

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

EdgeIntensity Property

See Also Example [Applies To](#)

Returns or sets the intensity of light used to draw the edges of objects in a three-dimensional chart.

Syntax

object.**EdgeIntensity** [= *edgeint*]

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

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>edgeint</i>	Single. The edge light intensity. Valid values are 0 to 1.0. An intensity of 0 turns edges off, drawing the edges as black lines; and an intensity of 1 fully illuminates the edges using the element's pen color.

Remarks

If this property is set, then the **Light** object's **EdgeVisible** property is automatically set to **True**.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

EdgePen Property

See Also Example [Applies To](#)

Returns the **Pen** object used to draw the edge of the data point on a chart.

Syntax

object.**EdgePen**

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

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

EdgeVisible Property

See Also Example [Applies To](#)

Returns or sets a value that determines whether edges are displayed on the elements in a three-dimensional chart.

Syntax

object.**EdgeVisible** [= *boolean*]

The **EdgeVisible** 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 edges are displayed, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	Edges are visible.
False	Edges are not displayed on elements in the three-dimensional chart.

This documentation is archived and is not being maintained.

Visual Basic: DataGrid Control

Visual Studio 6.0

EditActive Property

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

Sets or returns the editing status of the current cell. Not available at design time.

Syntax

object.**EditActive** [= *value*]

The **EditActive** 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 expression that determines the editing status, as described in Settings.

Settings

The settings for *value* are:

Setting	Description
True	The current cell is currently being edited by the user.
False	No editing is in progress.

Remarks

If the grid is not already in edit mode, setting **EditActive** to **True** will initiate editing on the current cell. The caret will be positioned at the end of the cell and the ColEdit event will be triggered.

If the grid is already in edit mode, setting **EditActive** to **False** will exit edit mode. If the cell has been modified, this will trigger the following events: BeforeColUpdate, AfterColUpdate, and AfterColEdit.

Note To cancel editing completely, set the **CurrentCellModified** property to **False**, then set **EditActive** to **False**.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

EditAtDesignTime Property

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

Returns or sets a value determining if a control can become active during the developer design time. The **EditAtDesignTime** property is read/write at the controls authoring time, and not available at the controls run time.

Settings

The settings for **EditAtDesignTime** are:

Setting	Description
True	Allows the control to become active at the developer design time. An Edit verb will appear on the controls context menu. When the developer who uses the control chooses Edit , the control will become active and behave as it does at end user run time.
False	(Default) The control cannot become active at developer design time.

Remarks

The control will only remain active while it is selected. When the developer selects another control, this control will no longer be active, even if the developer clicks back on this control. The developer must select **Edit** again from the context menu to make the control active.

Note When the control is activated in this fashion, the events of the **UserControl** object will occur, so that the control can operate normally, but the control will be unable to raise any events. The **RaiseEvent** method will simply be ignored; it will not cause an error.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

EditMode Property (Remote Data)

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

Returns a value that indicates the state of editing for the [current row](#).

Syntax

object.**EditMode**

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

Return Values

The **EditMode** property returns an [Integer](#) or constant as described in the following table:

Constant	Value	Description
rdEditNone	0	No editing operation is in progress.
rdEditInProgress	1	The Edit method has been invoked, and the current row is in the copy buffer .
rdEditAdd	2	The AddNew method has been invoked, and the current row in the copy buffer is a new row that hasn't been saved in the database .

Remarks

The **EditMode** property is most useful when you want to depart from the default functionality of a **RemoteData** control. You can check the value of the **EditMode** property and the value of the *action* parameter in the Validate event procedure to determine whether to invoke the **UpdateRow** method.

You can also check to see if the **LockEdits** property of the **rdoResultset** is **True** and the **EditMode** property setting is **rdEditInProgress** to determine whether the current [data page](#) is locked.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

Effect Property

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

Returns or sets the font effects in a chart.

Syntax

object.**Effect** [= *effects*]

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

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>effects</i>	Integer. A VtFontEffect constant describing the font effect.

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

Elevation Property

See Also Example [Applies To](#)

Returns or sets a value that describes the degree of elevation from which a three-dimensional chart is viewed.

Syntax

object.**Elevation** [= *degree*]

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

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>degree</i>	Single. The degree of elevation. Elevation can be any number from 0 to 90 degrees. If you set the elevation to 90 degrees, you look directly down onto the top of the chart. If you set the elevation to 0, you look directly at the side of the chart. The default elevation is 30 degrees. By default, degrees are used to measure elevation. However, these settings use the current settings for the AngleUnit property . The other options are: Grads and Radians.

