

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UBound Property

[See Also](#) [Example](#) [Applies To](#)

Returns the highest ordinal value of a control in a control array.

Syntax

object.**UBound**

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

Remarks

UBound is equal to the **Index** property value of the last control in the array.

© 2018 Microsoft

Visual Basic Reference

LBound, UBound Properties Example

This example prints the values of these two properties for a control array. Put an **OptionButton** control on a form, and set its **Index** property to 0 (to create a control array). To try this example, paste the code into the Declarations section of a form, and then press F5 and click the form.

```
Private Sub Form_Paint ()
    Static FlagFormPainted As Integer
    If FlagFormPainted <> True Then ' When form is painting for first time,
        For i = 1 To 3
            Load Option1(i) ' add three option buttons to array.
            Option1(i).Top = Option1(i - 1).Top + 350
            Option1(i).Visible = True
        Next I
        For I = 0 to 3 ' Put captions on the option buttons.
            Option1(i).Caption = "Option #" & CStr(i)
        Next I
        Option1(0).Value = True ' Select first option button.
        FlagFormPainted = True ' Form is done painting.
    End If
End Sub

Private Sub Form_Click ()
    Print "Control array's Count property is " & Option1().Count
    Print "Control array's LBound property is " & Option1().LBound
    Print "Control array's UBound property is " & Option1().UBound
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UIDead Property

See Also Example [Applies To](#)

Returns a boolean value indicating whether the control should be responsive to the user or not.

Syntax

object.**UIDead**

The **UIDead** property syntax has this part:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.

Settings

The possible boolean return values from the **UIDead** property are:

Setting	Description
True	The control should not respond to the user.
False	The control should respond to the user. If the container does not implement this ambient property, this will be the default value.

Remarks

This property is typically used to indicate that the container is in break mode: during this mode, the control should not respond to any user input. That is, the control should ignore mouse clicks and keystrokes, and not change the mouse cursor even when the mouse is over the control window. A container such as a Visual Basic form would set this flag to TRUE when the programmer stops the program during execution the container is not in design mode, yet not in run mode either; Visual Basic simply wants the controls to be inoperative.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UIDefault Property

See Also Example [Applies To](#)

Returns or sets the UIDefault attribute of a **Member** object.

Syntax

object.UIDefault

The *object* placeholder represents an object expression that evaluates to an object in the Applies To list.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UnattendedApp Property

See Also Example [Applies To](#)

Returns a value that determines if an application will run without any user interface.

Syntax

object.**UnattendedApp**= *boolean*

The **UnattendedApp** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression that specifies if the application will run without any user interface.

Settings

The settings for *boolean* are:

Constant	Value	Description
True	-1	The application has no user interface.
False	0	The application has a user interface.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Underline Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets the font style of the **Font** object to either underlined or nonunderlined.

Syntax

object.**Underline** [= *boolean*]

The **Underline** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression specifying the font style, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Turns on underline formatting.
False	(Default) Turns off underline formatting.

Remarks

The **Font** object isn't directly available at design time. Instead you set the **Underline** property by selecting a control's **Font** property in the Properties window and clicking the Properties button. In the Font dialog box, select the Underline check box. At [run time](#), however, you set **Underline** directly by specifying its setting for the **Font** object.

Visual Basic Reference

Bold, Italic, Size, StrikeThrough, Underline, Weight Properties Example

This example prints text on a form with each mouse click. To try this example, paste the code into the Declarations section of a form, and then press F5 and click the form twice.

```
Private Sub Form_Click ()
    Font.Bold = Not Font.Bold    ' Toggle bold.
    Font.StrikeThrough = Not Font.StrikeThrough ' Toggle strikethrough.
    Font.Italic = Not Font.Italic    ' Toggle italic.
    Font.Underline = Not Font.Underline    ' Toggle underline.
    Font.Size = 16    ' Set Size property.
    If Font.Bold Then
        Print "Font weight is " & Font.Weight & " (bold)."Weight & " (not bold)."
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

UniformAxis Property

See Also Example [Applies To](#)

Returns or sets a value that specifies whether the unit scale for all value axes in a chart is uniform.

