

NDde Namespace

This namespace contains classes for using Dynamic Data Exchange (DDE) in .NET.

[Namespace hierarchy](#)

Classes

Class	Description
DdeEventArgs	This is the base class for all NDde event argument classes.
DdeException	This is thrown when a DDE exception occurs.

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

DdeEventArgs Class

This is the base class for all NDde event argument classes.

For a list of all members of this type, see [DdeEventArgs Members](#).

[System.Object](#) EventArgs

DdeEventArgs

[DdeRegistrationEventArgs](#)

[DdeActivityEventArgs](#)

[DdeAdviseEventArgs](#)

[DdeDisconnectedEventArgs](#)

```
public class DdeEventArgs : EventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeEventArgs Members](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeEventArgs Members

[DdeEventArgs overview](#)

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	This returns a string containing the current values of all properties.

Protected Instance Constructors

 DdeEventArgs Constructor	Initializes a new instance of the DdeEventArgs class.
--	---

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

[DdeEventArgs Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeEventArgs Constructor

Initializes a new instance of the [DdeEventArgs](#) class.

```
protected DdeEventArgs();
```

See Also

[DdeEventArgs Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeEventArgs Methods

The methods of the **DdeEventArgs** class are listed below. For a complete list of **DdeEventArgs** class members, see the [DdeEventArgs Members](#) 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	This returns a string containing the current values of all properties.

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

[DdeEventArgs Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeEventArgs.ToString Method

This returns a string containing the current values of all properties.

```
public override string ToString();
```

Return Value

A string containing the current values of all properties.

See Also

[DdeEventArgs Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException Class

This is thrown when a DDE exception occurs.

For a list of all members of this type, see [DdeException Members](#).

[System.Object](#) Exception
DdeException

```
public class DdeException : Exception
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeException Members](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException Members

[DdeException overview](#)

Public Instance Properties

 Code	This gets an error code returned by the DDEML.
 Data (inherited from Exception)	Gets a collection of key/value pairs that provide additional, user-defined information about the exception.
 HelpLink (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.
 Message (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.
 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 .
 GetBaseException (inherited)	When overridden in a derived

from Exception)	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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetObjectData	
 GetType (inherited from Exception)	Gets the runtime type of the current instance.
 ToString (inherited from Exception)	Creates and returns a string representation of the current exception.

Protected Instance Constructors

 DdeException Constructor	
---	--

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.
 MemberwiseClone (inherited from Object)	Creates a shallow copy of the current Object .

See Also

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException Constructor

```
protected DdeException(  
    SerializationInfo info,  
    StreamingContext context  
);
```

Parameters

info

context

See Also

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException Properties

The properties of the **DdeException** class are listed below. For a complete list of **DdeException** class members, see the [DdeException Members](#) topic.

Public Instance Properties

 Code	This gets an error code returned by the DDEML.
 Data (inherited from Exception)	Gets a collection of key/value pairs that provide additional, user-defined information about the exception.
 HelpLink (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.
 Message (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.
 TargetSite (inherited from Exception)	Gets the method that throws 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.

See Also

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException.Code Property

This gets an error code returned by the DDEML.

```
public Int32 Code {get;}
```

Remarks

The value is zero if the exception was not thrown because of the DDEML.

- **0x0000** - DMLERR_NO_DMLERROR
- **0x4000** - DMLERR_ADVACKTIMEOUT
- **0x4001** - DMLERR_BUSY
- **0x4002** - DMLERR_DATAACKTIMEOUT
- **0x4003** - DMLERR_DLL_NOT_INITIALIZED
- **0x4004** - DMLERR_DLL_USAGE
- **0x4005** - DMLERR_EXECACKTIMEOUT
- **0x4006** - DMLERR_INVALIDPARAMETER
- **0x4007** - DMLERR_LOW_MEMORY
- **0x4008** - DMLERR_MEMORY_DMLERROR
- **0x4009** - DMLERR_NOTPROCESSED
- **0x400A** - DMLERR_NO_CONV_ESTABLISHED
- **0x400B** - DMLERR_POKEACKTIMEOUT
- **0x400C** - DMLERR_POSTMSG_FAILED
- **0x400D** - DMLERR_REENTRANCY
- **0x400E** - DMLERR_SERVER_DIED
- **0x400F** - DMLERR_SYS_DMLERROR
- **0x4010** - DMLERR_UNADVACKTIMEOUT
- **0x4011** - DMLERR_UNFOUND_QUEUE_ID

See Also

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException Methods

The methods of the **DdeException** class are listed below. For a complete list of **DdeException** class members, see the [DdeException Members](#) topic.

Public Instance Methods

 Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
 GetBaseException (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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetObjectData	
 GetType (inherited from Exception)	Gets the runtime type of the current instance.
 ToString (inherited from Exception)	Creates and returns a string representation of the current 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

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeException.GetObjectData Method

```
public override void GetObjectData(  
    SerializationInfo info,  
    StreamingContext context  
);
```

Parameters

info

context

Implements

[ISerializable](#).

See Also

[DdeException Class](#) | [NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

NDde.Advanced Namespace

This namespace contains classes for using advanced features of the library.

[Namespace hierarchy](#)

Classes

Class	Description
DdeContext	This provides an execution context for <code>DdeClient</code> and <code>DdeServer</code> .
DdeMessageLoop	This is a synchronizing object that can run a message loop on any thread.
DdeRegistrationEventArgs	This contains information about the <code>Register</code> and <code>Unregister</code> events.
DdeTransaction	This contains the parameters of the DDEML callback function.

Interfaces

Interface	Description
IDdeTransactionFilter	This defines a transaction filter.

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

DdeContext Class

This provides an execution context for [DdeClient](#) and [DdeServer](#).

For a list of all members of this type, see [DdeContext Members](#).

[System.Object](#) **DdeContext**

```
public class DdeContext : IDisposable,  
    ISynchronizeInvoke
```

Thread Safety

This type is safe for multithreaded operations.

Remarks

This class provides a context for DDE activity. All [DdeClient](#) and [DdeServer](#) objects must be associated with an instance of this class. If one is not specified in their constructors then a default instance of this class is used. This class must be initialized before it can begin sending and receiving DDE messages. This happens automatically upon its first use by a [DdeClient](#) or [DdeServer](#). An application can call [Initialize](#) to make the initialization process occur immediately. This is useful when a calling application expects this class to raise the [Register](#) and [Unregister](#) events or invoke the [ITransactionFilter.PreFilterTransaction](#) method before being used by a [DdeClient](#) or [DdeServer](#).

Since forms and controls implement [ISynchronizeInvoke](#) they can be used as the synchronizing object for this class. When an instance of this class is created to use a form or control as the synchronizing object it will use the UI thread for execution. This is the preferred way of creating an instance of this class when used in a windows application since it avoids multithreaded synchronization issues and cross thread marshaling. When an instance of this class is created without specifying a synchronizing object it will create and manage its own thread for execution. This is convenient if you wish to use this library in a console or service application, but with the

added cost of cross thread marshaling and the potential for deadlocking application threads.

Events are invoked on the thread hosting the `DdeContext`. All operations must be marshaled onto the thread hosting the `DdeContext`. Method calls will block until that thread becomes available. An exception will be generated if the thread does not become available in a timely manner.

Example

The following example demonstrates how to instantiate a `DdeContext` in a console application.

```
[C#]
using System;
using NDde.Advanced;

public class Example
{
    public static void Main()
    {
        // Create a context that uses a dedicated th
        DdeContext context = new DdeContext();
    }
}
```

```
[Visual Basic]
Imports NDde.Advanced

Public Class Example
    Public Shared Sub Main()
        ' Create a context that uses a d
        Dim context As DdeContext = New
    End Sub
End Class
```

The following example demonstrates how to instantiate a `DdeContext` in a windows application.

```

[C#]
using System;
using NDde.Advanced;

public class Example : Form
{
    // Standard Form code omitted for brevity.

    private DdeContext context = null;

    private void Form1_Load(object sender, System.EventArgs e)
    {
        // Create a context that uses the UI thread
        context = new DdeContext(this);
    }
}

```

```

[Visual Basic]
Imports NDde.Advanced

Public Class Example
    Inherits Form

    Private context as DdeContext = Nothing

    Private Sub Form1_Load(ByVal sender As Object, ByVal e As EventArgs)
        ' Create a context that uses the UI thread
        context = New DdeContext(Me)
    End Sub
End Class

```

Requirements

Namespace: [NDde.Advanced](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeContext Members](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Members

[DdeContext overview](#)

Public Instance Constructors

 DdeContext	Overloaded.
--	-------------

Public Instance Properties

 Encoding	This gets or sets the default encoding that is used.
 InstanceId	This gets the DDEML instance identifier.
 InvokeRequired	This gets a bool indicating whether the caller must use Invoke.
 IsInitialized	This gets a bool indicating whether the context is initialized.

Public Instance Methods

 AddTransactionFilter	This adds a transaction filter to monitor DDE transactions.
 BeginInvoke	This begins an asynchronous operation to execute a delegate on the thread hosting this object.
 Dispose	This releases all resources held by this instance.
 EndInvoke	This returns the object that the delegate returned in the operation.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 Initialize	This initializes the context.
 Invoke	This executes a delegate on the thread hosting this object.
 RemoveTransactionFilter	This removes a transaction filter and stops it from monitoring DDE transactions.
 ToString (inherited from Object)	Returns a String that represents the current Object .

Public Instance Events

 Register	This is raised when a service name has been registered by a server using the DDEML.
 Unregister	This is raised when a service name has been unregistered by a server using the DDEML.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Constructor

Overload List

This initializes a new instance of the `DdeContext` class that uses a dedicated thread for execution.

[public DdeContext\(\);](#)

This initializes a new instance of the `DdeContext` class that uses the specified synchronizing object for execution.

[public DdeContext\(ISynchronizeInvoke\);](#)

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Constructor ()

This initializes a new instance of the `DdeContext` class that uses a dedicated thread for execution.

```
public DdeContext();
```

Remarks

This constructor is used when you want the context to create and manage its own thread for DDE message pumping.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#) | [DdeContext Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Constructor (ISynchronizeInvoke)

This initializes a new instance of the `DdeContext` class that uses the specified synchronizing object for execution.

```
public DdeContext(  
    ISynchronizeInvoke synchronizingObject  
);
```

Parameters

synchronizingObject

The synchronizing object to use for execution.

Remarks

This constructor is used when you want the context to use the specified synchronizing object for DDE message pumping. Since forms and controls implement `ISynchronizeInvoke` they can be used as the synchronizing object. In that case the windows application UI thread that is hosting the form or control is used.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when synchronizer is a null reference.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#) | [DdeContext Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Properties

The properties of the **DdeContext** class are listed below. For a complete list of **DdeContext** class members, see the [DdeContext Members](#) topic.

Public Instance Properties

 Encoding	This gets or sets the default encoding that is used.
 Instanceld	This gets the DDEML instance identifier.
 InvokeRequired	This gets a bool indicating whether the caller must use Invoke.
 IsInitialized	This gets a bool indicating whether the context is initialized.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.Encoding Property

This gets or sets the default encoding that is used.

```
public Encoding Encoding {get; set;}
```

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.InstanceId Property

This gets the DDEML instance identifier.

```
public Int32 InstanceId {get;}
```

Remarks

This can be used in any DDEML function requiring an instance identifier.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.InvokeRequired Property

This gets a bool indicating whether the caller must use Invoke.

```
public Boolean InvokeRequired {get;}
```

Implements

[ISynchronizeInvoke.](#)

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.IsInitialized Property

This gets a bool indicating whether the context is initialized.

```
public Boolean IsInitialized {get;}
```

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Methods

The methods of the **DdeContext** class are listed below. For a complete list of **DdeContext** class members, see the [DdeContext Members](#) topic.

Public Instance Methods

 AddTransactionFilter	This adds a transaction filter to monitor DDE transactions.
 BeginInvoke	This begins an asynchronous operation to execute a delegate on the thread hosting this object.
 Dispose	This releases all resources held by this instance.
 EndInvoke	This returns the object that the delegate returned in the operation.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 Initialize	This initializes the context.
 Invoke	This executes a delegate on the thread hosting this object.
 RemoveTransactionFilter	This removes a transaction filter and stops it from monitoring DDE transactions.

 [ToString](#) (inherited from **Object**)

Returns a [String](#) that represents the current [Object](#).

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.AddTransactionFilter Method

This adds a transaction filter to monitor DDE transactions.

```
public void AddTransactionFilter(  
    IDdeTransactionFilter filter  
);
```

Parameters

filter

The implementation of `ITransactionFilter` that you want to add.

Remarks

Transaction filters can be used to intercept the DDEML callback.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when filter is a null reference.
InvalidOperationException	This is thrown when the filter was already added.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.BeginInvoke Method

This begins an asynchronous operation to execute a delegate on the thread hosting this object.

```
public IAsyncResult BeginInvoke(  
    Delegate method,  
    Object\[\] args  
);
```

Parameters

method

The delegate to execute.

args

The arguments to pass to the delegate.

Return Value

An [IAsyncResult](#) object for this operation.

Implements

[ISynchronizeInvoke](#).

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.Dispose Method

This releases all resources held by this instance.

```
public void Dispose();
```

Implements

[IDisposable.](#)

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.EndInvoke Method

This returns the object that the delegate returned in the operation.

```
public object EndInvoke(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The `IAsyncResult` object returned by a call to `BeginInvoke`.

Return Value

The object returned by the delegate.

Implements

[ISynchronizeInvoke](#).

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.Initialize Method

This initializes the context.

