Visual Basic Reference

Visual Studio 6.0

Value Property (ActiveX Controls)

See Also Example Applies To

Returns or sets the value of an object.

Syntax

object.**Value** [= *integer*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
integer	For a **Slider** control, a long integer that specifies the current position of the slider. For the **ProgressBar** control, an integer that specifies the value of the **ProgressBar** control. For other controls, see Settings below.

Settings

For the **Button** object, the settings for *integer* are:

Constant	Value	Description
tbrUnPressed	0	(Default). The button is not currently pressed or checked.
tbrPressed	1	The button is currently pressed or checked.

Remarks

- **Slider** controlreturns or sets the current position of the slider. **Value** is always between the values for the **Max** and **Min** properties, inclusive, for a **Slider** control.
- **ProgressBar** controlreturns or sets a value indicating an operation's approximate progress toward completion. Incrementing the **Value** property doesn't change the appearance of the **ProgressBar** control by the exact value of the **Value** property. **Value** is always in the range between the values for the **Max** and **Min** properties, inclusive. Not available at design time.

Visual Basic Reference

Value Property (ActiveX Controls) Example

This example uses the **Value** property to determine which icon from an associated **ImageList** control is displayed on the **Toolbar** control. To try the example, place a **Toolbar** control on a form and paste the code into the form's Declarations section. Then run the example.

```
Private Sub Toolbar1_ButtonClick(ByVal Button As Button)
   ' Use the Key value to determine which button has been clicked.
   Select Case Button.Key

   Case "Done"   ' A check button.
      If Button.Value = vbUnchecked Then
         ' The button is unchecked.
         Button.Value = vbChecked   ' Check the button.
         ' Assuming there is a ListImage object with
         ' key "down."
         Button.Image = "down"
      Else   ' Uncheck the button
         Button.Value = vbUnchecked
         ' Assuming there is a ListImage object with
         ' key " up."
         Button.Image = "up"
      End If

   ' More Cases are possible.
   End Select
End Sub
```

Visual Basic: DataGrid Control

Visual Studio 6.0

Value Property (Column Object)

See Also Example Applies To

Sets or returns the underlying data value in a column for the current row. Not available at design time.

Syntax

object.**Value** [= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	A string expression that represents the underlying data value in a column for the current row.

Remarks

The **Value** property is useful for simulating data entry within a cell. When this property is set, the value displayed in the cell respects the setting of the column's **NumberFormat** property.

This property always returns a string variant, even if the data type of the underlying field is numeric.

Use the **Text** property to access the formatted data value in a column for the current row.

Visual Basic Reference

Visual Studio 6.0

Value Property (DEDesigner Extensibility)

See Also Example Applies To

Returns or sets the value of the **DEParameter** object. The DataEnvironment object uses this value when executing the **DECommand** object, unless a value is specified at run time.

Syntax

object.**Value** [=*value*]

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

Part	Description
object	An object expression that evaluates to an item in the Applies To list.
value	A variant expression that specifies the value of the **DEParameter** object.

Remarks

This property corresponds to the ADO Parameter Value property.

Visual Basic: Windows Controls

Visual Studio 6.0

Value Property (MonthView, DateTimePicker Controls)

See Also Example Applies To

Returns or sets the date currently displayed.

Syntax

object.**Value** [= *date*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
date	A date expression specifying a date displayed by the control.

Remarks

The **Value** property is the default property of the control.

Visual Basic: RDO Data Control

Visual Studio 6.0

Value Property (Remote Data)

See Also Example Applies To

Returns or sets the value of an object.

Syntax

object.**Value** [= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	An expression that evaluates to a value appropriate for the data type, as specified by the **Type** property of an object. (Data type is Variant.)

Remarks

Use the **Value** property to retrieve and alter data in **rdoResultset** objects. The data type of the data returned is indicated by the **Type** property of the object.