Syntax

object.**UniformAxis** [= *boolean*]

The **UniformAxis** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression that controls the unit scale, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) The unit scale for all value axes is uniform.
False	The unit scale is not uniform. The unit scale is determined by the plot size and positioning set according to the AutoLayout or LocationRect property. If AutoLayout is True , the plot size and position are based on the size and position of other automatically laid out elements. If False , the coordinates specified by LocationRect are used to position the plot and determine the axes unit scale.

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

Updatable Property (Remote Data)

[See Also](#) [Example](#) [Applies To](#)

Returns a [Boolean](#) value that indicates whether changes can be made to a [remote data object](#).

Syntax

object.**Updatable**

The *object* placeholder represents an [object expression](#) that evaluates to an object in the Applies To list.

Return Values

The **Updatable** property return values are:

Value	Description
True	The object can be changed or updated.
False	The object can't be changed or updated. This is the only setting for static-type rdoResultset objects.

Remarks

If the **Updatable** property setting is **True**, the specified:

- **rdoConnection** object refers to an updatable [data source](#).
- **rdoQuery** object refers to an updatable [result set](#).
- **rdoResultset** contains updatable [rows](#).
- **rdoTable** object refers to a [table](#) whose data can be changed through use of a query.
- **rdoColumn** object refers to data that can be changed. Only **rdoColumn** objects which are part of an **rdoResultset** object can be changed.

You can use the **Updatable** property with all types of **rdoResultset** objects.

Many types of **rdoResultset** objects can contain columns that can't be updated. For example, you can create a [forward-only rdoResultset](#) that is derived from nonupdatable sources or that contains computed or derived columns.

If the object contains only nonupdatable columns, the value of the **Updatable** property is **False**. When one or more columns are updatable, the property's value is **True**. You can edit only the updatable columns. A trappable error occurs if you try to

assign a new value to a nonupdatable column.

Because an updatable object can contain columns that cannot be updated, check the **Updatable** property of each **rdoColumn** before editing a row in the **rdoResultset**.

Even when a cursor cannot be updated, it might still be possible to update the data through use of an action query. In many cases, database tables are protected and not updatable by design as they are protected from direct access by the system administrator and the remote system's permission scheme. If this is the case, check with your system administrator for the availability of stored procedures or special login accounts that permit you to perform your changes.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

UpdateCriteria Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value that specifies how the WHERE clause is constructed for each row during an optimistic batch update operation.

Syntax

object.**UpdateCriteria** [= *value*]

The **UpdateCriteria** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A Long integer representing the type of cursor as described in Settings

Settings

The **UpdateCriteria** property has these settings:

Constant	Value	rdoResultset type
rdCriteriaKey	0	(Default). Uses just the key column(s) in the where clause.
rdCriteriaAllCols	1	Uses the key column(s) and all updated columns in the where clause.
rdCriteriaUpdCols	2	Uses the key column(s) and all the columns in the where clause.
rdCriteriaTimeStamp	3	Uses just the timestamp column if available (will generate a runtime error if no timestamp column is in the result set).

Remarks

When a batch mode operation is executed, RDO and the ClientBatch cursor library create a series of UPDATE statements to make the needed changes. An SQL WHERE clause is created for each update to isolate the rows that are marked as changed (by the **Status** property). Because some remote servers use triggers or other ways to enforce referential integrity, is it often important to limit the columns being updated to just those affected by the change. This way, only the absolute minimum

amount of trigger code is executed. As a result, the update operation is executed more quickly, and with fewer potential errors.

You should set the **UpdateCriteria** property to **rdCriteriaKey** when BLOB columns are included in the result set.

Setting this property to a value other than the ones listed here results in a runtime error.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

UpdateCriteria Property Constants

[See Also](#)

These constants are used to determine how RDO manages update operations when the **BatchUpdate** method is used.

Constant	Value	Description
rdCriteriaKey	0	Uses <i>just</i> the key column(s) in the SQL WHERE clause.
rdCriteriaAllCols	1	Uses the key column(s) <i>and</i> all updated columns in the SQL WHERE clause.
rdCriteriaUpdCols	2	Uses the key column(s) and <i>all</i> columns in the SQL WHERE clause.
rdCriteriaTimeStamp	3	Uses just the timestamp column in the SQL WHERE clause. (Generates a trappable error if no timestamp column is available).