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

EmbossHighlight Property

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

Sets or returns the highlight color for an image displayed on a **CoolBar** control.

Syntax

object.**EmbossHighlight** [= *color*]

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

Part	Description
<i>object</i>	An object expression that evaluates to a CoolBar control.
<i>color</i>	A value or constant that determines the background or foreground colors of an object, as described in Settings.

Settings

Visual Basic uses the Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for *color* are:

Setting	Description
Normal RGB colors	Colors specified by using the Color palette or by using the RGB or QBColor functions in code.
System default colors	Colors specified by system color constants listed in the Visual Basic (VB) object library in the Object Browser. The Windows operating environment substitutes the user's choices as specified in the Control Panel settings.

Remarks

The default setting for the **EmbossHighlight** property is the system default color specified by the constant **vbButtonHighlight**.

The valid range for a normal RGB color is 0 to 16,777,215 (&HFFFFFF). The high byte of a number in this range equals 0; the lower 3 bytes, from least to most significant byte, determine the amount of red, green, and blue, respectively. The red, green,

and blue components are each represented by a number between 0 and 255 (&HFF). If the high byte isn't 0, Visual Basic uses the system colors, as defined in the user's Control Panel settings and by constants listed in the Visual Basic (VB) [object library](#) in the Object Browser.

Note This property has no effect on the display of an image unless the **Picture** property of the **CoolBar** control contains a valid image and the **EmbossPicture** property is set to **True**.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

EmbossPicture Property

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

Returns or sets a value indicating whether the background picture on a CoolBar control will be displayed in two colors or multiple colors.

Syntax

object.**EmbossPicture** [= *boolean*]

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

Part	Description
<i>object</i>	An object expression that evaluates to a CoolBar control.
<i>boolean</i>	A Boolean expression specifying whether the background picture is displayed in two colors or in multiple colors.

Settings

The settings for *boolean* are:

Setting	Description
False	(Default) Background picture is displayed in multiple colors.
True	Background picture is displayed in two colors.

Remarks

By default, the background picture on a **CoolBar** control will be displayed using the colors contained in the image file assigned to the **Picture** property. When the **EmbossPicture** property is **True**, the background picture will be dithered to the two colors specified in the **EmbossHighlight** and **EmbossShadow** properties.

Note This property has no effect unless the **Picture** property of the **CoolBar** control contains a valid image. If the **UseCoolbarPicture** property of a contained **Band** object is set to **False**, this property has no effect on that **Band**.

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

EmbossShadow Property

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

Sets or returns the shadow color for an image displayed on a **CoolBar** control.

Syntax

object.**EmbossShadow** [= *color*]

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

Part	Description
<i>object</i>	An object expression that evaluates to a CoolBar control.
<i>color</i>	A value or constant that determines the background or foreground colors of an object, as described in Settings.

Settings

Visual Basic uses the Microsoft Windows operating environment red-green-blue (RGB) color scheme. The settings for *color* are:

Setting	Description
Normal RGB colors	Colors specified by using the Color palette or by using the RGB or QBColor functions in code.
System default colors	Colors specified by system color constants listed in the Visual Basic (VB) object library in the Object Browser. The Windows operating environment substitutes the user's choices as specified in the Control Panel settings.

Remarks

The default setting for the **EmbossShadow** property is the system default color specified by the constant **vbButtonFace**.

The valid range for a normal RGB color is 0 to 16,777,215 (&HFFFFFF). The high byte of a number in this range equals 0; the lower 3 bytes, from least to most significant byte, determine the amount of red, green, and blue, respectively. The red, green, and blue components are each represented by a number between 0 and 255 (&HFF). If the high byte isn't 0, Visual Basic uses

the system colors, as defined in the user's Control Panel settings and by constants listed in the Visual Basic (VB) object library in the Object Browser.

Note This property has no effect on the display of an image unless the **Picture** property of the **CoolBar** control contains a valid image and the **EmbossPicture** property is set to **True**.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Enabled Property

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

Returns or sets a value that determines whether a form or control can respond to user-generated events.

Syntax

object.**Enabled** [= *boolean*]

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

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list. If <i>object</i> is omitted, the form associated with the active form module is assumed to be <i>object</i> .
<i>boolean</i>	A Boolean expression that specifies whether <i>object</i> can respond to user-generated events.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Allows <i>object</i> to respond to events.
False	Prevents <i>object</i> from responding to events.

Remarks

The **Enabled** property allows forms and controls to be enabled or disabled at [run time](#). For example, you can disable objects that don't apply to the current state of the application. You can also disable a control used purely for display purposes, such as a text box that provides read-only information.