The **Value** property is the default property of the **rdoColumn** and **rdoParameter** objects. Therefore, the following lines of code are equivalent (assuming Column1 is at the first ordinal position):

```
Dim MyResultset As rdoResultset
X = MyResultset!Column1
X = MyResultset!Column1.Value
X = MyResultset(0)
X = MyResultset(0).Value
X = MyResultset("Column1").Value
X = MyResultset("Column1")
X = RemoteData1.Resultset("Column1")
X = RemoteData1.Resultset(0)
F$ = "Column1" : X = MyResultset(F$).Value
X = MyResultset(F$)
Set X = MyResultset(0): X.Value : X
```

Visual Basic: Windows Controls

Visual Studio 6.0

Value Property (UpDown Control)

See Also Example Applies To

Sets or returns the current position of the scroll value.

Syntax

object.**Value** [= *long*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
long	A long integer that specifies the current value, as described below.

Remarks

The **Value** property specifies the current value within the range of the **Min** and **Max** properties. This property is incremented or decremented when the arrow buttons are clicked. The settings of the Min and Max properties determine whether the value is incremented or decremented when the arrow buttons are clicked.

If the **SyncBuddy** property is set to **True**, the **BuddyProperty** property is synchronized when there is a change in the **Value** property or when the **BuddyProperty** property value is changed.

Visual Basic Reference

Visual Studio 6.0

Value Property

See Also Example Applies To

- **CheckBox** and **OptionButton** controls returns or sets the state of the control.
- **CommandButton** control returns or sets a value indicating whether the button is chosen; not available at design time.
- **Field** object returns or sets the content of a field; not available at design time.
- **HScrollBar** and **VScrollBar** controls (horizontal and vertical scroll bars) returns or sets the current position of the scroll bar, whose return value is always between the values for the **Max** and **Min** properties, inclusive.

Syntax

object.**Value** [= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	Value specifying the state, content, or position of a control, as described in Settings.

Settings

The settings for *value* are:

- **CheckBox** control 0 is Unchecked (default), 1 is Checked, and 2 is Grayed (dimmed).
- **CommandButton** control **True** indicates the button is chosen; **False** (default) indicates the button isn't chosen. Setting the **Value** property to **True** in code invokes the button's Click event.
- **Field** object restricted only by the Field data types.
- **HScrollBar** and **VScrollBar** controls set values between 32,768 and 32,767 to position the scroll box.
- **OptionButton** control **True** indicates the button is selected; **False** (default) indicates the button isn't selected.

Remarks

A default property of an object is assumed, and doesn't need to be specified in code. For example, **Field** is the default property of any **Recordset**, and **Value** is the default property of a **Field** object. This makes the two statements below equivalent:

```
Dn.Fields("PubID").Value = X
Dn.("PubID") = X
```

The first statement *specifies* the default properties; the second statement *assumes* them.

Visual Basic Reference

Value Property Example

This example displays an **HScrollBar** (horizontal scroll bar) control's numeric value in a **TextBox** control. To try this example, paste the code into the Declarations section of a form that has a **TextBox** control and an **HScrollBar** control. Press F5 to run the program, and then click the scroll bar.

```
Private Sub Form_Load ()
   HScroll1.Min = 0   ' Initialize scroll bar.
   HScroll1.Max = 1000
   HScroll1.LargeChange = 100
   HScroll1.SmallChange = 1
End Sub

Private Sub HScroll1_Change ()
   Text1.Text = Format (HScroll1.Value)
End Sub
```

Visual Basic: MSChart Control

ValueFormat Property

See Also Example Applies To

Returns or sets the format used to display the label as a value.

Syntax

object.**ValueFormat** [= *format*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
format	String. Describes the format used to display a label as a value.

Remarks

Use the **DataPointLabel** object's **Component** property to change the label type.

Visual Studio 6.0

Visual Basic: MSChart Control

ValueScale Property

See Also Example Applies To

Returns a reference to a **ValueScale** object that describes the scale used to display a value axis.

Syntax

object.**ValueScale**

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

Visual Basic: Windows Controls

Visual Studio 6.0

VariantHeight Property

See Also Example Applies To

Returns or sets a value indicating whether a **CoolBar** control displays all bands using the same height.

Syntax

object.**VariantHeight** [= *boolean*]

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

Part	Description
object	An object expression that evaluates to a **CoolBar** control.
boolean	A Boolean expression specifying whether all bands are a uniform height.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Band heights may vary.
False	Band heights are equal.