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

UpdateInterval Property (Multimedia MCI Control)

[See Also](#) [Example](#) [Applies To](#)

Specifies the number of milliseconds between successive StatusUpdate events.

Syntax

```
[form.]MMControl.UpdateInterval[ = milliseconds%]
```

Remarks

The argument *milliseconds%* specifies the number of milliseconds between events. If milliseconds is 0, no StatusUpdate events occur.

Data Type

Integer

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UpdateMode Property

See Also Example [Applies To](#)

Returns or sets the update mode of the collection.

Syntax

object.**UpdateMode** [= *integer*]

The **UpdateMode** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>integer</i>	A value or constant that specifies the update mode, as shown in Settings.

Settings

The settings for *integer* are:

Constant	Value	Description
vbUsePropertyAttributes	0	The recordset updates according to the settings found in the Data Binding pane of the Property Attributes dialog box.
vbUpdateWhenPropertyChanges	1	The recordset updates when the property's value changes.
vbUpdateWhenRowChanges	2	The recordset updates only when the user leaves the row.

Remarks

Whether or not a data source is updateable depends on the data provider. For example, if the data source is an **ADO** Recordset opened with its **LockType** property set to **adLockReadOnly**, it cannot be updated.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

UpdateOperation Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value that specifies if the optimistic batch update should use an Update statement or a Delete followed by an Insert.

Syntax

object.**UpdateOperation** [= *value*]

The **UpdateOperation** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A Long integer representing the type of cursor as described in Settings:

Settings

The **UpdateOperation** property has these settings:

Constant	Value	UpdateOperation type
rdOperationUpdate	0	(Default) Uses an Update statement for each modified row.
rdOperationDelIns	1	Uses a pair of Delete and Insert statements for each modified row.

Remarks

Setting the **UpdateOperation** property to a value other than the ones listed here results in a runtime error.

This property determines whether the optimistic batch update cursor library uses an update statement, or a pair of delete and insert statements when sending modifications back to the database server. In the latter case, two separate operations are required to update the row. In some cases, especially where the remote system implements Delete, Insert and Update triggers, choosing the correct **UpdateOperation** property can significantly impact performance.

Newly added rows will always generate Insert statements and deleted rows will always generate Delete statements, so this property only applies to how the cursor library updates modified rows.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UpdateOptions Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value specifying how an object is updated when linked data is modified.

Syntax

object.**UpdateOptions** [= *number*]

The **UpdateOptions** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>number</i>	A integer specifying how an object is updated, as described in Settings.

Settings

The settings for *number* are:

Constant	Value	Description
vbOLEAutomatic	0	(Default) Automatic. The object is updated each time the linked data changes.
vbOLEFrozen	1	Frozen. The object is updated whenever the user saves the linked data from within the application in which it was created.
vbOLEManual	2	Manual. The object is updated only by using the Update method.

Remarks

This property is useful for linked objects where other users or applications can access and modify the linked data.

When an object's data is changed, the Updated event is invoked.

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

UpDown Property

See Also [Example](#) [Applies To](#)

Returns or sets a value that determines if an up-down control appears to the right of the DateTimePicker control.

Syntax

object.**UpDown** [= *boolean*]

The **UpDown** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression specifying whether or not an up-down control appears to the right of the DateTimePicker control.

Settings

The settings for *boolean* are:

Setting	Description
True	An up-down control appears.
False	(Default) An up-down control does not appear.

Remarks

The up-down buttons can be used instead of the drop-down calendar to select a date.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Internet Control

Visual Studio 6.0

URL Property

See Also Example [Applies To](#)

Returns or sets the URL that is used by the **Execute** or **OpenURL** methods.

Syntax

object.**URL** [= *url*]

The **URL** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>url</i>	String that specifies the URL to be used in the Execute method.

Data Type

String

Remarks

Invoking the **OpenURL** or **Execute** method changes the value of this property.

Changing this property will have no effect until the next **OpenURL** or **Execute** method is called.

The **URL** property must contain at least a protocol and remote host name.