Disabling a **Timer** control by setting **Enabled** to **False** cancels the countdown set up by the control's **Interval** property.

For a **Menu** control, **Enabled** is normally read/write at run time. But **Enabled** is read-only for menu items that are exposed or supplied by Visual Basic to add-ins, such as the Add-In Manager command on the Add-Ins menu.

Visual Basic Reference

Enabled Property Example

This example enables a **CommandButton** control whenever a **TextBox** control contains text. To try this example, paste the code into the Declarations section of a form with **CommandButton** and **TextBox** controls, and then press F5 and enter something into the text box.

```
Private Sub Form_Load ()
    Text1.Text = "" ' Clear the text box.
    Command1.Caption = "Save" ' Put caption on button.
End Sub

Private Sub Text1_Change ()
    If Text1.Text = "" Then ' See if text box is empty.
        Command1.Enabled = False ' Disable button.
    Else
        Command1.Enabled = True ' Enable button.
    End If
End Sub
```

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Enabled Property (ActiveX Controls)

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

Returns or sets a value that determines whether a form or control can respond to user-generated events.

Syntax

object.**Enabled** [= *boolean*]

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

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list. If <i>object</i> is omitted, the form associated with the active form module is assumed to be <i>object</i> .
<i>boolean</i>	A Boolean expression that specifies whether <i>object</i> can respond to user-generated events.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Allows <i>object</i> to respond to events.
False	Prevents <i>object</i> from responding to events.

Remarks

The **Enabled** property allows objects to be enabled or disabled at [run time](#). For example, you can disable objects that don't apply to the current state of the application. You can also disable a control used purely for display purposes, such as a text box that provides read-only information.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

Enabled Property (Multimedia MCI Control)

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

Determines if the control can respond to user-generated events, such as the KeyPress and mouse events.

Syntax

[form.]MMControl.**Enabled**[= {True | False}]

Remarks

This property permits the **Multimedia MCI** control to be enabled or disabled at run time. The effect of the **Enabled** property supersedes the effects of the **AutoEnable** and *ButtonEnabled* properties. For example, if the **Enabled** property is **False**, the **Multimedia MCI** control does not permit access to its buttons, regardless of the settings of the **AutoEnable** and *ButtonEnabled* properties.

The following table lists the **Enabled** property settings for the **Multimedia MCI** control.

Setting	Description
False	All buttons on the control are disabled (dimmed).
True	(Default) The control is enabled. Use the AutoEnable property to let the Multimedia MCI control automatically enable or disable the buttons in the control. Or, use the <i>ButtonEnabled</i> properties to enable or disable individual buttons in the control.

Data Type

Integer (Boolean)

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

Environment Property (Remote Data)

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

Returns a reference to a **RemoteData** control's underlying **rdoEnvironment** object.

Syntax

object.**Environment**

Set *environment* = *object*.**Environment**

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

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

Remarks

When a **RemoteData** control is initialized, **RemoteData** uses the default **rdoEnvironments(0)** the **Environment** property is initially set to this object.

If you assign another **rdoResultset** to the **RemoteData** control's **Resultset** property, the **Environment** property is set to the **rdoEnvironment** object used to create the [result set](#).

rdoEnvironment objects have properties and methods you can use to manage data. For example, you can use any method of an **rdoEnvironment** object, such as **OpenConnection**, **BeginTrans**, **CommitTrans**, or **RollbackTrans**, with the **Environment** property.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: MSComm Control

Visual Studio 6.0

EOFEnable Property

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

The **EOFEnable** property determines if the **MSComm** control looks for End Of File (EOF) characters during input. If an EOF character is found, the input will stop and the **OnComm** event will fire with the **CommEvent** property set to **comEvEOF**.

Syntax

object.**EOFEnable** [= *value*]

The **EOFEnable** 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 expression that determines whether the OnComm event is fired when an EOF character is found, as described in Settings.

Settings

The settings for *value* are:

Setting	Description
True	The OnComm event is fired when an EOF character is found.
False	(Default) The OnComm event isn't fired when an EOF character is found.

Remarks

When **EOFEnable** property is set to **False**, the control will not scan the input stream for EOF characters.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

Error Property (Multimedia MCI Control)

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

Specifies the error code returned from the last MCI command. This property is not available at design time and is read-only at run time.

Syntax

`[form.]MMControl.Error`

Remarks

If the last MCI command did not cause an error, this value is 0.

Data Type

Integer

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Error Property (WebClass Object)

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

Returns a reference to the Active Server Pages **Error** object.