```
public void Initialize();
```

Remarks

This class must be initialized before it can begin sending and receiving DDE messages. This happens automatically upon its first use by a [DdeClient](#) or [DdeServer](#). An application can call [Initialize](#) to make the initialization process occur immediately. This is useful when a calling application expects this class to raise the [Register](#) and [Unregister](#) events or invoke the [ITransactionFilter.PreFilterTransaction](#) method before being used by a [DdeClient](#) or [DdeServer](#).

If you attempt to use a synchronizer that is not hosted on a thread running a windows message loop an exception will be thrown.

Explicitly calling this method will allow added [ITransactionFilter](#) objects to begin intercepting the DDEML callback function.

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the context is already initialized.
DdeException	This is thrown when the context could not be initialized.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.Invoke Method

This executes a delegate on the thread hosting this object.

```
public object Invoke(  
    Delegate method,  
    Object\[\] args  
);
```

Parameters

method

The delegate to execute.

args

The arguments to pass to the delegate.

Return Value

The object returned by the delegate.

Implements

[ISynchronizeInvoke](#).

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.RemoveTransactionFilter Method

This removes a transaction filter and stops it from monitoring DDE transactions.

```
public void RemoveTransactionFilter(  
    IDdeTransactionFilter filter  
);
```

Parameters

filter

The implementation of `ITransactionFilter` that you want to remove.

Remarks

Transaction filters can be used to intercept the DDEML callback.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when filter is a null reference.
InvalidOperationException	This is thrown when the filter was not previously added.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext Events

The events of the **DdeContext** class are listed below. For a complete list of **DdeContext** class members, see the [DdeContext Members](#) topic.

Public Instance Events

 Register	This is raised when a service name has been registered by a server using the DDEML.
 Unregister	This is raised when a service name has been unregistered by a server using the DDEML.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.EventHandler<TEventArgs> Event

This is raised when a service name has been registered by a server using the DDEML.

```
public event EventHandler<TEventArgs> Registered
```

Remarks

This event will not be raised by servers that do not use the DDEML.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeContext.EventHandler<TEventArgs> Event

This is raised when a service name has been unregistered by a server using the DDEML.

```
public event EventHandler<TEventArgs> Unregist
```

Remarks

This event will not be raised by servers that do not use the DDEML.

See Also

[DdeContext Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop Class

This is a synchronizing object that can run a message loop on any thread.

For a list of all members of this type, see [DdeMessageLoop Members](#).

[System.Object](#) **DdeMessageLoop**

```
public class DdeMessageLoop : IDisposable,  
    ISynchronizeInvoke
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeMessageLoop Members](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop Members

[DdeMessageLoop overview](#)

Public Instance Constructors

 DdeMessageLoop Constructor	This initializes a new instance of the DdeMessageLoop class.
--	--

Public Instance Methods

 Dispose	This releases all resources held by this instance.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 Run	Overloaded. This starts a message loop on the current thread.
 ToString (inherited from Object)	Returns a String that represents the current Object .

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop Constructor

This initializes a new instance of the `DdeMessageLoop` class.

```
public DdeMessageLoop( );
```

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop Methods

The methods of the **DdeMessageLoop** class are listed below. For a complete list of **DdeMessageLoop** class members, see the [DdeMessageLoop Members](#) topic.

Public Instance Methods

 Dispose	This releases all resources held by this instance.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 Run	Overloaded. This starts a message loop on the current thread.
 ToString (inherited from Object)	Returns a String that represents the current Object .

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop.Dispose Method

This releases all resources held by this instance.

```
public void Dispose();
```

Implements

[IDisposable.](#)

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop.Run Method

This starts a message loop on the current thread.

Overload List

This starts a message loop on the current thread.

[public void Run\(\);](#)

This starts a message loop on the current thread and shows the specified form.

[public void Run\(Form\);](#)

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop.Run Method ()

This starts a message loop on the current thread.

```
public void Run();
```

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#) |
[DdeMessageLoop.Run Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageLoop.Run Method (Form)

This starts a message loop on the current thread and shows the specified form.

```
public void Run(  
    Form form  
);
```

Parameters

form

The Form to display.

See Also

[DdeMessageLoop Class](#) | [NDde.Advanced Namespace](#) |
[DdeMessageLoop.Run Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeRegistrationEventArgs Class

This contains information about the `Register` and `Unregister` events.

For a list of all members of this type, see [DdeRegistrationEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 DdeRegistrationEventArgs

```
public class DdeRegistrationEventArgs :  
    DdeEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeRegistrationEventArgs Members](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeRegistrationEventArgs Members

[DdeRegistrationEventArgs overview](#)

Public Instance Properties

 Service	This gets the service name associated with this event.
---	--

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeRegistrationEventArgs Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeRegistrationEventArgs Properties

The properties of the **DdeRegistrationEventArgs** class are listed below. For a complete list of **DdeRegistrationEventArgs** class members, see the [DdeRegistrationEventArgs Members](#) topic.

Public Instance Properties

 Service	This gets the service name associated with this event.
---	--

See Also

[DdeRegistrationEventArgs Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeRegistrationEventArgs.Service Property

This gets the service name associated with this event.

```
public String Service {get;}
```

See Also

[DdeRegistrationEventArgs Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction Class

This contains the parameters of the DDEML callback function.

For a list of all members of this type, see [DdeTransaction Members](#).

[System.Object](#) **DdeTransaction**

```
public class DdeTransaction
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Remarks

The `dwRet` property contains the value returned by the DDEML callback function and is the only member that can be modified. See the MSDN documentation for more information about the members of this class.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

Requirements

Namespace: [NDde.Advanced](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeTransaction Members](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction Members

[DdeTransaction overview](#)

Public Instance Properties

 dwData1	See the MSDN documentation for information about this member.
 dwData2	See the MSDN documentation for information about this member.
 dwRet	This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.
 hConv	See the MSDN documentation for information about this member.
 hData	See the MSDN documentation for information about this member.
 hsz1	See the MSDN documentation for information about this member.
 hsz2	See the MSDN documentation for information about this member.
 uFmt	See the MSDN documentation for information about this member.
 uType	See the MSDN documentation for information about this member.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString	

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction Properties

The properties of the **DdeTransaction** class are listed below. For a complete list of **DdeTransaction** class members, see the [DdeTransaction Members](#) topic.

Public Instance Properties

 dwData1	See the MSDN documentation for information about this member.
 dwData2	See the MSDN documentation for information about this member.
 dwRet	This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.
 hConv	See the MSDN documentation for information about this member.
 hData	See the MSDN documentation for information about this member.
 hsz1	See the MSDN documentation for information about this member.
 hsz2	See the MSDN documentation for information about this member.
 uFmt	See the MSDN documentation for information about this member.
 uType	See the MSDN documentation for information about this

member.

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.dwData1 Property

See the MSDN documentation for information about this member.

```
public IntPtr dwData1 {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.dwData2 Property

See the MSDN documentation for information about this member.

```
public IntPtr dwData2 {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.dwRet Property

This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.

```
public IntPtr dwRet {get; set;}
```

Remarks

This will be ignored if the PreFilterTransaction method returns false.

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.hConv Property

See the MSDN documentation for information about this member.

```
public IntPtr hConv {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.hData Property

See the MSDN documentation for information about this member.

```
public IntPtr hData {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.hsz1 Property

See the MSDN documentation for information about this member.

```
public IntPtr hsz1 {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.hsz2 Property

See the MSDN documentation for information about this member.

```
public IntPtr hsz2 {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.uFmt Property

See the MSDN documentation for information about this member.

```
public Int32 uFmt {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.uType Property

See the MSDN documentation for information about this member.

```
public Int32 uType {get;}
```

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction Methods

The methods of the **DdeTransaction** class are listed below. For a complete list of **DdeTransaction** class members, see the [DdeTransaction Members](#) 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString	

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeTransaction.ToString Method

```
public override string ToString();
```

Return Value

See Also

[DdeTransaction Class](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

IDdeTransactionFilter Interface

This defines a transaction filter.

For a list of all members of this type, see [IDdeTransactionFilter Members](#).

```
public interface IDdeTransactionFilter
```

Remarks

Use a transaction filter to intercept the DDEML callback function. The `PreFilterTransaction` method will be called every time the DDEML callback function executes. The `Transaction` object passed into the method contains the parameters of the DDE callback function. By using a transaction filter the developer has complete control over the DDEML. See the MSDN documentation for more information on using the DDEML.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

Requirements

Namespace: [NDde.Advanced](#)

Assembly: NDde (in NDde.dll)

See Also

[IDdeTransactionFilter Members](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

IDdeTransactionFilter Members

[IDdeTransactionFilter overview](#)

Public Instance Methods

 PreFilterTransaction	This filters a transaction before it is dispatched.
--	---

See Also

[IDdeTransactionFilter Interface](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

IDdeTransactionFilter Methods

The methods of the **IDdeTransactionFilter** interface are listed below. For a complete list of **IDdeTransactionFilter** interface members, see the [IDdeTransactionFilter Members](#) topic.

Public Instance Methods

 PreFilterTransaction	This filters a transaction before it is dispatched.
--	---

See Also

[IDdeTransactionFilter Interface](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

IDdeTransactionFilter.PreFilterTransaction Method

This filters a transaction before it is dispatched.

```
bool PreFilterTransaction(  
    DdeTransaction t  
);
```

Parameters

t

The transaction to be dispatched.

Return Value

True to filter the transaction and stop it from being dispatched, false otherwise.

Remarks

This method is called everytime the DDEML callback function executes.

CAUTION Incorrect usage of the DDEML can cause this library to function incorrectly and can lead to resource leaks.

See Also

[IDdeTransactionFilter Interface](#) | [NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

NDde.Advanced.Monitor Namespace

This namespace contains classes for creating DDE monitors.

[Namespace hierarchy](#)

Classes

Class	Description
DdeActivityEventArgs	This contains information about events on <code>DdeMonitor</code> .
DdeCallbackActivityEventArgs	This contains information about the <code>CallbackActivity</code> event.
DdeConversationActivityEventArgs	This contains information about the <code>ConversationActivity</code> event.
DdeErrorActivityEventArgs	This contains information about the <code>ErrorActivity</code> event.
DdeLinkActivityEventArgs	This contains information about the <code>LinkActivity</code> event.
DdeMessageActivityEventArgs	This contains information about the <code>MessageActivity</code> event.
DdeMonitor	This is used to monitor DDE activity.

Enumerations

Enumeration	Description
DdeMessageActivityKind	This represents the kind of message

	contained in <code>DdeMessageActivityEventArgs</code> .
DdeMonitorFlags	This specifies the different kinds of DDE activity that can be monitored.

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

DdeActivityEventArgs Class

This contains information about events on `DdeMonitor`.

For a list of all members of this type, see [DdeActivityEventArgs Members](#).

[System.Object](#) EventArgs

[DdeEventArgs](#)

DdeActivityEventArgs

[DdeCallbackActivityEventArgs](#)

[DdeConversationActivityEventArgs](#)

[DdeErrorActivityEventArgs](#)

[DdeLinkActivityEventArgs](#)

[DdeMessageActivityEventArgs](#)

```
public class DdeActivityEventArgs :  
    DdeEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeActivityEventArgs Members

[DdeActivityEventArgs overview](#)

Public Instance Properties

 TaskHandle	This gets the task handle of the application associated with this event.
--	--

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

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

[DdeActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeActivityEventArgs Properties

The properties of the **DdeActivityEventArgs** class are listed below. For a complete list of **DdeActivityEventArgs** class members, see the [DdeActivityEventArgs Members](#) topic.

Public Instance Properties

 TaskHandle	This gets the task handle of the application associated with this event.
--	--

See Also

[DdeActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeActivityEventArgs.TaskHandle Property

This gets the task handle of the application associated with this event.

```
public IntPtr TaskHandle {get;}
```

See Also

[DdeActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs Class

This contains information about the [CallbackActivity](#) event.

For a list of all members of this type, see [DdeCallbackActivityEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 [DdeActivityEventArgs](#)
 DdeCallbackActivityEventArgs

```
public class DdeCallbackActivityEventArgs
: DdeActivityEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeCallbackActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs Members

[DdeCallbackActivityEventArgs overview](#)

Public Instance Properties

 dwData1	See the MSDN documentation for information about this member.
 dwData2	See the MSDN documentation for information about this member.
 dwRet	This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.
 hConv	See the MSDN documentation for information about this member.
 hData	See the MSDN documentation for information about this member.
 hsz1	See the MSDN documentation for information about this member.
 hsz2	See the MSDN documentation for information about this member.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 uFmt	See the MSDN documentation for information about this member.
 uType	See the MSDN documentation

for information about this member.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs Properties

The properties of the **DdeCallbackActivityEventArgs** class are listed below. For a complete list of **DdeCallbackActivityEventArgs** class members, see the [DdeCallbackActivityEventArgs Members](#) topic.

Public Instance Properties

 dwData1	See the MSDN documentation for information about this member.
 dwData2	See the MSDN documentation for information about this member.
 dwRet	This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.
 hConv	See the MSDN documentation for information about this member.
 hData	See the MSDN documentation for information about this member.
 hsz1	See the MSDN documentation for information about this member.
 hsz2	See the MSDN documentation for information about this member.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 uFmt	See the MSDN documentation

	for information about this member.
 uType	See the MSDN documentation for information about this member.

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.dwData1 Property

See the MSDN documentation for information about this member.

```
public IntPtr dwData1 {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.dwData2 Property

See the MSDN documentation for information about this member.

```
public IntPtr dwData2 {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.dwRet Property

This gets the return value of the DDEML callback function. See the MSDN documentation for information about this member.

```
public IntPtr dwRet {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.hConv Property

See the MSDN documentation for information about this member.

```
public IntPtr hConv {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.hData Property

See the MSDN documentation for information about this member.