Remarks

When this property is set to **False**, the **CoolBar** control will display all bands based on the largest **MinHeight** property of all visible bands. When this property is set to **True**, the height of each band is based on its own **MinHeight** property if there is a single band in the row. If there is more than one band in a row, the bands will be displayed using the largest MinHeight property of all visible bands in that row.

Visual Basic Extensibility Reference

Visual Studio 6.0

VBComponents Property

See Also Example Applies To Specifics

Returns a collection of the components contained in a project.

Remarks

Use the **VBComponents** collection to access, add, or remove components in a project. A component can be a form, module, or class. The **VBComponents** collection is a standard collection that can be used in a **For Each** block.

You can use the **Parent** property to return the project the **VBComponents** collection is in.

In Visual Basic for Applications, you can use the **Import** method to add a component to a project from a file.

Visual Basic Extensibility Reference

Visual Studio 6.0

VBComponentsEvents Property

See Also Example Applies To

Returns an event object of type **VBComponentsEvents**.

Syntax

object.**VBComponentsEvents (***vbproject* **As vbProject)**

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	An object of type **vbProject** which specifies the project which contains the components.

Visual Basic Extensibility Reference

Visual Studio 6.0

VBControls Property

See Also Example Applies To

Returns a collection containing all controls on a form.

Syntax

object.**VBControls**

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

Visual Basic Extensibility Reference

Visual Studio 6.0

VBControlsEvents Property

See Also Example Applies To

Returns all events supported by the controls on a form.

Syntax

object.**VBControlsEvents**(*vbproject* **As Variant**, *vbform* **As VBForm**)

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A variant expression specifying the project which contains the controls.
vbform	The form containing the controls.

Remarks

Returns an event object of type **VBControlsEvents**. This event is sourced from a VBForm or a control on a VBForm that can contain controls.

Visual Basic Extensibility Reference

Visual Studio 6.0

VBE Property

See Also Example Applies To Specifics

Returns the root of the **VBE** object. Read-only.

Remarks

All objects have a **VBE** property that points to the root of the **VBE** object.

Visual Basic Extensibility Reference

VBE Property Example

The following example uses the **VBE** and **Name** properties to return the name of the active project.

```
Debug.Print Application.VBE.ActiveVBProject.Name
```

Visual Basic Extensibility Reference

Visual Studio 6.0

VBProjects Property

See Also Example Applies To Specifics

Returns the VBProjects collection, which represents all of the projects currently open in the Visual Basic IDE.

Syntax

object.**VBProjects**

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

Visual Basic Extensibility Reference

Visual Studio 6.0

VBProjectsEvents Property

See Also Example Applies To

Returns an event object of type **VBProjectsEvents**.

Syntax

object.**VBProjectsEvents**

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

Remarks

This is identical to using the VBProjects collection events.

Visual Basic Reference

Visual Studio 6.0

Verb Property

See Also Example Applies To

Returns or sets a value specifying an operation to perform when an object is activated using the **Action** property.

Note The **Verb** property is included for compatibility with the **Action** property in earlier versions. For current functionality, use the **DoVerb** method.

Syntax

object.**Verb** [= *number*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
number	A value that specifies the operation to perform.

Remarks

Each object can support its own set of verbs. Use the **ObjectVerbs** and **ObjectVerbsCount** properties to access the list of verbs supported by an object. Set **Verb** = 1 to specify the first verb in the list, set **Verb** = 2 to specify the second verb in the list, and so on.

Set **AutoActivate** to 2 (Double-Click) to automatically activate an object when it's double-clicked by the user.

Set **AutoVerbMenu** = **True** to display a pop-up menu containing the object's verbs when the user clicks the object with the right mouse button.

Visual Basic Reference

Visual Studio 6.0

Version Property (DEDesigner Extensibility)

See Also Example Applies To

Returns the current version number of the Data Environment designer. Read-only.

Syntax

object.**Version** [=*string*]

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

Part	Description
object	An object expression that evaluates to an item in the Applies To list.
string	The version of the current Data Environment designer.

Remarks

For this release, the **Version** property returns a value of "1.0".

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

Version Property (MSHFlexGrid)

SeeAlso Example Applies To

Returns the version of the **MSHFlexGrid** that is currently loaded in memory.