Syntax

object.**Error**

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

Remarks

The **Error** property value is null during normal execution, but valid during execution of the FatalError event.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

ErrorMessage Property (Multimedia MCI Control)

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

Describes the error code stored in the **Error** property. This property is not available at design time and is read-only at run time.

Syntax

`[form.]MMControl.ErrorMessage`

Data Type

String

© 2017 Microsoft



This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

ErrorNumber Property

See Also Example [Applies To](#)

Returns a numeric value specifying an error.

Syntax

object.**ErrorNumber**

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

Return Type

Long

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: DataGrid Control

Visual Studio 6.0

ErrorText Property

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

Returns the error message string from the underlying data source. Not available at design time.

Syntax

object.**ErrorText**

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

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

Remarks

When a database error occurs as a result of user interaction with the grid, such as when the user enters text into a numeric field and then attempts to update the current record by moving to another row, the grid's Error event will fire. However, the error code passed to the event handler in the *DataError* parameter may not identify the specific error that occurred, or may even differ across operating environments. For these reasons, the **ErrorText** property is provided so that your application can parse the actual error message to determine the nature of the error.

Note The **ErrorText** property is only valid within a **DataGrid** control's Error event handler. A trappable error will occur if you attempt to access it in any other context.

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

ErrorThreshold Property (Remote Data)

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

Returns or sets a value that determines the severity level that constitutes a fatal error.

Note This property is provided for backward compatibility with RDO version 1.0 code. It should be replaced with code that implements the **rdoEngine** object's InfoMessage event which provides equivalent functionality.

Remarks

In version 4.x of Microsoft SQL Server, it is not possible to set the severity of errors using the RAISERROR statement. As a result, the ErrorThreshold property was needed to permit your code to filter those messages beyond a threshold of severity.

Version 6.x of Microsoft SQL Server now supports the inclusion of a severity level in the RAISERROR statement so it is no longer necessary to use the ErrorThreshold property.

All errors that are returned with a severity of less than 10 are trapped by the ODBC layers and set the SQL_SUCCESS_WITH_INFO result code. This causes RDO to raise the InfoMessage event but not stop query processing.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

EventParameters Property (EventInfo Object)

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

Returns a reference to the **Parameters** collection.

Syntax

object.**Parameters**

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

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Extensibility Reference

Visual Studio 6.0

Events Property

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

Supplies properties that enable add-ins to connect to all events in Visual Basic for Applications.

Syntax

object.**Events**

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

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

EventsFrozen Property

See Also Example [Applies To](#)

Returns a value indicating if the container is currently ignoring events being raised by the control. The **EventsFrozen** property is not available at the controls authoring time, and read-only at the controls run time.

Syntax

object.**EventsFrozen**

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

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

Remarks

When the **EventsFrozen** property is **True**, the container is ignoring any events that are being raised by the control. If the control wants to raise an event that cannot be lost, it must queue them up until **EventsFrozen** is **False**.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

Excluded Property

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

Returns or sets a value that determines whether a series is included on the chart.

Syntax

object.**Excluded** [= *boolean*]

The **Excluded** 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 whether a series is included on the chart, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	The chart is drawn without including the series.
False	(Default) The series is included when the chart is drawn. A series may be included in a chart, but still not display because it is Hidden .

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Exclusive Property

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

Returns or sets a value that indicates whether the underlying database for a **Data** control is opened for single-user or multi-user access.

Syntax

object.**Exclusive** [= *value*]

The **Exclusive** 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 expression that determines user access, as described in Settings.

Settings

The settings for *value* are:

Setting	Description
True	The database is open for single-user access. No one else can open the database until it's closed.
False	(Default) The database is open for multi-user access. Other users can open the database and have access to the data while it's open.

Remarks

The *value* of this property, along with the **DatabaseName**, **ReadOnly**, and **Connect** properties, is used to open a database. In the Professional and Enterprise Editions, this property corresponds to the *exclusive* argument in the **OpenDatabase** method.

The **Exclusive** property is used only when opening the **Database**. If you change the value of this property at [run time](#), you must use the **Refresh** method for the change to take effect. If someone else already has the database open, you can't open it for exclusive use and a trappable error results.

Database operations are faster if the database is opened for exclusive use.

After you open a database for exclusive use, your application can have as many instances open as necessary. However, other applications running on your system are not permitted to open the database.

The **Exclusive** property is ignored for databases accessed through ODBC.

Data Type

Boolean

© 2017 Microsoft



This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

EXENAME Property

See Also Example [Applies To](#)

Returns the root name of the executable file (without the extension) that is currently running. If running in the development environment, returns the name of the project.