```
public IntPtr hData {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.hsz1 Property

See the MSDN documentation for information about this member.

```
public IntPtr hsz1 {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.hsz2 Property

See the MSDN documentation for information about this member.

```
public IntPtr hsz2 {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.uFmt Property

See the MSDN documentation for information about this member.

```
public Int32 uFmt {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeCallbackActivityEventArgs.uType Property

See the MSDN documentation for information about this member.

```
public Int32 uType {get;}
```

See Also

[DdeCallbackActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs Class

This contains information about the `ConversationActivity` event.

For a list of all members of this type, see [DdeConversationActivityEventArgs Members](#).

[System.Object](#) EventArgs

[DdeEventArgs](#)

[DdeActivityEventArgs](#)

DdeConversationActivityEventArgs

```
public class DdeConversationActivityEventArgs : DdeActivityEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeConversationActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs Members

[DdeConversationActivityEventArgs overview](#)

Public Instance Properties

 ClientHandle	This gets the handle to the client application associated with the conversation.
 IsEstablished	This gets a bool indicating whether the conversation is being established.
 ServerHandle	This gets the handle to the server application associated with the conversation.
 Service	This gets the service name associated with the conversation.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 Topic	This gets the topic name associated with the conversation.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from	Gets the Type of the current

Object)	instance.
ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs Properties

The properties of the **DdeConversationActivityEventArgs** class are listed below. For a complete list of **DdeConversationActivityEventArgs** class members, see the [DdeConversationActivityEventArgs Members](#) topic.

Public Instance Properties

 ClientHandle	This gets the handle to the client application associated with the conversation.
 IsEstablished	This gets a bool indicating whether the conversation is being established.
 ServerHandle	This gets the handle to the server application associated with the conversation.
 Service	This gets the service name associated with the conversation.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 Topic	This gets the topic name associated with the conversation.

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs.ClientHandle Property

This gets the handle to the client application associated with the conversation.

```
public IntPtr ClientHandle {get;}
```

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs.IsEstablished Property

This gets a bool indicating whether the conversation is being established.

```
public Boolean IsEstablished {get;}
```

Remarks

The value returned by this property will be true if the conversation is being established. If the conversation is being terminated then the value will be false.

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs.ServerHandle Property

This gets the handle to the server application associated with the conversation.

```
public IntPtr ServerHandle {get;}
```

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationActivityEventArgs.Service Property

This gets the service name associated with the conversation.

```
public String Service {get;}
```

See Also

[DdeConversationActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversationEventArgs.Topic Property

This gets the topic name associated with the conversation.

```
public String Topic {get;}
```

See Also

[DdeConversationEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeErrorActivityEventArgs Class

This contains information about the `ErrorActivity` event.

For a list of all members of this type, see [DdeErrorActivityEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 [DdeActivityEventArgs](#)
 DdeErrorActivityEventArgs

```
public class DdeErrorActivityEventArgs :  
    DdeActivityEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeErrorActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeErrorActivityEventArgs Members

[DdeErrorActivityEventArgs overview](#)

Public Instance Properties

 Code	This gets an error code returned by the DDEML.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeErrorActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeErrorActivityEventArgs Properties

The properties of the **DdeErrorActivityEventArgs** class are listed below. For a complete list of **DdeErrorActivityEventArgs** class members, see the [DdeErrorActivityEventArgs Members](#) topic.

Public Instance Properties

 Code	This gets an error code returned by the DDEML.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.

See Also

[DdeErrorActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeErrorActivityEventArgs.Code Property

This gets an error code returned by the DDEML.

```
public Int32 Code {get;}
```

See Also

[DdeErrorActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs Class

This contains information about the `LinkActivity` event.

For a list of all members of this type, see [DdeLinkActivityEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 [DdeActivityEventArgs](#)
 DdeLinkActivityEventArgs

```
public class DdeLinkActivityEventArgs :  
    DdeActivityEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeLinkActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs Members

[DdeLinkActivityEventArgs overview](#)

Public Instance Properties

 ClientHandle	This gets the handle to the client application associated with the link.
 Format	This gets the format of the data associated with the link.
 IsEstablished	This gets a bool indicating whether the link is being established.
 IsHot	This gets a bool indicating whether the link is hot.
 IsServerInitiated	This gets a bool indicating whether the link was terminated by the server.
 Item	This gets the item name associated with the link.
 ServerHandle	This gets the handle to the server application associated with the link.
 Service	This gets the service name associated with the link.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 Topic	This gets the topic name associated with the link.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
◆ GetType (inherited from Object)	Gets the Type of the current instance.
◆ ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs Properties

The properties of the **DdeLinkActivityEventArgs** class are listed below. For a complete list of **DdeLinkActivityEventArgs** class members, see the [DdeLinkActivityEventArgs Members](#) topic.

Public Instance Properties

 ClientHandle	This gets the handle to the client application associated with the link.
 Format	This gets the format of the data associated with the link.
 IsEstablished	This gets a bool indicating whether the link is being established.
 IsHot	This gets a bool indicating whether the link is hot.
 IsServerInitiated	This gets a bool indicating whether the link was terminated by the server.
 Item	This gets the item name associated with the link.
 ServerHandle	This gets the handle to the server application associated with the link.
 Service	This gets the service name associated with the link.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.
 Topic	This gets the topic name associated with the link.

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.ClientHandle Property

This gets the handle to the client application associated with the link.

```
public IntPtr ClientHandle {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.Format Property

This gets the format of the data associated with the link.

```
public Int32 Format {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.IsEstablished Property

This gets a bool indicating whether the link is being established.

```
public Boolean IsEstablished {get;}
```

Remarks

The value returned by this property will be true if the conversation is being established. If the conversation is being terminated then the value will be false.

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.IsHot Property

This gets a bool indicating whether the link is hot.

```
public Boolean IsHot {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.IsServerInitiated Property

This gets a bool indicating whether the link was terminated by the server.

```
public Boolean IsServerInitiated {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.Item Property

This gets the item name associated with the link.

```
public String Item {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.ServerHandle Property

This gets the handle to the server application associated with the link.

```
public IntPtr ServerHandle {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.Service Property

This gets the service name associated with the link.

```
public String Service {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeLinkActivityEventArgs.Topic Property

This gets the topic name associated with the link.

```
public String Topic {get;}
```

See Also

[DdeLinkActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityEventArgs Class

This contains information about the `MessageActivity` event.

For a list of all members of this type, see [DdeMessageActivityEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 [DdeActivityEventArgs](#)
 DdeMessageActivityEventArgs

```
public class DdeMessageActivityEventArgs :  
    DdeActivityEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeMessageActivityEventArgs Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityEventArgs Members

[DdeMessageActivityEventArgs overview](#)

Public Instance Properties

 Kind	This gets the kind of message associated with this event.
 Message	This gets the message associated with this event.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeMessageActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityEventArgs Properties

The properties of the **DdeMessageActivityEventArgs** class are listed below. For a complete list of **DdeMessageActivityEventArgs** class members, see the [DdeMessageActivityEventArgs Members](#) topic.

Public Instance Properties

 Kind	This gets the kind of message associated with this event.
 Message	This gets the message associated with this event.
 TaskHandle (inherited from DdeActivityEventArgs)	This gets the task handle of the application associated with this event.

See Also

[DdeMessageActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityEventArgs.Kind Property

This gets the kind of message associated with this event.

```
public DdeMessageActivityKind Kind {get;}
```

See Also

[DdeMessageActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityEventArgs.Message Property

This gets the message associated with this event.

```
public Message Message {get;}
```

See Also

[DdeMessageActivityEventArgs Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMessageActivityKind Enumeration

This represents the kind of message contained in `DdeMessageActivityEventArgs`.

```
public enum DdeMessageActivityKind
```

Members

Member Name	Description
Post	The message was posted by a DDE application.
Send	The message was sent by a DDE application.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Class

This is used to monitor DDE activity.

For a list of all members of this type, see [DdeMonitor Members](#).

[System.Object](#) **DdeMonitor**

```
public class DdeMonitor : IDisposable
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeMonitor Members](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Members

[DdeMonitor overview](#)

Public Instance Constructors

 DdeMonitor Constructor	This initializes a new instance of the <code>DdeMonitor</code> class.
--	---

Public Instance Properties

 Context	This gets the context associated with this instance.
---	--

Public Instance Methods

 Dispose	This releases all resources held by this instance.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 Start	This starts monitoring the system for DDE activity.
 ToString (inherited from Object)	Returns a String that represents the current Object .

Public Instance Events

 CallbackActivity	This is raised anytime a DDEML callback is executed.

 ConversationActivity	This is raised anytime a conversation is established or terminated.
 ErrorActivity	This is raised anytime there is an error.
 LinkActivity	This is raised anytime an advise loop is established or terminated.
 MessageActivity	This is raised anytime a DDE message is sent or posted.

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Constructor

This initializes a new instance of the `DdeMonitor` class.

```
public DdeMonitor();
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Properties

The properties of the **DdeMonitor** class are listed below. For a complete list of **DdeMonitor** class members, see the [DdeMonitor Members](#) topic.

Public Instance Properties

 Context	This gets the context associated with this instance.
---	--

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.Context Property

This gets the context associated with this instance.

```
public DdeContext Context {get; set;}
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Methods

The methods of the **DdeMonitor** class are listed below. For a complete list of **DdeMonitor** class members, see the [DdeMonitor Members](#) topic.

Public Instance Methods

Dispose	This releases all resources held by this instance.
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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
Start	This starts monitoring the system for DDE activity.
ToString (inherited from Object)	Returns a String that represents the current Object .

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.Dispose Method

This releases all resources held by this instance.

```
public void Dispose();
```

Implements

[IDisposable.](#)

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.Start Method

This starts monitoring the system for DDE activity.

```
public void Start(  
    DdeMonitorFlags flags  
);
```

Parameters

flags

A bitwise combination of `DdeMonitorFlags` that indicate what DDE activity will be monitored.

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor Events

The events of the **DdeMonitor** class are listed below. For a complete list of **DdeMonitor** class members, see the [DdeMonitor Members](#) topic.

Public Instance Events

 CallbackActivity	This is raised anytime a DDEML callback is executed.
 ConversationActivity	This is raised anytime a conversation is established or terminated.
 ErrorActivity	This is raised anytime there is an error.
 LinkActivity	This is raised anytime an advise loop is established or terminated.
 MessageActivity	This is raised anytime a DDE message is sent or posted.

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.EventHandler<TEventArgs> Event

This is raised anytime a DDEML callback is executed.

```
public event EventHandler<TEventArgs> Callback
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.EventHandler<TEventArgs> Event

This is raised anytime a conversation is established or terminated.

```
public event EventHandler<TEventArgs> Conversation
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.EventHandler<TEventArgs> Event

This is raised anytime there is an error.

```
public event EventHandler<TEventArgs> ErrorAc
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.EventHandler<TEventArgs> Event

This is raised anytime an advise loop is established or terminated.

```
public event EventHandler<TEventArgs> LinkAct
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitor.EventHandler<TEventArgs> Event

This is raised anytime a DDE message is sent or posted.

```
public event EventHandler<TEventArgs> Message
```

See Also

[DdeMonitor Class](#) | [NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeMonitorFlags Enumeration

This specifies the different kinds of DDE activity that can be monitored.

This enumeration has a [FlagsAttribute](#) attribute that allows a bitwise combination of its member values.

```
public enum DdeMonitorFlags
```

Members

Member Name	Description	Value
Callback	This indicates activity caused by the execution of a DDEML callback.	134217728
Conversation	This indicates activity caused by conversation.	1073741824
Error	This indicates activity caused by an error.	268435456
Link	This indicates activity caused by an advise loop.	536870912
Message	This indicates activity caused by DDE messages.	100663296

Requirements

Namespace: [NDde.Advanced.Monitor](#)

Assembly: NDde (in NDde.dll)

See Also

[NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)
Generated from assembly NDde [2.1.563.0]

NDde.Client Namespace

This namespace contains classes for creating DDE client applications.

[Namespace hierarchy](#)

Classes

Class	Description
DdeAdviseEventArgs	This contains information about the <code>Advise</code> event.
DdeClient	This represents the client side of a DDE conversation.
DdeDisconnectedEventArgs	This contains information about the <code>Disconnected</code> event.

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

DdeAdviseEventArgs Class

This contains information about the `Advise` event.

For a list of all members of this type, see [DdeAdviseEventArgs Members](#).

[System.Object](#) EventArgs
 [DdeEventArgs](#)
 DdeAdviseEventArgs

```
public class DdeAdviseEventArgs :  
    DdeEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Client](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeAdviseEventArgs Members](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs Members

[DdeAdviseEventArgs overview](#)

Public Instance Properties

 Data	This gets the data associated with this notification or null if this is not a hot advise loop.
 Format	This gets the format of the data included in this notification.
 Item	This gets the item name associated with this notification.
 State	This gets an application defined data object associated with this advise loop.
 Text	This gets the text associated with this notification or null if this is not a hot advise loop.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs Properties

The properties of the **DdeAdviseEventArgs** class are listed below. For a complete list of **DdeAdviseEventArgs** class members, see the [DdeAdviseEventArgs Members](#) topic.

Public Instance Properties

 Data	This gets the data associated with this notification or null if this is not a hot advise loop.
 Format	This gets the format of the data included in this notification.
 Item	This gets the item name associated with this notification.
 State	This gets an application defined data object associated with this advise loop.
 Text	This gets the text associated with this notification or null if this is not a hot advise loop.