The **URL** property can be either a directory or a file. For example, both of the URLs below are valid:

```
' Setting this URL will return only the file directory:  
Inet1.URL = "HTTP://www.microsoft.com"  
' However, this URL will return the text of the file:  
Inet1.URL = "HTTP://www.microsoft.com/disclaimer.txt"
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

URLData Property

[See Also](#) [Example](#) [Applies To](#)

Sets or returns a parameter that is appended to the end of any URL generated using the **URLFor** method or discovered during **WriteTemplate** processing.

Syntax

object.**URLData**[=*string*]

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>string</i>	Information that is appended to the end of URLs generated by a WebClass.

Remarks

This information is appended automatically to each URL discovered during HTML template processing using the **WriteTemplate** method and by calls to the **URLFor** method. You can store state information in the appended parameter. The state can be retrieved in subsequent calls using the Request.QueryString collection.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

UseCoolbarColors Property

See Also Example [Applies To](#)

Sets or returns whether a **Band** object will use the **CoolBar** controls **ForeColor** and **BackColor** properties.

Syntax

object.UseCoolbarColors [= *boolean*]

The **UseCoolbarColors** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to a Band object.
<i>boolean</i>	A Boolean expression specifying whether the band will override the CoolBar controls ForeColor and BackColor .

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Band will display using the CoolBar controls ForeColor and BackColor properties.
False	Band will display using its own ForeColor and BackColor properties.

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

UseCoolbarPicture Property

[See Also](#) [Example](#) [Applies To](#)

Sets or returns whether a **Band** object will use the **CoolBar** controls **Picture** property.

Syntax

object.**UseCoolbarPicture** [= *boolean*]

The **UseCoolbarPicture** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to a Band object.
<i>boolean</i>	A Boolean expression specifying whether the bands Picture property will override the CoolBar controls Picture .

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Band will display the CoolBar controls Picture .
False	Band will display its own Picture .

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

UseMaskColor Property (ImageList Controls)

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value that determines whether the color assigned in the **MaskColor** property is used as a mask.

Syntax

object.**UseMaskColor** [= *value*]

The **UseMaskColor** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A boolean value that determines whether the MaskColor property is used, as described in Settings.

Settings

The settings for *value* are:

Value	Description
True	The MaskColor is used to create transparent regions.
False	(Default) The MaskColor is not used.

Remarks

The **MaskColor** and **UseMaskColor** properties are used only when the **Style** property is set to 1 - Graphical.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UseMaskColor Property

See Also Example [Applies To](#)

Returns or sets a value that determines whether the color assigned in the **MaskColor** property is used as a mask. (That is, used to create transparent regions.)

Syntax

object.**UseMaskColor** [= *boolean*]

The **UseMaskColor** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression that specifies whether the color assigned to the MaskColor property is used as a mask.

Settings

The settings for *boolean* are:

Setting	Description
True	The color assigned to the MaskColor property is used as a mask, creating a transparent region wherever that color is.
False	(Default) The color assigned to the MaskColor property is ignored, and the color remains opaque.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UseMnemonic Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value that specifies whether an ampersand (&) included in the text of the **Caption** property of the **Label** control defines an access key.

Syntax

object.**UseMnemonic** [= *boolean*]

The **UseMnemonic** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>boolean</i>	A Boolean expression specifying whether the Label control enables an access key, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Any ampersand appearing in the text of the Caption property causes the character following the ampersand to become an access key. The ampersand itself isn't displayed in the interface of the Label control.
False	Any ampersand appearing in the text of the Caption property is displayed as an ampersand in the interface of the Label control.

Remarks

At [run time](#), pressing ALT+ the access key defined in the **Label** control's **Caption** property moves [focus](#) to the control that follows the **Label** control in the tab order.

Visual Basic Reference

UseMnemonic Property Example

This example reads the setting of the **UseMnemonic** property of a **Label** control. To try this example, paste the code into the Declarations section of a form that contains a **Label**, and then press F5 and click the form.

```
Private Sub Form_Click()  
    If Label1.UseMnemonic And InStr(Label1, "&") Then  
        MsgBox "The label has an access key character."  
    ElseIf Label1.UseMnemonic And Not InStr(Label1, "&") Then  
        MsgBox "The label supports an access key character but doesn't have an ampersand."  
    Else  
        MsgBox "The label doesn't support an access key character."  
    End If  
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

UserMode Property

See Also Example [Applies To](#)

Returns a boolean value indicating whether the control is being used by a form designer or a form user.