Syntax

object.**Version** [=*integer*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
integer	An integer representing the version number of the **MSHFlexGrid**.

Remarks

The version number is a three-digit integer. The first digit represents the major version number and the last two digits represent the minor version number. For example, version 3.5 returns an integer of 350.

Visual Basic: RDO Data Control

Visual Studio 6.0

Version Property (Remote Data)

See Also Example Applies To

Returns a value that indicates the version of the data source associated with the object.

Syntax

object.**Version**

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

Return Values

The **Version** property return value is a 10-character string expression.

Remarks

For an **rdoConnection** object, this property identifies the version of the data source used when the connection was created. This value is the version of ODBC to which the driver manager conforms. The version is in the form ##.##.####, where the first two digits are the major version number, the next two digits are the minor version, and the last four digits are the build number.

Visual Basic Extensibility Reference

Visual Studio 6.0

Version Property

See Also Example Applies To Specifics

Returns a String containing the version of Visual Basic for Applications that the application is using. Read-only.

Remarks

The **Version** property value is a string beginning with one or two digits, a period, and two digits; the rest of the string is undefined and may contain text or numbers.

Visual Basic Extensibility Reference

Version Property Example

The following example uses the **Version** property to return the version number of the host application.

```
Debug.Print Application.VBE.Version
```

Visual Studio 6.0

Visual Basic: MSChart Control

VertAlignment Property

See Also Example Applies To

Returns or sets the method used to vertically align text.

Syntax

object.**VertAlignment** [= *type*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
type	Integer. A VtVerticalAlignment constant used to describe the vertical alignment method of text.

Visual Basic: Windows Controls

Visual Studio 6.0

View Property (ListView Control)

See Also Example Applies To

Returns or sets the appearance of the **ListItem** objects in a **ListView** control.

Syntax

object.**View** [= *value*]

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

Part	Description
object	The object expression that evaluates to a **ListView** control.
value	An integer or constant specifying the control's appearance, as described in Settings.

Settings

The settings for *value* are:

Constant	Value	Description
lvwIcon	0	(Default) Icon. Each **ListItem** object is represented by a full-sized (standard) icon and a text label.
lvwSmallIcon	1	SmallIcon. Each **ListItem** object is represented by a small icon and a text label that appears to the right of the icon. The items appear horizontally.
lvwList	2	List. Each **ListItem** object is represented by a small icon and a text label that appears to the right of the icon. The **ListItem** objects are arranged vertically, each on its own line with information arranged in columns.
lvwReport	3	Report. Each **ListItem** object is displayed with its small icon and text labels. You can provide additional information about each **ListItem** object in a subitem. The icons, text labels, and information appear in columns with the leftmost column containing the small icon, followed by the text label. Additional columns display the text for each of the item's subitems.

Remarks

In Icon view only, use the **LabelWrap** property to specify if the **ListItem** object's labels are wrapped or not.

In Report view, you can hide the column headers by setting the **HideColumnHeaders** property to **True**. You can also use the ColumnClick event and the **Sorted**, **SortOrder**, and **SortKey** properties to sort the **ListItem** objects or subitems when a user clicks a column header. The user can change the size of the column by grabbing the right border of a column header and dragging it to the desired size.

Visual Basic: Windows Controls

Icon, SmallIcon, Icons, SmallIcons, View Properties Example

This example populates a **ListView** control with the contents of the Publishers table in the Biblio.mdb database. Four **OptionButton** controls are labeled with **View** property choices. You must place two **ImageList** controls on the form, one to contain images for the **Icon** property, and a second to contain images for the **SmallIcon** property of each **ListItem** object. To try the example, place a **ListView**, a control array of four **OptionButton** controls, and two **ImageList** controls on a form and paste the code into the form's Declarations section.