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs.Data Property

This gets the data associated with this notification or null if this is not a hot advise loop.

```
public Byte\[\] Data {get;}
```

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs.Format Property

This gets the format of the data included in this notification.

```
public Int32 Format {get;}
```

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs.Item Property

This gets the item name associated with this notification.

```
public String Item {get;}
```

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs.State Property

This gets an application defined data object associated with this advise loop.

```
public Object State {get;}
```

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeAdviseEventArgs.Text Property

This gets the text associated with this notification or null if this is not a hot advise loop.

```
public String Text {get;}
```

See Also

[DdeAdviseEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Class

This represents the client side of a DDE conversation.

For a list of all members of this type, see [DdeClient Members](#).

[System.Object](#) **DdeClient**

```
public class DdeClient : IDisposable
```

Thread Safety

This type is safe for multithreaded operations.

Remarks

DDE conversations are established by specifying a service name and topic name pair. The service name is usually the name of the application acting as a DDE server. A DDE server can respond to multiple service names, but most servers usually only respond to one. The topic name is a logical context for data and is defined by the server application. A server can and usually does support many topic names.

After a conversation has been established by calling [Connect](#) an application can read and write data using the [Request](#) and [Poke](#) methods respectively by specifying an item name supported by the active conversation. An item name identifies a unit of data. An application can also be notified of changes by initiating an advise loop on an item name using the [StartAdvise](#) method. Advise loops can either be warm or hot. A hot advise loop returns the data associated with an item name when it changes whereas a warm advise loop only notifies the application without sending any data. Commands can be sent to the server using the [Execute](#) method.

Callbacks and events are invoked on the thread hosting the [DdeContext](#). All operations must be marshaled onto the thread hosting the [DdeContext](#) associated with this object. Method calls will block until that thread becomes available. An exception will be generated if the thread does not become available in a timely manner.

Example

The following example demonstrates how to use a DdeClient.

```
[C#]
using System;
using System.Text;
using NDde.Client;

public sealed class Client
{
    public static void Main(string[] arg
    {
        // Wait for the user to press EN
        Console.WriteLine("The Server sa
        Console.WriteLine("Press ENTER t
        Console.ReadLine();
        try
        {
            // Create a client that conn
            using (DdeClient client = ne
            {
                // Subscribe to the Disc
                client.Disconnected += 0

                // Connect to the server
                client.Connect();

                // Synchronous Execute 0
                client.Execute("mycomman

                // Synchronous Poke Oper
                client.Poke("myitem", Da

                // Synchronous Request Op
                Console.WriteLine("Reque

                // Asynchronous Execute
```

```

        client.BeginExecute("myc

        // Asynchronous Poke Ope
        client.BeginPoke("myitem

        // Asynchronous Request
        client.BeginRequest("myi

        // Advise Loop
        client.StartAdvise("myit
        client.Advise += OnAdvis

        // Wait for the user to
        Console.WriteLine("Press
        Console.ReadLine());
    }
}
catch (Exception e)
{
    Console.WriteLine(e.ToString
    Console.WriteLine("Press ENT
    Console.ReadLine());
}
}

private static void OnExecuteComple
{
    try
    {
        DdeClient client = (DdeClie
        client.EndExecute(ar);
        Console.WriteLine("OnExecute
    }
    catch (Exception e)
    {
        Console.WriteLine("OnExecute
    }
}

```

```
}

private static void OnPokeComplete(I
{
    try
    {
        DdeClient client = (DdeClien
client.EndPoke(ar);
        Console.WriteLine("OnPokeCom
    }
    catch (Exception e)
    {
        Console.WriteLine("OnPokeCom
    }
}

private static void OnRequestComple
{
    try
    {
        DdeClient client = (DdeClien
byte[] data = client.EndRequ
        Console.WriteLine("OnRequest
    }
    catch (Exception e)
    {
        Console.WriteLine("OnRequest
    }
}

private static void OnStartAdviseCom
{
    try
    {
        DdeClient client = (DdeClien
client.EndStartAdvise(ar);
        Console.WriteLine("OnStartAd
```

```

    }
    catch (Exception e)
    {
        Console.WriteLine("OnStartAd
    }
}

private static void OnStopAdviseComp
{
    try
    {
        DdeClient client = (DdeClie
        client.EndStopAdvise(ar);
        Console.WriteLine("OnStopAdv
    }
    catch (Exception e)
    {
        Console.WriteLine("OnStopAdv
    }
}

private static void OnAdvise(object
{
    Console.WriteLine("OnAdvise: " +
}

private static void OnDisconnected(o
{
    Console.WriteLine(
        "OnDisconnected: " +
        "IsServerInitiated=" + args.
        "IsDisposed=" + args.IsDispo
    }
}

} // class

```

```
[Visual Basic]
```

```
Imports System.Text  
Imports NDde.Client
```

```
Module Program
```

```
Sub Main()
```

```
    ' Wait for the user to press ENT  
    Console.WriteLine("The Server sa  
    Console.WriteLine("Press ENTER t  
    Console.ReadLine()
```

```
Try
```

```
    ' Create a client that conne  
    Using client As DdeClient =
```

```
        ' Subscribe to the Disco  
        AddHandler client.Discon
```

```
        ' Connect to the server.  
        client.Connect()
```

```
        ' Synchronous Execute Op  
        client.Execute("mycomman
```

```
        ' Synchronous Poke Opera  
        client.Poke("myitem", Da
```

```
        ' Synchronous Request Ope  
        Console.WriteLine("Reque
```

```
        ' Asynchronous Execute O  
        client.BeginExecute("myc
```

```
        ' Asynchronous Poke Oper  
        client.BeginPoke("myitem
```

```

        ' Asynchronous Request 0
        client.BeginRequest("myi

        ' Advise Loop
        client.StartAdvise("myit
        AddHandler client.Advise

        ' Wait for the user to p
        Console.WriteLine("Press
        Console.ReadLine()

    End Using

    Catch e As Exception

        Console.WriteLine(e.ToString
        Console.WriteLine("Press ENT
        Console.ReadLine()

    End Try

End Sub

Private Sub OnExecuteComplete(ByVal
    Try
        Dim client As DdeClient = Di
        client.EndExecute(ar)
        Console.WriteLine("OnExecute
    Catch e As Exception
        Console.WriteLine("OnExecute
    End Try
End Sub

Private Sub OnPokeComplete(ByVal ar
    Try
        Dim client As DdeClient = Di

```

```
        client.EndPoke(ar)
        Console.WriteLine("OnPokeCom
Catch e As Exception
        Console.WriteLine("OnPokeCom
    End Try
End Sub

Private Sub OnRequestComplete(ByVal
    Try
        Dim client As DdeClient = Di
        client.EndRequest(ar)
        Console.WriteLine("OnRequest
Catch e As Exception
        Console.WriteLine("OnRequest
    End Try
End Sub

Private Sub OnStartAdviseComplete(By
    Try
        Dim client As DdeClient = Di
        client.EndStartAdvise(ar)
        Console.WriteLine("OnStartAd
Catch e As Exception
        Console.WriteLine("OnStartAd
    End Try
End Sub

Private Sub OnStopAdviseComplete(ByV
    Try
        Dim client As DdeClient = Di
        client.EndStopAdvise(ar)
        Console.WriteLine("OnStopAdv
Catch e As Exception
        Console.WriteLine("OnStopAdv
    End Try
End Sub
```

```
Private Sub OnAdvise(ByVal sender As  
    Console.WriteLine("OnAdvise: " +  
End Sub  
  
Private Sub OnDisconnected(ByVal sen  
    Console.WriteLine( _  
        "OnDisconnected: " + _  
        "IsServerInitiated=" + args.IsS  
        "IsDisposed=" + args.IsDisposed  
End Sub  
  
End Module
```

Requirements

Namespace: [NDde.Client](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeClient Members](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Members

[DdeClient overview](#)

Public Instance Constructors

 DdeClient	Overloaded.
---	-------------

Public Instance Properties

 Context	This gets the context associated with this instance.
 Handle	This gets the DDEML handle associated with this conversation.
 IsConnected	This gets a bool indicating whether the conversation is established.
 IsPaused	This gets a bool indicating whether this conversation is paused.
 Service	This gets the service name associated with this conversation.
 Topic	This gets the topic name associated with this conversation.

Public Instance Methods

 Abandon	This terminates an asynchronous operation.
 BeginExecute	This begins an asynchronous operation to send a command to the server application.
 BeginPoke	This begins an asynchronous operation to send data to the

	server application.
BeginRequest	This begins an asynchronous operation to request data using the specified item name.
BeginStartAdvise	Overloaded.
BeginStopAdvise	This begins an asynchronous operation to terminate the advise loop for the specified item name.
Connect	This establishes a conversation with a server that supports the specified service name and topic name pair.
Disconnect	This terminates the current conversation.
Dispose	Overloaded. This terminates the current conversation and releases all resources held by this instance.
EndExecute	This throws any exception that occurred during the asynchronous operation.
EndPoke	This throws any exception that occurred during the asynchronous operation.
EndRequest	This gets the data returned by the server application for the operation.
EndStartAdvise	This throws any exception that occurred during the operation.
EndStopAdvise	This throws any exception that occurred during the operation.
Equals (inherited from Object)	Determines whether the

	specified Object is equal to the current Object .
◆ Execute	This sends a command to the server application.
◆ GetHashCode (inherited from Object)	Serves as a hash function for a particular type. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
◆ GetType (inherited from Object)	Gets the Type of the current instance.
◆ Pause	This pauses the current conversation.
◆ Poke	Overloaded.
◆ Request	Overloaded.
◆ Resume	This resumes the current conversation.
◆ StartAdvise	Overloaded.
◆ StopAdvise	This terminates the advise loop for the specified item name.
◆ ToString (inherited from Object)	Returns a String that represents the current Object .
◆ TryConnect	This establishes a conversation with a server that supports the specified service name and topic name pair.
◆ TryExecute	This sends a command to the server application.
◆ TryPoke	This sends data to the server application.
◆ TryRequest	This requests data using the specified item name.

Public Instance Events

 Advise	This is raised when the data has changed for an item name that has an advise loop.
 Disconnected	This is raised when the client has been disconnected.

Protected Instance Methods

 Dispose	Overloaded. This contains the implementation to release all resources held by this instance.
 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

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Constructor

Overload List

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair.

```
public DdeClient\(String,String\);
```

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair and uses the specified context.

```
public DdeClient\(String,String,DdeContext\);
```

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair using the specified synchronizing object.

```
public DdeClient\(String,String,ISynchronizeInvoke\);
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Constructor (String, String)

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair.

```
public DdeClient(  
    String service,  
    String topic  
);
```

Parameters

service

A service name supported by a server application.

topic

A topic name support by a server application.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when servic or topic exceeds 255 characters.
ArgumentNullException	This is thrown when service or topic is a null reference.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Constructor (String, String, ISynchronizeInvoke)

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair using the specified synchronizing object.

```
public DdeClient(  
    String service,  
    String topic,  
    ISynchronizeInvoke synchronizingObject  
);
```

Parameters

service

A service name supported by a server application.

topic

A topic name support by a server application.

synchronizingObject

The synchronizing object to use for this instance.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when service or topic exceeds 255 characters.
ArgumentNullException	This is thrown when service or topic is a null reference.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Constructor (String, String, DdeContext)

This initializes a new instance of the `DdeClient` class that can connect to a server that supports the specified service name and topic name pair and uses the specified context.

```
public DdeClient(  
    String service,  
    String topic,  
    DdeContext context  
);
```

Parameters

service

A service name supported by a server application.

topic

A topic name support by a server application.

context

The context to use for execution.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when servic or topic exceeds 255 characters.
ArgumentNullException	This is thrown when service or topic is a null reference.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Properties

The properties of the **DdeClient** class are listed below. For a complete list of **DdeClient** class members, see the [DdeClient Members](#) topic.

Public Instance Properties

 Context	This gets the context associated with this instance.
 Handle	This gets the DDEML handle associated with this conversation.
 IsConnected	This gets a bool indicating whether the conversation is established.
 IsPaused	This gets a bool indicating whether this conversation is paused.
 Service	This gets the service name associated with this conversation.
 Topic	This gets the topic name associated with this conversation.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Context Property

This gets the context associated with this instance.

```
public virtual DdeContext Context {get; set;}
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Handle Property

This gets the DDEML handle associated with this conversation.

```
public virtual IntPtr Handle {get;}
```

Remarks

This can be used in any DDEML function requiring a conversation handle.

CAUTION Incorrect usage of the DDEML can cause this object to function incorrectly and can lead to resource leaks.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.IsConnected Property

This gets a bool indicating whether the conversation is established.

```
public virtual Boolean IsConnected {get;}
```

Remarks

CAUTION Do not assume that the conversation is still established after checking this property. The conversation can terminate at any time.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.IsPaused Property

This gets a bool indicating whether this conversation is paused.

```
public virtual Boolean IsPaused {get;}
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Service Property

This gets the service name associated with this conversation.