Syntax

object.**UserMode**

The **UserMode** property syntax has this part:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.

Settings

The possible boolean return values from the **UserMode** property are:

Setting	Description
True	The control is currently being used by a form user. If the container does not implement this ambient property, this will be the default value. In Visual Basic, this is Run Mode.
False	The control is currently being used by a form designer (the developer). In Visual Basic, this is Design Mode.

This documentation is archived and is not being maintained.

Visual Basic: ADO Data Control

Visual Studio 6.0

UserName Property (ADO Data Control)

See Also Example Applies To

Returns or sets a value that represents a user of an **ADO Recordset** object.

Syntax

object.**UserName** [= *value*]

The **UserName** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A string expression that contains a user name as described in Settings.

Settings

The user-name syntax depends on the data source.

Remarks

When a **UserName** and **Password** are supplied, the control uses the values to create a connection string (**ConnectionString** property).

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: MAPI Controls

Visual Studio 6.0

UserName Property

[See Also](#) [Example](#) [Applies To](#)

Specifies the account user name, or the Profile to be used to establish a session.

Syntax

```
object.UserName [ = value ]
```

The **UserName** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A string expression specifying the user account.

Remarks

On computers that have Microsoft Exchange or Microsoft Outlook installed, this property sets the Profile to be used when establishing a session. Profiles include the actual user name and password to be used. New profiles are set up by clicking on the **Mail and Fax** icon in the Windows Control Panel.

On computers that use Microsoft Mail, this property contains the name of the user account desired for sign-on or sign-off. If the **LogonUI** property is **True**, an empty string in the **UserName** property indicates that a sign-on dialog box with an empty name field should be generated. The default is an empty string.

Data Type

String

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

UserName Property (Remote Data)

[See Also](#) [Example](#) [Applies To](#)

Returns or sets a value that represents a user of an **rdoEnvironment** object. Use the **UserName** property with the **Password** property to connect to an ODBC [data source](#).

Syntax

object.**UserName** [= *value*]

The **UserName** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>value</i>	A string expression that contains a user name as described in Settings. (Data type is String .)

Settings

The user name syntax depends on the ODBC data source.

Return Values

The **UserName** property represents the user of an **rdoEnvironment** object. The user name is set when the **rdoEnvironment** is either created automatically by the **RemoteData** control, by the first reference to a [remote data object](#), or when the **rdoCreateEnvironment** method is executed.

You can determine the default user name with the **rdoDefaultUser** property of the **rdoEngine** object. If no specific user name is supplied in **UserName**, the value of the **rdoDefaultUser** property is used.

This documentation is archived and is not being maintained.

Visual Basic: Internet Control

Visual Studio 6.0

UserName Property

[See Also](#) [Example](#) [Applies To](#)

Returns or sets the name that will be sent with requests to remote computers. If this property is left blank, the control will send "anonymous" as the user name when requests are made.

Syntax

object.**UserName**[= *name*]

The **UserName** property syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>name</i>	String that specifies the UserName to be used in the Execute method.

Data Type

String

Remarks

Invoking the **OpenURL** or **Execute** method changes the value of this property.

Changing this property will have no effect until the next **OpenURL** or **Execute** method is called.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

UsesWindows Property (Multimedia MCI Control)

[See Also](#) [Example](#) [Applies To](#)

Determines if the currently open MCI device uses a window for output. This property is not available at design time and is read-only at run time.

Syntax

*[form.]*MMControl.**UsesWindows**

Remarks

The following table lists the **UsesWindows** property return values for the **Multimedia MCI** control.

Value	Description
False	The current device does not use a window for output.
True	The current device uses a window.

Currently, only MMMovie and Overlay devices use windows for display. Because there is no way to determine whether a device uses windows, the value of **UsesWindows** is set during processing of an **Open** command by checking the device type. If the device type is MMMovie, Overlay, or VCR, the device uses windows.

For devices that use windows, you can use the **hWndDisplay** property to set the window that will display output.

Data Type

Integer (Boolean)

© 2018 Microsoft