources and perform other cleanup operations before the Object is reclaimed by garbage collection.
MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object.

See Also

XmlDocumentationCache Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team Generated from assembly NDoc.Core [1.2.1300.31138]

XmlDocumentationCache.GetSummary Method

Returns the original summary for a member inherited from a specified type.

```
public XmlNode GetSummary(
    string memberID,
    Type declaringType
);
```

Parameters

memberID

The member ID to lookup.

declaringType

The type that declares that member.

Return Value

The summary xml node. If not found, returns an empty node.

See Also

XmlDocumentationCache Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team

XmlDocumentationCache.GetXmlFor Method

Gets the xml documentation for the assembly of the specified type.

```
public XmlDocument GetXmlFor(
    Type type
);
```

Return Value

The xml document. If the xml file was not found, returns an empty document.

See Also

XmlDocumentationCache Class | NDoc.Core Namespace

Copyright © 2000-2003 the NDoc Team