```
public virtual String Service {get; set;}
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Topic Property

This gets the topic name associated with this conversation.

```
public virtual String Topic {get; set;}
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Methods

The methods of the **DdeClient** class are listed below. For a complete list of **DdeClient** class members, see the [DdeClient Members](#) topic.

Public Instance Methods

 Abandon	This terminates an asynchronous operation.
 BeginExecute	This begins an asynchronous operation to send a command to the server application.
 BeginPoke	This begins an asynchronous operation to send data to the server application.
 BeginRequest	This begins an asynchronous operation to request data using the specified item name.
 BeginStartAdvise	Overloaded.
 BeginStopAdvise	This begins an asynchronous operation to terminate the advise loop for the specified item name.
 Connect	This establishes a conversation with a server that supports the specified service name and topic name pair.
 Disconnect	This terminates the current conversation.
 Dispose	Overloaded. This terminates the current conversation and releases all resources held by this instance.
 EndExecute	This throws any exception that

	occurred during the asynchronous operation.
◆ EndPoke	This throws any exception that occurred during the asynchronous operation.
◆ EndRequest	This gets the data returned by the server application for the operation.
◆ EndStartAdvise	This throws any exception that occurred during the operation.
◆ EndStopAdvise	This throws any exception that occurred during the operation.
◆ Equals (inherited from Object)	Determines whether the specified Object is equal to the current Object .
◆ Execute	This sends a command to the server application.
◆ GetHashCode (inherited from Object)	Serves as a hash function for a particular type. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
◆ GetType (inherited from Object)	Gets the Type of the current instance.
◆ Pause	This pauses the current conversation.
◆ Poke	Overloaded.
◆ Request	Overloaded.
◆ Resume	This resumes the current conversation.
◆ StartAdvise	Overloaded.
◆ StopAdvise	This terminates the advise loop

	for the specified item name.
 ToString (inherited from Object)	Returns a String that represents the current Object .
 TryConnect	This establishes a conversation with a server that supports the specified service name and topic name pair.
 TryExecute	This sends a command to the server application.
 TryPoke	This sends data to the server application.
 TryRequest	This requests data using the specified item name.

Protected Instance Methods

 Dispose	Overloaded. This contains the implementation to release all resources held by this instance.
 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

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Abandon Method

This terminates an asynchronous operation.

```
public virtual void Abandon(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The *IAsyncResult* object returned by a call that begins an asynchronous operation.

Remarks

This method does nothing if the asynchronous operation has already completed.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <i>asyncResult</i> is an invalid <i>IAsyncResult</i> .
ArgumentNullException	This is thrown when <i>asyncResult</i> is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the asynchronous operation could not be abandoned.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginExecute Method

This begins an asynchronous operation to send a command to the server application.

```
public virtual IAsyncResult BeginExecute(  
    String command,  
    AsyncCallback callback,  
    Object state  
);
```

Parameters

command

The command to be sent to the server application.

callback

The delegate to invoke when this operation completes.

state

An application defined data object to associate with this operation.

Return Value

An [IAsyncResult](#) object for this operation.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when command exceeds 255 characters.
ArgumentNullException	This is thrown when command is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the asynchronous operation could not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginPoke Method

This begins an asynchronous operation to send data to the server application.

```
public virtual IAAsyncResult BeginPoke(  
    String item,  
    Byte\[\] data,  
    Int32 format,  
    AsyncCallback callback,  
    Object state  
);
```

Parameters

item

An item name supported by the current conversation.

data

The data to send.

format

The format of the data.

callback

The delegate to invoke when this operation completes.

state

An application defined data object to associate with this operation.

Return Value

An [IAAsyncResult](#) object for this operation.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters.
ArgumentNullException	This is thrown when item or

	data is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the asynchronous operation could not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginRequest Method

This begins an asynchronous operation to request data using the specified item name.

```
public virtual IAsyncResult BeginRequest(  
    String item,  
    Int32 format,  
    AsyncCallback callback,  
    Object state  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to return.

callback

The delegate to invoke when this operation completes.

state

An application defined data object to associate with this operation.

Return Value

An [IAsyncResult](#) object for this operation.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the client is not connected.

[DdeException](#)

This is thrown when the asynchronous operation could not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginStartAdvise Method

Overload List

This begins an asynchronous operation to initiate an advise loop on the specified item name.

[public virtual IAsyncResult BeginStartAdvise\(String,Int32,Boolean,A](#)

This begins an asynchronous operation to initiate an advise loop on the specified item name.

[public virtual IAsyncResult BeginStartAdvise\(String,Int32,Boolean,B](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginStartAdvise Method (String, Int32, Boolean, AsyncCallback, Object)

This begins an asynchronous operation to initiate an advise loop on the specified item name.

```
public virtual IAsyncResult BeginStartAdvise(  
    String item,  
    Int32 format,  
    Boolean hot,  
    AsyncCallback callback,  
    Object asyncState  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to be returned.

hot

A bool indicating whether data should be included with the notification.

callback

The delegate to invoke when this operation completes.

asyncState

An application defined data object to associate with this operation.

Return Value

An [IAsyncResult](#) object for this operation.

Events

Event Type	Reason
Advise	

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is already being advised or when the client is not connected.
DdeException	This is thrown when the asynchronous operation could not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.BeginStartAdvise Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginStartAdvise Method (String, Int32, Boolean, Boolean, AsyncCallback, Object, Object)

This begins an asynchronous operation to initiate an advise loop on the specified item name.

```
public virtual IAsyncResult BeginStartAdvise(  
    String item,  
    Int32 format,  
    Boolean hot,  
    Boolean acknowledge,  
    AsyncCallback callback,  
    Object asyncState,  
    Object adviseState  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to be returned.

hot

A bool indicating whether data should be included with the notification.

acknowledge

A bool indicating whether the client should acknowledge each advisory before the server will send another.

callback

The delegate to invoke when this operation completes.

asyncState

An application defined data object to associate with this operation.

adviseState

An application defined data object to associate with this advise loop.

Return Value

An `IAsyncResult` object for this operation.

Events

Event Type	Reason
Advise	

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is already being advised or when the client is not connected.
DdeException	This is thrown when the asynchronous operation could not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.BeginStartAdvise Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.BeginStopAdvise Method

This begins an asynchronous operation to terminate the advise loop for the specified item name.

```
public virtual IAsyncResult BeginStopAdvise(  
    String item,  
    AsyncCallback callback,  
    Object state  
);
```

Parameters

item

An item name that has an active advise loop.

callback

The delegate to invoke when this operation completes.

state

An application defined data object to associate with this operation.

Return Value

An [IAsyncResult](#) object for this operation.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is not being advised or when the client is not connected.
DdeException	This is thrown when the asynchronous operation could

not begin.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Connect Method

This establishes a conversation with a server that supports the specified service name and topic name pair.

```
public virtual void Connect();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the client is already connected.
DdeException	This is thrown when the client could not connect to the server.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Disconnect Method

This terminates the current conversation.

```
public virtual void Disconnect();
```

Events

Event Type	Reason
Disconnected	

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the client was not previously connected.
DdeException	This is thrown when the client could not disconnect from the server.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Dispose Method

This terminates the current conversation and releases all resources held by this instance.

Overload List

This terminates the current conversation and releases all resources held by this instance.

[public void Dispose\(\);](#)

This contains the implementation to release all resources held by this instance.

[protected virtual void Dispose\(Boolean\);](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Dispose Method ()

This terminates the current conversation and releases all resources held by this instance.

```
public void Dispose();
```

Implements

[IDisposable.](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Dispose Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Dispose Method (Boolean)

This contains the implementation to release all resources held by this instance.

```
protected virtual void Dispose(  
    Boolean disposing  
);
```

Parameters

disposing

True if called by Dispose, false otherwise.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Dispose Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EndExecute Method

This throws any exception that occurred during the asynchronous operation.

```
public virtual void EndExecute(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The *IAsyncResult* object returned by a call to *BeginExecute*.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <i>asyncResult</i> is an invalid <i>IAsyncResult</i> .
ArgumentNullException	This is thrown when <i>asyncResult</i> is a null reference.
DdeException	This is thrown when the server does not process the command.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EndPoke Method

This throws any exception that occurred during the asynchronous operation.

```
public virtual void EndPoke(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The `IAsyncResult` object returned by a call to `BeginPoke`.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <code>asyncResult</code> is an invalid <code>IAsyncResult</code> .
ArgumentNullException	This is thrown when <code>asyncResult</code> is a null reference.
DdeException	This is thrown when the server does not process the data.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EndRequest Method

This gets the data returned by the server application for the operation.

```
public virtual byte[] EndRequest(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The *IAsyncResult* object returned by a call to *BeginRequest*.

Return Value

The data returned by the server application.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <i>asyncResult</i> is an invalid <i>IAsyncResult</i> .
ArgumentNullException	This is thrown when <i>asyncResult</i> is a null reference.
DdeException	This is thrown when the server does not process the request.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EndStartAdvise Method

This throws any exception that occurred during the operation.

```
public virtual void EndStartAdvise(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The *IAsyncResult* object returned by a call to *BeginPoke*.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <i>asyncResult</i> is an invalid <i>IAsyncResult</i> .
ArgumentNullException	This is thrown when <i>asyncResult</i> is a null reference.
DdeException	This is thrown when the server does not initiate the advise loop.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EndStopAdvise Method

This throws any exception that occurred during the operation.

```
public virtual void EndStopAdvise(  
    IAsyncResult asyncResult  
);
```

Parameters

asyncResult

The *IAsyncResult* object returned by a call to *BeginPoke*.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when <i>asyncResult</i> is an invalid <i>IAsyncResult</i> .
ArgumentNullException	This is thrown when <i>asyncResult</i> is a null reference.
DdeException	This is thrown when the server does not terminate the advise loop.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Execute Method

This sends a command to the server application.

```
public virtual void Execute(  
    String command,  
    Int32 timeout  
);
```

Parameters

command

The command to be sent to the server application.

timeout

The amount of time in milliseconds to wait for a response.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when command exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when command is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the server does not process the command.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)
Generated from assembly NDde [2.1.563.0]

DdeClient.Pause Method

This pauses the current conversation.

```
public virtual void Pause();
```

Remarks

Synchronous operations will timeout if the conversation is paused. Asynchronous operations can begin, but will not complete until the conversation has resumed.

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the conversation is already paused.
DdeException	This is thrown when the conversation could not be paused or when the client is not connected.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Poke Method

Overload List

This sends data to the server application.

[public virtual void Poke\(String,Byte\[\],Int32,Int32\);](#)

This sends data to the server application.

[public virtual void Poke\(String,String,Int32\);](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Poke Method (String, Byte[], Int32, Int32)

This sends data to the server application.

```
public virtual void Poke(  
    String item,  
    Byte[] data,  
    Int32 format,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

data

The data to send.

format

The format of the data.

timeout

The amount of time in milliseconds to wait for a response.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item or data is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the server

does not process the data.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Poke Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Poke Method (String, String, Int32)

This sends data to the server application.

```
public virtual void Poke(  
    String item,  
    String data,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

data

The data to send.

timeout

The amount of time in milliseconds to wait for a response.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item or data is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the server does not process the data.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Poke Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Request Method

Overload List

This requests data using the specified item name.

[public virtual string Request\(String,Int32\);](#)

This requests data using the specified item name.

[public virtual byte\[\] Request\(String,Int32,Int32\);](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Request Method (String, Int32)

This requests data using the specified item name.

```
public virtual string Request(  
    String item,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

timeout

The amount of time in milliseconds to wait for a response.

Return Value

The data returned by the server application in CF_TEXT format.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the server does not process the request.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Request Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Request Method (String, Int32, Int32)

This requests data using the specified item name.

```
public virtual byte\[\] Request(  
    String item,  
    Int32 format,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to return.

timeout

The amount of time in milliseconds to wait for a response.

Return Value

The data returned by the server application.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the client is not connected.
DdeException	This is thrown when the server

does not process the request.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.Request Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.Resume Method

This resumes the current conversation.

```
public virtual void Resume();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the conversation was not previously paused or when the client is not connected.
DdeException	This is thrown when the conversation could not be resumed.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.StartAdvise Method

Overload List

This initiates an advise loop on the specified item name.

[public virtual void StartAdvise\(String,Int32,Boolean,Boolean,Int32,O](#)

This initiates an advise loop on the specified item name.

[public virtual void StartAdvise\(String,Int32,Boolean,Int32\):](#)

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.StartAdvise Method (String, Int32, Boolean, Boolean, Int32, Object)

This initiates an advise loop on the specified item name.

```
public virtual void StartAdvise(  
    String item,  
    Int32 format,  
    Boolean hot,  
    Boolean acknowledge,  
    Int32 timeout,  
    Object adviseState  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to return.

hot

A bool indicating whether data should be included with the notification.

acknowledge

A bool indicating whether the client should acknowledge each advisory before the server will send another.

timeout

The amount of time in milliseconds to wait for a response.

adviseState

An application defined data object to associate with this advise loop.

Remarks

This operation will timeout if the conversation is paused.

Events

Event Type	Reason
Advise	

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is already being advised or when the client is not connected.
DdeException	This is thrown when the server does not initiate the advise loop.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.StartAdvise Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.StartAdvise Method (String, Int32, Boolean, Int32)

This initiates an advise loop on the specified item name.

```
public virtual void StartAdvise(  
    String item,  
    Int32 format,  
    Boolean hot,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to return.

hot

A bool indicating whether data should be included with the notification.

timeout

The amount of time in milliseconds to wait for a response.

Remarks

This operation will timeout if the conversation is paused.

Events

Event Type	Reason
Advise	

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or

	timeout is negative.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is already being advised or when the client is not connected.
DdeException	This is thrown when the server does not initiate the advise loop.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#) | [DdeClient.StartAdvise Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.StopAdvise Method

This terminates the advise loop for the specified item name.

```
public virtual void StopAdvise(  
    String item,  
    Int32 timeout  
);
```

Parameters

item

An item name that has an active advise loop.

timeout

The amount of time in milliseconds to wait for a response.

Remarks

This operation will timeout if the conversation is paused.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when item exceeds 255 characters or timeout is negative.
ArgumentNullException	This is thrown when item is a null reference.
InvalidOperationException	This is thrown when the item is not being advised or when the client is not connected.
DdeException	This is thrown when the server does not terminate the advise loop.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.TryConnect Method

This establishes a conversation with a server that supports the specified service name and topic name pair.

```
public virtual int TryConnect();
```

Return Value

Zero if the operation succeed or non-zero if the operation failed.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.TryExecute Method

This sends a command to the server application.

```
public virtual int TryExecute(  
    String command,  
    Int32 timeout  
);
```

Parameters

command

The command to be sent to the server application.

timeout

The amount of time in milliseconds to wait for a response.

Return Value

Zero if the operation succeed or non-zero if the operation failed.

Remarks

This operation will timeout if the conversation is paused.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.TryPoke Method

This sends data to the server application.