Note The example will not run unless you add a reference to the Microsoft DAO 3.51 Object Library by using the References command on the Tools menu. Run the example and click on the **ComboBox** control to switch views.

```
Private Sub Option1_Click(Index as Integer)
   ' Set the ListView control's View property to the
   ' Index of Option1
   ListView1.View = Index
End Sub

Private Sub Form_Load()
   ' Create an object variable for the ColumnHeader object.
   Dim clmX As ColumnHeader
   ' Add ColumnHeaders. The width of the columns is the width
   ' of the control divided by the number of ColumnHeader objects.
   Set clmX = ListView1.ColumnHeaders. _
   Add(, , "Company", ListView1.Width / 3)
   Set clmX = ListView1.ColumnHeaders. _
   Add(, , "Address", ListView1.Width / 3)
   Set clmX = ListView1.ColumnHeaders. _
   Add(, , "Phone", ListView1.Width / 3)

   ListView1.BorderStyle = ccFixedSingle ' Set BorderStyle property.
   ListView1.View = lvwReport ' Set View property to Report.

   ' Add one image to ImageList1--the Icons ImageList.
   Dim imgX As ListImage
   Set imgX = ImageList1.ListImages. _
   Add(, , LoadPicture("icons\mail\mail01a.ico"))
   ' Add an image to ImageList2--the SmallIcons ImageList.
   Set imgX = ImageList2.ListImages. _
   Add(, , LoadPicture("bitmaps\assorted\w.bmp"))

   ' To use ImageList controls with the ListView control, you must
   ' associate a particular ImageList control with the Icons and
   ' SmallIcons properties.
   ListView1.Icons = ImageList1
   ListView1.SmallIcons = ImageList2
   ' Label OptionButton controls with View options.
      Option1(0).Caption = "Icon"
      Option1(1).Caption = "SmallIcon"
      Option1(2).Caption = "List"
      Option1(3).Caption = "Report"
```

```
      ListView1.View = lvwIcon ' Set to Icon view

   ' Create object variables for the Data Access objects.
   Dim myDb As Database, myRs As Recordset
   ' Set the Database to the BIBLIO.MDB database.
   Set myDb = DBEngine.Workspaces(0).OpenDatabase("BIBLIO.MDB")
   ' Set the recordset to the Publishers table.
   Set myRs = myDb.OpenRecordset("Publishers", dbOpenDynaset)

   ' Create a variable to add ListItem objects.
   Dim itmX As ListItem

   ' While the record is not the last record, add a ListItem object.
   ' Use the Name field for the ListItem object's text.
   ' Use the Address field for the ListItem object's SubItem(1)
   ' Use the Phone field for the ListItem object's SubItem(2)

   While Not myRs.EOF

      Set itmX = ListView1.ListItems.Add(, , CStr(myRs!Name))
      itmX.Icon = 1   ' Set an icon from ImageList1.
      itmX.SmallIcon = 1   ' Set an icon from ImageList2.

      ' If the Address field is not Null, set SubItem 1 to the field.
      If Not IsNull(myRs!Address) Then
         itmX.SubItems(1) = CStr(myRs!Address) ' Address field.
      End If

      ' If the Phone field is not Null, set SubItem 2 to the field.
      If Not IsNull(myRs!Telephone) Then
         itmX.SubItems(2) = myRs!Telephone  ' Phone field.
      End If

      myRs.MoveNext  ' Move to next record.
   Wend
End Sub
```

Visual Studio 6.0

Visual Basic: MSChart Control

View3D Property

See Also Example Applies To

Returns a reference to a **View3D** object that describes the physical orientation of a three-dimensional chart.

Syntax

object.**View3D**

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

Visual Basic Reference

Visual Studio 6.0

ViewportHeight, ViewportLeft, ViewportTop, ViewportWidth Properties

See Also Example Applies To

Returns the current height, left, top, or width value of the Viewport.

Syntax

object.**ViewportHeight**

object.**ViewportLeft**

object.**ViewportTop**

object.**ViewportWidth**

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

Return Type

Single

Remarks

The application used to view the ActiveX document controls the size of the Viewport. However, you can use the **MinHeight** and **MinWidth** properties to resize the **UserDocument**. For example, the code below resizes a **PictureBox** control according to the ViewportHeight and ViewportWidth properties.

```
Private Sub UserDocument_Resize()
   Picture1.Width = UserDocument.ViewportWidth - _
      Picture1.Left
   Picture1.Height = UserDocument.ViewportHeight - _
      Picture1.Top
End Sub
```

Visual Basic Reference

Visual Studio 6.0

Visible Property (ActiveX Controls)

See Also Example Applies To

Returns or sets a value indicating whether an object is visible or hidden.