Syntax

object.**EXENAME**

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

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

Expanded Property

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

Returns or sets a value that determines whether a **Node** object in a **TreeView** control is currently expanded or collapsed.

Syntax

object.Expanded[= *boolean*]

The **Expanded** 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 node is expanded or collapsed.

The settings for *boolean* are:

Setting	Description
True	The Node is currently expanded.
False	The Node is currently collapsed.

Remarks

You can use the **Expanded** property to programmatically expand a **Node** object. The following code has the same effect as double-clicking the first **Node**:

```
TreeView1.Nodes(1).Expanded = True
```

When a **Node** object is expanded, the Expand event is generated.

If a **Node** object has no child nodes, the property value is ignored.

© 2017 Microsoft

Visual Basic: Windows Controls

Expanded Property Example

This example adds several **Node** objects to a **TreeView** control. When you click the form, the **Expanded** property for each **Node** is set to **True**. To try the example, place a **TreeView** control on a form and paste the code into the form's Declarations section. Run the example, and click the form to expand all the **Node** objects.

```
Private Sub Form_Load()  
    Dim nodX As Node  
    Dim i as Integer  
    TreeView1.BorderStyle = vbFixedSingle ' Show border.  
  
    ' Create a root node.  
    Set nodX = TreeView1.Nodes.Add(,,"root","Root")  
  
    For i = 1 to 5 ' Add 5 child nodes.  
        Set nodX = TreeView1.Nodes.Add(i,tvwChild,"Node " & CStr(i))  
    Next i  
End Sub  
  
Private Sub Form_Click()  
    Dim I as Integer  
    For I = 1 to TreeView1.Nodes.Count  
        ' Expand all nodes.  
        TreeView1.Nodes(i).Expanded = True  
    Next I  
End Sub
```

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

ExpandedImage Property

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

Returns or sets the index or key value of a **ListImage** object in an associated **ImageList** control; the **ListImage** is displayed when a **Node** object is expanded.

Syntax

`object.ExpandedImage[= number]`

The **ExpandedImage** 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 numeric expression or string expression that specifies, respectively, the index value or the key value of the image to be displayed.

Remarks

This property allows you to change the image associated with a **Node** object when the user double-clicks the node or when the **Node** object's **Expanded** property is set to **True**.

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

ExportFormats Property

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

Returns a reference to the **ExportFormats** collection.

Syntax

object.**ExportFormats**

object.**ExportFormats**(*index*)

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

Part	Description
<i>object</i>	Required. An object expression that evaluates to an object in the Applies To list.
<i>index</i>	Optional. The Index or Key of a member of the collection.

Remarks

The property uses collection syntax to specify either the collection or a member of the collection to return.

The **ExportFormat** object's **FileFormatString** property determines what text will be displayed in the Save As Type box of the Export dialog box.

© 2017 Microsoft

Visual Basic Reference

Add Method, ExportFormats Collection, Template Property Example

This example creates a template, adds an **ExportFormat** object to the **ExportFormats** collection using the new template, and exports the report using the **ExportFormat** object.

```
Private Sub ExportDailyReport()  
    DataReport1.Title = "Daily Report" ' This title appears in the report.  
    Dim strTemplate As String  
    ' Create the template.  
    strTemplate = _  
    "<HTML>" & vbCrLf & _  
    "<HEAD>" & vbCrLf & _  
    "<TITLE>" & "MyCompany: " & rptTagTitle & _  
    "</TITLE>" & vbCrLf & _  
    "<BODY>" & vbCrLf & _  
    rptTagBody & vbCrLf & _  
    "<BODY>" & vbCrLf & _  
    "</HTML>"  
  
    ' Add a new ExportFormat object using the template.  
    DataReport1.ExportFormats.Add _  
    Key:="DailyReport", _  
    FormatType:=rptFmtHTML, _  
    FileFormatString:="Daily Report (*.htm)", _  
    FileFilter:="*.HTM", _  
    Template:=strTemplate  
  
    ' Export the report using the new ExportFormat object.  
    DataReport1.ExportReport _  
    FormatIndexOrKey:="DailyReport", _  
    FileName:="C:\Temp\DailyRpt", _  
    Overwrite:=True, _  
    ShowDialog:=False, _  
    Range:=rptRangeFromTo, _  
    Pagefrom:=1, _  
    Pageto:=10  
End Sub
```

© 2017 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Extender Property

See Also Example [Applies To](#)

Returns the **Extender** object for this control that holds the properties of the control that are kept track of by the container. The **Extender** property is not available at the controls authoring time, and read-only at the controls run time.

Syntax

object.**Extender**

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

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

© 2017 Microsoft