```
public virtual int TryPoke(  
    String item,  
    Byte\[\] data,  
    Int32 format,  
    Int32 timeout  
);
```

Parameters

item

An item name supported by the current conversation.

data

The data to send.

format

The format of the data.

timeout

The amount of time in milliseconds to wait for a response.

Return Value

Zero if the operation succeed or non-zero if the operation failed.

Remarks

This operation will timeout if the conversation is paused.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.TryRequest Method

This requests data using the specified item name.

```
public virtual int TryRequest(  
    String item,  
    Int32 format,  
    Int32 timeout,  
    out Byte\[\]& data  
);
```

Parameters

item

An item name supported by the current conversation.

format

The format of the data to return.

timeout

The amount of time in milliseconds to wait for a response.

data

The data returned by the server application.

Return Value

Zero if the operation succeeded or non-zero if the operation failed.

Remarks

This operation will timeout if the conversation is paused.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient Events

The events of the **DdeClient** class are listed below. For a complete list of **DdeClient** class members, see the [DdeClient Members](#) topic.

Public Instance Events

 Advise	This is raised when the data has changed for an item name that has an advise loop.
 Disconnected	This is raised when the client has been disconnected.

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EventHandler<TEventArgs> Event

This is raised when the data has changed for an item name that has an advise loop.

```
public event EventHandler<TEventArgs> Advise;
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeClient.EventHandler<TEventArgs> Event

This is raised when the client has been disconnected.

```
public event EventHandler<TEventArgs> Disconn
```

See Also

[DdeClient Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeDisconnectedEventArgs Class

This contains information about the `Disconnected` event.

For a list of all members of this type, see [DdeDisconnectedEventArgs Members](#).

[System.Object](#) EventArgs

[DdeEventArgs](#)

DdeDisconnectedEventArgs

```
public class DdeDisconnectedEventArgs :  
    DdeEventArgs
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Client](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeDisconnectedEventArgs Members](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeDisconnectedEventArgs Members

[DdeDisconnectedEventArgs overview](#)

Public Instance Properties

 IsDisposed	This gets a bool indicating whether the client disconnected because <code>Dispose</code> was explicitly called.
 IsServerInitiated	This gets a bool indicating whether the client disconnected because of the server.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from DdeEventArgs)	This returns a string containing the current values of all properties.

See Also

[DdeDisconnectedEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeDisconnectedEventArgs Properties

The properties of the **DdeDisconnectedEventArgs** class are listed below. For a complete list of **DdeDisconnectedEventArgs** class members, see the [DdeDisconnectedEventArgs Members](#) topic.

Public Instance Properties

 IsDisposed	This gets a bool indicating whether the client disconnected because Dispose was explicitly called.
 IsServerInitiated	This gets a bool indicating whether the client disconnected because of the server.

See Also

[DdeDisconnectedEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeDisconnectedEventArgs.IsDisposed Property

This gets a bool indicating whether the client disconnected because `Dispose` was explicitly called.

```
public Boolean IsDisposed {get;}
```

Remarks

The value will be true if `Dispose` was explicitly called on `DdeClient`. The `DdeClient` sending this event has been disposed and can no longer be accessed. Any exception thrown in the currently executing method will be ignored.

See Also

[DdeDisconnectedEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeDisconnectedEventArgs.IsServerInitiated Property

This gets a bool indicating whether the client disconnected because of the server.

```
public Boolean IsServerInitiated {get;}
```

See Also

[DdeDisconnectedEventArgs Class](#) | [NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

NDde.Server Namespace

This namespace contains classes for creating DDE server applications.

[Namespace hierarchy](#)

Classes

Class	Description
DdeConversation	This represents a DDE conversation established on a DdeServer .
DdeServer	This represents the server side of DDE conversations.

Structures

Structure	Description
DdeServer.ExecuteResult	This is the return value of the <code>OnExecute</code> method.
DdeServer.PokeResult	This is the return value of the <code>OnPoke</code> method.
DdeServer.RequestResult	This is the return value of the <code>OnRequest</code> method.

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

DdeConversation Class

This represents a DDE conversation established on a [DdeServer](#).

For a list of all members of this type, see [DdeConversation Members](#).

[System.Object](#) **DdeConversation**

```
public class DdeConversation
```

Thread Safety

This type is safe for multithreaded operations.

Requirements

Namespace: [NDde.Server](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeConversation Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation Members

[DdeConversation overview](#)

Public Instance Properties

 Handle	This gets the DDEML handle associated with this conversation.
 IsPaused	This gets a bool indicating whether this conversation is paused.
 Service	This gets the service name associated with this conversation.
 Tag	This gets an application defined data object associated with this conversation.
 Topic	This gets the topic name associated with this conversation.

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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString	This returns a string containing the current values of all

properties.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation Properties

The properties of the **DdeConversation** class are listed below. For a complete list of **DdeConversation** class members, see the [DdeConversation Members](#) topic.

Public Instance Properties

 Handle	This gets the DDEML handle associated with this conversation.
 IsPaused	This gets a bool indicating whether this conversation is paused.
 Service	This gets the service name associated with this conversation.
 Tag	This gets an application defined data object associated with this conversation.
 Topic	This gets the topic name associated with this conversation.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.Handle Property

This gets the DDEML handle associated with this conversation.

```
public IntPtr Handle {get;}
```

Remarks

This can be used in any DDEML function requiring a conversation handle.

CAUTION Incorrect usage of the DDEML can cause this object to function incorrectly and can lead to resource leaks.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.IsPaused Property

This gets a bool indicating whether this conversation is paused.

```
public Boolean IsPaused {get;}
```

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.Service Property

This gets the service name associated with this conversation.

```
public String Service {get;}
```

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.Tag Property

This gets an application defined data object associated with this conversation.

```
public Object Tag {get; set;}
```

Remarks

Use this property to carry state information with the conversation.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.Topic Property

This gets the topic name associated with this conversation.

```
public String Topic {get;}
```

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation Methods

The methods of the **DdeConversation** class are listed below. For a complete list of **DdeConversation** class members, see the [DdeConversation Members](#) 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString	This returns a string containing the current values of all properties.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeConversation.ToString Method

This returns a string containing the current values of all properties.

```
public override string ToString();
```

Return Value

A string containing the current values of all properties.

See Also

[DdeConversation Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Class

This represents the server side of DDE conversations.

For a list of all members of this type, see [DdeServer Members](#).

[System.Object](#) **DdeServer**

```
public class DdeServer : IDisposable
```

Thread Safety

This type is safe for multithreaded operations.

Remarks

DDE conversations are established by specifying a service name and topic name pair. The service name is usually the name of the application acting as a DDE server. A DDE server can respond to multiple service names, but most servers usually only respond to one. The topic name is a logical context for data and is defined by the server application. A server can and usually does support many topic names.

After this object has registered its service name by calling the [Register](#) method clients can connect to it by specifying the service name the server registered and a topic name that it supports.

Event methods are invoked on the thread hosting the [DdeContext](#). All operations must be marshaled onto the thread hosting the [DdeContext](#) associated with this object. Method calls will block until that thread becomes available. An exception will be generated if the thread does not become available in a timely manner.

Notes to Inheritors: The event methods must be overridden in a subclass as needed.

Example

The following example demonstrates how to use a DdeServer.

```
[C#]  
using System;
```

```

using System.Collections;
using System.Timers;
using NDde.Server;

public class Server
{
    public static void Main()
    {
        try
        {
            // Create a server that will
            using (DdeServer server = ne
            {
                // Register the service
                server.Register();

                // Wait for the user to
                Console.WriteLine("Press
                Console.ReadLine());
            }
        }
        catch (Exception e)
        {
            Console.WriteLine(e);
            Console.WriteLine("Press ENT
            Console.ReadLine());
        }
    }

    private sealed class MyServer : DdeS
    {
        private System.Timers.Timer _Tim

        public MyServer(string service)
        {
            // Create a timer that will
            _Timer.Elapsed += this.OnTim

```

```

        _Timer.Interval = 1000;
        _Timer.SynchronizingObject =
        _Timer.Start();
    }

    private void OnTimerElapsed(object
    {
        // Advise all topic name and
        Advise("*", "*");
    }

    protected override bool OnBefore
    {
        Console.WriteLine("OnBeforeC
            + " Service='" + base.Se
            + " Topic='" + topic + "

        return true;
    }

    protected override void OnAfterC
    {
        Console.WriteLine("OnAfterCo
            + " Service='" + convers
            + " Topic='" + conversat
            + " Handle=" + conversat
    }

    protected override void OnDiscon
    {
        Console.WriteLine("OnDisconn
            + " Service='" + convers
            + " Topic='" + conversat
            + " Handle=" + conversat
    }

    protected override bool OnStartA

```

```

    {
        Console.WriteLine("OnStartAd
            + " Service='" + convers
            + " Topic='" + conversat
            + " Handle=" + conversat
            + " Item='" + item + "'"
            + " Format=" + format.To

        // Initiate the advisory loo
        return format == 1;
    }

protected override void OnStopAd
{
    Console.WriteLine("OnStopAdv
        + " Service='" + convers
        + " Topic='" + conversat
        + " Handle=" + conversat
        + " Item='" + item + "'"
    }

protected override ExecuteResult
{
    Console.WriteLine("OnExecute
        + " Service='" + convers
        + " Topic='" + conversat
        + " Handle=" + conversat
        + " Command='" + command

    // Tell the client that the
    return ExecuteResult.Process
}

protected override PokeResult On
{
    Console.WriteLine("OnPoke:".
        + " Service='" + convers

```

```

        + " Topic='" + conversat
        + " Handle=" + conversat
        + " Item='" + item + "'"
        + " Data=" + data.Length
        + " Format=" + format.To

        // Tell the client that the
        return PokeResult.Processed;
    }

protected override RequestResult
{
    Console.WriteLine("OnRequest
        + " Service='" + convers
        + " Topic='" + conversat
        + " Handle=" + conversat
        + " Item='" + item + "'"
        + " Format=" + format.To

        // Return data to the client
        if (format == 1)
        {
            return new RequestResult
        }
        return RequestResult.NotProc
    }

protected override byte[] OnAdvi
{
    Console.WriteLine("OnAdvise:
        + " Service='" + this.Se
        + " Topic='" + topic + "
        + " Item='" + item + "'"
        + " Format=" + format.To

        // Send data to the client o
        if (format == 1)

```

```

        {
            return System.Text.Encod
        }
        return null;
    }

} // class

} // class

```

[Visual Basic]

```

Imports NDde.Server

Module Program

    Sub Main()

        Try

            ' Create a server that will
            Using server As DdeServer =

                ' Register the service n
                server.Register()

                ' Wait for the user to p
                Console.WriteLine("Press
                Console.ReadLine()

            End Using

            Catch e As Exception

                Console.WriteLine(e)
                Console.WriteLine("Press ENT
                Console.ReadLine()

```

```

        End Try

    End Sub

    Private Class MyServer
        Inherits DdeServer

        Private WithEvents theTimer As S

        Public Sub New(ByVal service As
            MyBase)
            MyBase.New(service)
            ' Create a timer that will b
            theTimer.Interval = 1000
            theTimer.SynchronizingObject
            theTimer.Start()
        End Sub

        Private Sub theTimer_Elapsed(ByV
            Me)
            Me.Advise("*", "*")
        End Sub

        Protected Overrides Function OnB
            Console.WriteLine("OnBeforeC
            + " Service='" + MyBase.Ser
            + " Topic='" + topic + "'")

            Return True
        End Function

        Protected Overrides Sub OnAfterC
            Console.WriteLine("OnAfterCo
            + " Service='" + conversati
            + " Topic='" + conversation
            + " Handle=" + conversation
        End Sub
    
```

```
Protected Overrides Sub OnDisconn
    Console.WriteLine("OnDisconn
        + " Service='" + conversati
        + " Topic='" + conversation
        + " Handle=" + conversation
    End Sub
```

```
Protected Overrides Function OnS
    Console.WriteLine("OnStartAd
        + " Service='" + conversati
        + " Topic='" + conversation
        + " Handle=" + conversation
        + " Item='" + item + "'" _
        + " Format=" + format.ToStr

        ' Initiate the advisory loop
    Return format = 1
    End Function
```

```
Protected Overrides Sub OnStopAd
    Console.WriteLine("OnStopAdv
        + " Service='" + conversat
        + " Topic='" + conversatio
        + " Handle=" + conversatio
        + " Item='" + item + "'")
    End Sub
```

```
Protected Overrides Function OnE
    Console.WriteLine("OnExecute
        + " Service='" + conversati
        + " Topic='" + conversation
        + " Handle=" + conversation
        + " Command='" + command +

        ' Tell the client that the c
    Return ExecuteResult.Process
    End Function
```

```

Protected Overrides Function OnP
    Console.WriteLine("OnPoke:".
        + " Service='" + conversati
        + " Topic='" + conversation
        + " Handle=" + conversation
        + " Item='" + item + "'" _
        + " Data=" + data.Length.To
        + " Format=" + format.ToStr

        ' Tell the client that the d
    Return PokeResult.Processed
End Function

```

```

Protected Overrides Function OnR
    Console.WriteLine("OnRequest
        + " Service='" + conversati
        + " Topic='" + conversation
        + " Handle=" + conversation
        + " Item='" + item + "'" _
        + " Format=" + format.ToStr

        ' Return data to the client
    If format = 1 Then
        Return New RequestResult
    End If
    Return RequestResult.NotProc
End Function

```

```

Protected Overrides Function OnA
    Console.WriteLine("OnAdvise:
        + " Service='" + Me.Service
        + " Topic='" + topic + "'"
        + " Item='" + item + "'" _
        + " Format=" + format.ToStr

        ' Send data to the client on

```

```
        If format = 1 Then
            Return System.Text.Encod
        End If
        Return Nothing
    End Function

End Class

End Module
```

Requirements

Namespace: [NDde.Server](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeServer Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Members

[DdeServer overview](#)

Public Instance Constructors

 DdeServer	Overloaded.
---	-------------

Public Instance Properties

 Context	This gets the context associated with his instance.
 IsRegistered	This gets a bool indicating whether the service name is registered.
 Service	This gets the service name associated with this server.

Public Instance Methods

 Advise	This notifies all clients that data has changed for the specified topic name and item name pair.
 Disconnect	Overloaded.
 Dispose	Overloaded. This unregisters service name and releases all resources held by this instance.
 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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
 GetType (inherited from Object)	Gets the Type of the current instance.

 Pause	Overloaded.
 Register	This registers the service name.
 Resume	Overloaded.
 ToString (inherited from Object)	Returns a String that represents the current Object .
 Unregister	This unregisters the service name.

Protected Instance Methods

 Dispose	Overloaded. This contains the implementation to release all resources held by this instance.
 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 .
 OnAdvise	This is invoked when the server is performing a hot advise.
 OnAfterConnect	This is invoked when a client has successfully established a conversation.
 OnBeforeConnect	This is invoked when a client attempts to establish a conversation.
 OnDisconnect	This is invoked when a client terminates a conversation.
 OnExecute	This is invoked when a client sends a command.
 OnPoke	This is invoked when a client

	sends data.
 OnRequest	This is invoked when a client attempts to request data.
 OnStartAdvise	This is invoked when a client attempts to initiate an advise loop.
 OnStopAdvise	This is invoked when a client terminates an advise loop.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Constructor

Overload List

This initializes a new instance of the `DdeServer` class that can register the specified service name.

[public DdeServer\(String\);](#)

This initializes a new instance of the `DdeServer` class that can register the specified service name and uses the specified context.

[public DdeServer\(String,DdeContext\);](#)

This initializes a new instance of the `DdeServer` class that can register the specified service name and using the specified synchronizing object.

[public DdeServer\(String,ISynchronizeInvoke\);](#)

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Constructor (String)

This initializes a new instance of the `DdeServer` class that can register the specified service name.

```
public DdeServer(  
    String service  
);
```

Parameters

service

The service name that this instance can register.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when service exceeds 255 characters..
ArgumentNullException	This is thrown when service is a null reference.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Constructor (String, ISynchronizeInvoke)

This initializes a new instance of the `DdeServer` class that can register the specified service name and using the specified synchronizing object.

```
public DdeServer(  
    String service,  
    ISynchronizeInvoke synchronizingObject  
);
```

Parameters

service

The service name that this instance can register.

synchronizingObject

The synchronizing object to use for this instance.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when service exceeds 255 characters..
ArgumentNullException	This is thrown when service is a null reference.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Constructor (String, DdeContext)

This initializes a new instance of the `DdeServer` class that can register the specified service name and uses the specified context.

```
public DdeServer(  
    String service,  
    DdeContext context  
);
```

Parameters

service

The service name that this instance can register.

context

The context to use for execution.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when service exceeds 255 characters..
ArgumentNullException	This is thrown when service is a null reference.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer Constructor Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Properties

The properties of the **DdeServer** class are listed below. For a complete list of **DdeServer** class members, see the [DdeServer Members](#) topic.

Public Instance Properties

 Context	This gets the context associated with his instance.
 IsRegistered	This gets a bool indicating whether the service name is registered.
 Service	This gets the service name associated with this server.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Context Property

This gets the context associated with his instance.

```
public DdeContext Context {get; set;}
```

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.IsRegistered Property

This gets a bool indicating whether the service name is registered.

```
public virtual Boolean IsRegistered {get;}
```

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Service Property

This gets the service name associated with this server.

```
public virtual String Service {get; set;}
```

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer Methods

The methods of the **DdeServer** class are listed below. For a complete list of **DdeServer** class members, see the [DdeServer Members](#) topic.

Public Instance Methods

Advise	This notifies all clients that data has changed for the specified topic name and item name pair.
Disconnect	Overloaded.
Dispose	Overloaded. This unregisters service name and releases all resources held by this instance.
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. GetHashCode is suitable for use in hashing algorithms and data structures like a hash table.
GetType (inherited from Object)	Gets the Type of the current instance.
Pause	Overloaded.
Register	This registers the service name.
Resume	Overloaded.
ToString (inherited from Object)	Returns a String that represents the current Object .
Unregister	This unregisters the service name.

Protected Instance Methods

--	--

 Dispose	Overloaded. This contains the implementation to release all resources held by this instance.
 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 .
 OnAdvise	This is invoked when the server is performing a hot advise.
 OnAfterConnect	This is invoked when a client has successfully established a conversation.
 OnBeforeConnect	This is invoked when a client attempts to establish a conversation.
 OnDisconnect	This is invoked when a client terminates a conversation.
 OnExecute	This is invoked when a client sends a command.
 OnPoke	This is invoked when a client sends data.
 OnRequest	This is invoked when a client attempts to request data.
 OnStartAdvise	This is invoked when a client attempts to initiate an advise loop.
 OnStopAdvise	This is invoked when a client terminates an advise loop.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Advise Method

This notifies all clients that data has changed for the specified topic name and item name pair.

```
public virtual void Advise(  
    String topic,  
    String item  
);
```

Parameters

topic

A topic name supported by this server.

item

An item name supported by this server.

Remarks

Use an asterisk to indicate that the topic name, item name, or both should be wild.

Exceptions

Exception Type	Condition
ArgumentException	This is thrown when topic or item exceeds 255 characters..
ArgumentNullException	This is thrown when topic or item is a null reference.
InvalidOperationException	This is thrown when the server is not registered.
DdeException	This is thrown when the notification could not be posted.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Disconnect Method

Overload List

This terminates all conversations.

[public virtual void Disconnect\(\);](#)

This terminates the specified conversation.

[public virtual void Disconnect\(DdeConversation\);](#)

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Disconnect Method ()

This terminates all conversations.

```
public virtual void Disconnect();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the server is not registered.
DdeException	This is thrown when the conversations could not be terminated.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) |
[DdeServer.Disconnect Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Disconnect Method (DdeConversation)

This terminates the specified conversation.

```
public virtual void Disconnect(  
    DdeConversation conversation  
);
```

Parameters

conversation

The conversation to terminate.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when conversation is a null reference.
InvalidOperationException	This is thrown when the server is not registered.
DdeException	This is thrown when the conversation could not be terminated.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) |
[DdeServer.Disconnect Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Dispose Method

This unregisters service name and releases all resources held by this instance.

Overload List

This unregisters service name and releases all resources held by this instance.

[public void Dispose\(\);](#)

This contains the implementation to release all resources held by this instance.

[protected virtual void Dispose\(Boolean\);](#)

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Dispose Method ()

This unregisters service name and releases all resources held by this instance.

```
public void Dispose();
```

Implements

[IDisposable.](#)

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Dispose Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Dispose Method (Boolean)

This contains the implementation to release all resources held by this instance.

```
protected virtual void Dispose(  
    Boolean disposing  
);
```

Parameters

disposing

True if called by Dispose, false otherwise.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Dispose Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnAdvise Method

This is invoked when the server is performing a hot advise.

```
protected virtual byte\[\] OnAdvise(  
    String topic,  
    String item,  
    Int32 format  
);
```

Parameters

topic

The topic name associated with this event.

item

The item name associated with this event.

format

The format of the data.

Return Value

The data that will be sent to the clients.

Remarks

The default implementation sends nothing to the clients.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnAfterConnect Method

This is invoked when a client has successfully established a conversation.

```
protected virtual void OnAfterConnect(  
    DdeConversation conversation  
);
```

Parameters

conversation

The conversation associated with this event.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnBeforeConnect Method

This is invoked when a client attempts to establish a conversation.

```
protected virtual bool OnBeforeConnect(  
    String topic  
);
```

Parameters

topic

The topic name associated with this event.

Return Value

True to allow the connection, false otherwise.

Remarks

The default implementation accepts all connections.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnDisconnect Method

This is invoked when a client terminates a conversation.

```
protected virtual void OnDisconnect(  
    DdeConversation conversation  
);
```

Parameters

conversation

The conversation associated with this event.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnExecute Method

This is invoked when a client sends a command.

```
protected virtual ExecuteResult OnExecute(  
    DdeConversation conversation,  
    String command  
);
```

Parameters

conversation

The conversation associated with this event.

command

The command to be executed.

Return Value

An [ExecuteResult](#) indicating the result.

Remarks

The default implementation returns [ExecuteResult.NotProcessed](#) to the client.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnPoke Method

This is invoked when a client sends data.

```
protected virtual PokeResult OnPoke(  
    DdeConversation conversation,  
    String item,  
    Byte\[\] data,  
    Int32 format  
);
```

Parameters

conversation

The conversation associated with this event.

item

The item name associated with this event.

data

The data associated with this event.

format

The format of the data.

Return Value

A [PokeResult](#) indicating the result.

Remarks

The default implementation returns [PokeResult.NotProcessed](#) to the client.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnRequest Method

This is invoked when a client attempts to request data.

```
protected virtual RequestResult OnRequest(  
    DdeConversation conversation,  
    String item,  
    Int32 format  
);
```

Parameters

conversation

The conversation associated with this event.

item

The item name associated with this event.

format

The format of the data.

Return Value

A [RequestResult](#) indicating the result.

Remarks

The default implementation returns [RequestResult.NotProcessed](#) to the client.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnStartAdvise Method

This is invoked when a client attempts to initiate an advise loop.

```
protected virtual bool OnStartAdvise(  
    DdeConversation conversation,  
    String item,  
    Int32 format  
);
```

Parameters

conversation

The conversation associated with this event.

item

The item name associated with this event.

format

The format of the data.

Return Value

True to allow the advise loop, false otherwise.

Remarks

The default implementation accepts all advise loops.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.OnStopAdvise Method

This is invoked when a client terminates an advise loop.

```
protected virtual void OnStopAdvise(  
    DdeConversation conversation,  
    String item  
);
```

Parameters

conversation

The conversation associated with this event.

item

The item name associated with this event.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Pause Method

Overload List

This pauses all conversations.

[public virtual void Pause\(\);](#)

This pauses the specified conversation.

[public virtual void Pause\(DdeConversation\);](#)

Remarks

Pausing a conversation causes this server to queue events until the conversation resumes.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Pause Method ()

This pauses all conversations.

```
public virtual void Pause();
```

Remarks

Pausing a conversation causes this object to queue events until the conversation resumes.

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the server is not registered.
DdeException	This is thrown when the conversations could not be paused.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Pause Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Pause Method (DdeConversation)

This pauses the specified conversation.

```
public virtual void Pause(  
    DdeConversation conversation  
);
```

Parameters

conversation

The conversation to pause.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when conversation is a null reference.
InvalidOperationException	This is thrown when the conversation is already paused or when the server is not registered.
DdeException	This is thrown when the conversation could not be paused.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Pause Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Register Method

This registers the service name.

```
public virtual void Register();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the server is already registered.
DdeException	This is thrown when the service name could not be registered.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Resume Method

Overload List

This resumes all conversations.

[public virtual void Resume\(\);](#)

This resumes the specified conversation.

[public virtual void Resume\(DdeConversation\);](#)

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Resume Method ()

This resumes all conversations.

```
public virtual void Resume();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the server is not registered.
DdeException	This is thrown when the conversations could not be resumed.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Resume Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Resume Method (DdeConversation)

This resumes the specified conversation.

```
public virtual void Resume(  
    DdeConversation conversation  
);
```

Parameters

conversation

The conversation to resume.

Exceptions

Exception Type	Condition
ArgumentNullException	This is thrown when conversation is a null reference.
InvalidOperationException	This is thrown when the conversation is not paused or when the server is not registered.
DdeException	This is thrown when the conversation could not be resumed.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#) | [DdeServer.Resume Overload List](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.Unregister Method

This unregisters the service name.

```
public virtual void Unregister();
```

Exceptions

Exception Type	Condition
InvalidOperationException	This is thrown when the server is not registered.

See Also

[DdeServer Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Structure

This is the return value of the `OnExecute` method.

For a list of all members of this type, see [DdeServer.ExecuteResult Members](#).

[System.Object](#) ValueType
DdeServer.ExecuteResult

```
public struct DdeServer.ExecuteResult
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Server](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeServer.ExecuteResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Members

[DdeServer.ExecuteResult overview](#)

Public Static Fields

 NotProcessed	Return this value if the command was not executed successfully.
 PauseConversation	Return this value to pause the conversation and execute the command asynchronously. After the conversation has been resumed the <code>OnExecute</code> method will run again.
 Processed	Return this value if the command was executed successfully.
 TooBusy	Return this value if the server is too busy.

Public Static Operators

 Equality Operator	This determines whether two <code>ExecuteResult</code> objects are equal.
 Inequality Operator	This determines whether two <code>ExecuteResult</code> objects are not equal.

Public Instance Methods

 Equals	This determines whether two object instances are equal.
 GetHashCode	This returns a hash code for the object.
 GetType (inherited from	Gets the Type of the current

Object)	instance.
ToString (inherited from ValueType)	Returns the fully qualified type name of this instance.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Fields

The fields of the **DdeServer.ExecuteResult** structure are listed below. For a complete list of **DdeServer.ExecuteResult** structure members, see the [DdeServer.ExecuteResult Members](#) topic.

Public Static Fields

 § NotProcessed	Return this value if the command was not executed successfully.
 § PauseConversation	Return this value to pause the conversation and execute the command asynchronously. After the conversation has been resumed the OnExecute method will run again.
 § Processed	Return this value if the command was executed successfully.
 § TooBusy	Return this value if the server is too busy.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.NotProcessed Field

Return this value if the command was not executed successfully.

```
public static readonly NotProcessed NotProces
```

See Also

[DdeServer.ExecuteResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.PauseConversation Field

Return this value to pause the conversation and execute the command asynchronously. After the conversation has been resumed the `OnExecute` method will run again.

```
public static readonly PauseConversation Pause
```

See Also

[DdeServer.ExecuteResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.Processed Field

Return this value if the command was executed successfully.

```
public static readonly Processed Processed;
```

See Also

[DdeServer.ExecuteResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.TooBusy Field

Return this value if the server is too busy.

```
public static readonly TooBusy TooBusy;
```

See Also

[DdeServer.ExecuteResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Methods

The methods of the **DdeServer.ExecuteResult** structure are listed below. For a complete list of **DdeServer.ExecuteResult** structure members, see the [DdeServer.ExecuteResult Members](#) topic.

Public Instance Methods

 Equals	This determines whether two object instances are equal.
 GetHashCode	This returns a hash code for the object.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from ValueType)	Returns the fully qualified type name of this instance.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.Equals Method

This determines whether two object instances are equal.

```
public override bool Equals(  
    Object o  
);
```

Parameters

o

The object to compare with the current object.

Return Value

True if the specified object is equal to the current object, false otherwise.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult.GetHashCode Method

This returns a hash code for the object.

```
public override int GetHashCode();
```

Return Value

A hash code for the object.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Operators

The operators of the **DdeServer.ExecuteResult** structure are listed below. For a complete list of **DdeServer.ExecuteResult** structure members, see the [DdeServer.ExecuteResult Members](#) topic.

Public Static Operators

 Equality Operator	This determines whether two <code>ExecuteResult</code> objects are equal.
 Inequality Operator	This determines whether two <code>ExecuteResult</code> objects are not equal.

See Also

[DdeServer.ExecuteResult Structure](#) | [DdeServer.ExecuteResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Equality Operator

This determines whether two `ExecuteResult` objects are equal.

```
public static bool operator ==(
    DdeServer.ExecuteResult lhs,
    DdeServer.ExecuteResult rhs
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are equal, false otherwise.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.ExecuteResult Inequality Operator

This determines whether two `ExecuteResult` objects are not equal.

```
public static bool operator !=(  
    DdeServer.ExecuteResult lhs,  
    DdeServer.ExecuteResult rhs  
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are not equal, false otherwise.

See Also

[DdeServer.ExecuteResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Structure

This is the return value of the `OnPoke` method.

For a list of all members of this type, see [DdeServer.PokeResult Members](#).

[System.Object](#) ValueType
DdeServer.PokeResult

```
public struct DdeServer.PokeResult
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Server](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeServer.PokeResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Members

[DdeServer.PokeResult overview](#)

Public Static Fields

 NotProcessed	Return this value if the poke was not successful.
 PauseConversation	Return this value to pause the conversation and execute the poke asynchronously. After the conversation has been resumed the OnPoke method will run again.
 Processed	Return this value if the poke was successful.
 TooBusy	Return this value if the server is too busy.

Public Static Operators

 Equality Operator	This determines whether two PokeResult objects are equal.
 Inequality Operator	This determines whether two ExecuteResult objects are not equal.

Public Instance Methods

 Equals	This determines whether two object instances are equal.
 GetHashCode	This returns a hash code for the object.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from	Returns the fully qualified type

ValueType)

name of this instance.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Fields

The fields of the **DdeServer.PokeResult** structure are listed below. For a complete list of **DdeServer.PokeResult** structure members, see the [DdeServer.PokeResult Members](#) topic.

Public Static Fields

 NotProcessed	Return this value if the poke was not successful.
 PauseConversation	Return this value to pause the conversation and execute the poke asynchronously. After the conversation has been resumed the OnPoke method will run again.
 Processed	Return this value if the poke was successful.
 TooBusy	Return this value if the server is too busy.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.NotProcessed Field

Return this value if the poke was not successful.

```
public static readonly NotProcessed NotProces
```

See Also

[DdeServer.PokeResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.PauseConversation Field

Return this value to pause the conversation and execute the poke asynchronously. After the conversation has been resumed the `OnPoke` method will run again.

```
public static readonly PauseConversation Pause
```

See Also

[DdeServer.PokeResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.Processed Field

Return this value if the poke was successful.

```
public static readonly Processed Processed;
```

See Also

[DdeServer.PokeResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.TooBusy Field

Return this value if the server is too busy.

```
public static readonly TooBusy TooBusy;
```

See Also

[DdeServer.PokeResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Methods

The methods of the **DdeServer.PokeResult** structure are listed below. For a complete list of **DdeServer.PokeResult** structure members, see the [DdeServer.PokeResult Members](#) topic.

Public Instance Methods

 Equals	This determines whether two object instances are equal.
 GetHashCode	This returns a hash code for the object.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from ValueType)	Returns the fully qualified type name of this instance.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.Equals Method

This determines whether two object instances are equal.

```
public override bool Equals(  
    Object o  
);
```

Parameters

o

The object to compare with the current object.

Return Value

True if the specified object is equal to the current object, false otherwise.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult.GetHashCode Method

This returns a hash code for the object.

```
public override int GetHashCode();
```

Return Value

A hash code for the object.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Operators

The operators of the **DdeServer.PokeResult** structure are listed below. For a complete list of **DdeServer.PokeResult** structure members, see the [DdeServer.PokeResult Members](#) topic.

Public Static Operators

 Equality Operator	This determines whether two <code>PokeResult</code> objects are equal.
 Inequality Operator	This determines whether two <code>ExecuteResult</code> objects are not equal.

See Also

[DdeServer.PokeResult Structure](#) | [DdeServer.PokeResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Equality Operator

This determines whether two `PokeResult` objects are equal.

```
public static bool operator ==(
    DdeServer.PokeResult lhs,
    DdeServer.PokeResult rhs
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are equal, false otherwise.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.PokeResult Inequality Operator

This determines whether two `ExecuteResult` objects are not equal.

```
public static bool operator !=(  
    DdeServer.PokeResult lhs,  
    DdeServer.PokeResult rhs  
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are not equal, false otherwise.

See Also

[DdeServer.PokeResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Structure

This is the return value of the `OnRequest` method.

For a list of all members of this type, see [DdeServer.RequestResult Members](#).

[System.Object](#) ValueType
DdeServer.RequestResult

```
public struct DdeServer.RequestResult
```

Thread Safety

Public static (**Shared** in Visual Basic) members of this type are safe for multithreaded operations. Instance members are **not** guaranteed to be thread-safe.

Requirements

Namespace: [NDde.Server](#)

Assembly: NDde (in NDde.dll)

See Also

[DdeServer.RequestResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Members

[DdeServer.RequestResult overview](#)

Public Static Fields

 NotProcessed	Return this value if the request was not successful.
 PauseConversation	Return this value to pause the conversation and execute the request asynchronously. After the conversation has been resumed the OnRequest method will run again.

Public Static Operators

 Equality Operator	This determines whether two RequestResult objects are equal.
 Inequality Operator	This determines whether two ExecuteResult objects are not equal.

Public Instance Constructors

 DdeServer.RequestResult Constructor	This initializes the RequestResult struct with the data to return to the client.
---	--

Public Instance Properties

 Data	The data to send to the client application.
--	---

Public Instance Methods

 Equals	This determines whether two object instances are equal.
--	---

GetHashCode	This returns a hash code for the object.
GetType (inherited from Object)	Gets the Type of the current instance.
ToString (inherited from ValueType)	Returns the fully qualified type name of this instance.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Constructor

This initializes the `RequestResult` struct with the data to return to the client.

```
public DdeServer.RequestResult(  
    Byte[] data  
);
```

Parameters

data

The data to return to the client.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Fields

The fields of the **DdeServer.RequestResult** structure are listed below. For a complete list of **DdeServer.RequestResult** structure members, see the [DdeServer.RequestResult Members](#) topic.

Public Static Fields

 S NotProcessed	Return this value if the request was not successful.
 S PauseConversation	Return this value to pause the conversation and execute the request asynchronously. After the conversation has been resumed the OnRequest method will run again.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult.NotProcessed Field

Return this value if the request was not successful.

```
public static readonly NotProcessed NotProces
```

See Also

[DdeServer.RequestResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult.PauseConversation Field

Return this value to pause the conversation and execute the request asynchronously. After the conversation has been resumed the `OnRequest` method will run again.

```
public static readonly PauseConversation Pause
```

See Also

[DdeServer.RequestResult Class](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Properties

The properties of the **DdeServer.RequestResult** structure are listed below. For a complete list of **DdeServer.RequestResult** structure members, see the [DdeServer.RequestResult Members](#) topic.

Public Instance Properties

 Data	The data to send to the client application.
--	---

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult.Data Property

The data to send to the client application.

```
public Byte[] Data {get; set;}
```

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Methods

The methods of the **DdeServer.RequestResult** structure are listed below. For a complete list of **DdeServer.RequestResult** structure members, see the [DdeServer.RequestResult Members](#) topic.

Public Instance Methods

 Equals	This determines whether two object instances are equal.
 GetHashCode	This returns a hash code for the object.
 GetType (inherited from Object)	Gets the Type of the current instance.
 ToString (inherited from ValueType)	Returns the fully qualified type name of this instance.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult.Equals Method

This determines whether two object instances are equal.

```
public override bool Equals(  
    Object o  
);
```

Parameters

o

The object to compare with the current object.

Return Value

True if the specified object is equal to the current object, false otherwise.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult.GetHashCode Method

This returns a hash code for the object.

```
public override int GetHashCode();
```

Return Value

A hash code for the object.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Operators

The operators of the **DdeServer.RequestResult** structure are listed below. For a complete list of **DdeServer.RequestResult** structure members, see the [DdeServer.RequestResult Members](#) topic.

Public Static Operators

 Equality Operator	This determines whether two <code>RequestResult</code> objects are equal.
 Inequality Operator	This determines whether two <code>ExecuteResult</code> objects are not equal.

See Also

[DdeServer.RequestResult Structure](#) | [DdeServer.RequestResult Members](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Equality Operator

This determines whether two `RequestResult` objects are equal.

```
public static bool operator ==(
    DdeServer.RequestResult lhs,
    DdeServer.RequestResult rhs
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are equal, false otherwise.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

DdeServer.RequestResult Inequality Operator

This determines whether two `ExecuteResult` objects are not equal.

```
public static bool operator !=(  
    DdeServer.RequestResult lhs,  
    DdeServer.RequestResult rhs  
);
```

Parameters

lhs

The left hand side object.

rhs

Return Value

True if the two objects are not equal, false otherwise.

See Also

[DdeServer.RequestResult Structure](#) | [NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Generated from assembly NDde [2.1.563.0]

Hierarchy

[System.Object](#)

[System.EventArgs](#)

[NDde.DdeEventArgs](#)

[System.Exception](#) ---- [System.Runtime.Serialization.ISerializable](#)

[NDde.DdeException](#)

See Also

[NDde Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Hierarchy

[System.Object](#)

[NDde.Advanced.DdeContext](#) ----

[System.ComponentModel.ISynchronizeInvoke](#),
[System.IDisposable](#)

[NDde.Advanced.DdeMessageLoop](#) ----

[System.ComponentModel.ISynchronizeInvoke](#),
[System.IDisposable](#)

[NDde.Advanced.DdeTransaction](#)

[NDde.Advanced.IDdeTransactionFilter](#)

[System.EventArgs](#)

[NDde.DdeEventArgs](#)

[NDde.Advanced.DdeRegistrationEventArgs](#)

See Also

[NDde.Advanced Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Hierarchy

[System.Object](#)

[NDde.Advanced.Monitor.DdeMonitor](#) ---- [System.IDisposable](#)

[System.EventArgs](#)

[NDde.DdeEventArgs](#)

[NDde.Advanced.Monitor.DdeActivityEventArgs](#)

[NDde.Advanced.Monitor.DdeCallbackActivityEventArgs](#)

[NDde.Advanced.Monitor.DdeConversationActivityEventArgs](#)

[NDde.Advanced.Monitor.DdeErrorActivityEventArgs](#)

[NDde.Advanced.Monitor.DdeLinkActivityEventArgs](#)

[NDde.Advanced.Monitor.DdeMessageActivityEventArgs](#)

[System.ValueType](#)

[System.Enum](#) ---- [System.IComparable](#), [System.IConvertible](#),
[System.IFormattable](#)

[NDde.Advanced.Monitor.DdeMessageActivityKind](#)

[NDde.Advanced.Monitor.DdeMonitorFlags](#)

See Also

[NDde.Advanced.Monitor Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Hierarchy

[System.Object](#)

[NDde.Client.DdeClient](#) ---- [System.IDisposable](#)

[System.EventArgs](#)

[NDde.DdeEventArgs](#)

[NDde.Client.DdeAdviseEventArgs](#)

[NDde.Client.DdeDisconnectedEventArgs](#)

See Also

[NDde.Client Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)

Hierarchy

[System.Object](#)

[NDde.Server.DdeConversation](#)

[NDde.Server.DdeServer](#) ---- [System.IDisposable](#)

[System.ValueType](#)

[NDde.Server.DdeServer.ExecuteResult](#)

[NDde.Server.DdeServer.PokeResult](#)

[NDde.Server.DdeServer.RequestResult](#)

See Also

[NDde.Server Namespace](#)

[Send comments on this topic.](#)

Copyright (c) 2005-2006 by Brian Gideon (briangideon@yahoo.com)