Syntax

object.**Visible** [= *boolean*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object is visible or hidden.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Object is visible.
False	Object is hidden.

Remarks

The **Visible** property of the **Treeview** control's **Node** object is read-only at run time.

To hide an object at startup, set the **Visible** property to **False** at design time. Setting this property in code enables you to hide and later redisplay a control at run time in response to a particular event.

Note Using the **Show** or **Hide** method on a form is the same as setting the form's **Visible** property in code to **True** or **False**, respectively.

Visual Studio 6.0

Visual Basic: MSChart Control

Visible Property (MSChart)

See Also Example Applies To

Returns or sets a value that determines whether a chart element is displayed.

Syntax

object.**Visible** [= *boolean*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression that specifies whether the item is displayed, as described in Settings.

Settings

The settings for *boolean* are:

Setting	Description
True	The chart, axis title, label, or marker are displayed.
False	The elements are hidden.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

Visible Property (Multimedia MCI Control)

See Also Example Applies To

Determines if the **Multimedia MCI** control is visible or invisible at run time.

Syntax

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

Remarks

The effect of the **Visible** property supersedes the effects of the individual *Button***Visible** properties. When the **Multimedia MCI** control is visible, the individual *Button***Visible** properties govern the visibility of the associated buttons in the control. When the **Visible** property is **False**, the entire control is invisible, and the *Button***Visible** properties are not used.

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

Setting	Description
False	The control is invisible.
True	(Default) Each button is visible or hidden individually, depending on its *Button***Visible** property. This button's function is still available in the control.

Data Type

Integer (Boolean)

Visual Basic Reference

Visual Studio 6.0

Visible Property

See Also Example Applies To

Returns or sets a value indicating whether an object is visible or hidden.

Syntax

object.**Visible** [= *boolean*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
boolean	A Boolean expression specifying whether the object is visible or hidden.

Settings

The settings for *boolean* are:

Setting	Description
True	(Default) Object is visible.
False	Object is hidden.

Remarks

To hide an object at startup, set the **Visible** property to **False** at design time. Setting this property in code enables you to hide and later redisplay a control at run time in response to a particular event.

Note Using the **Show** or **Hide** method on a form is the same as setting the form's **Visible** property in code to **True** or **False**, respectively.

Visual Basic Reference

Visible Property Example

This example creates animation using two **PictureBox** controls. To try this example, paste the code into the Declarations section of a form that contains two icon-sized **PictureBox** controls. Set the **Name** property to FileCab for both **PictureBox** controls to create an array, and then press F5 and click the picture to view the animation.

```
Private Sub Form_Load ()
   Dim I   ' Declare variable.
   FileCab(0).BorderStyle = 0   ' Set BorderStyle.
   FileCab(1).BorderStyle = 0
   ' Load icons into picture boxes.
   FileCab(1).Picture = LoadPicture("ICONS\OFFICE\FILES03B.ICO")
   FileCab(0).Picture = LoadPicture("ICONS\OFFICE\FILES03A.ICO")
   For I = 0 To 1
      FileCab(I).Move 400, 400   ' Place graphics at same spot.
   Next I
   FileCab(1).Visible = False   ' Set to invisible.
   FileCab(0).Visible = True   ' Set to visible.
End Sub

Private Sub FileCab_Click (Index As Integer)
   Dim I   ' Declare variable.
   For I = 0 To 1
      ' Switch the visibility for both graphics.
      FileCab(I).Visible = Not FileCab(I).Visible
   Next I
End Sub
```

Visual Basic: DataGrid Control

Visual Studio 6.0

VisibleCols Property

See Also Example Applies To

Returns a value indicating the number of visible columns in the **DataGrid** control. Not available at design time and read-only at run time.

Syntax

object.**VisibleCols**

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.

Remarks

The **VisibleCols** property is an integer ranging from 0 to the total number of columns available, as determined by the **Count** property of the **Columns** collection.

This property returns the number of visible columns in the current split. The value returned includes both fully and partially displayed columns.

Use the **Split** property to determine the index of the current split.

Visual Basic: DataGrid Control

VisibleCols Property Example

This example defines buttons to move the grid a whole page left or right.

```
Private Sub PageRight_Click ()
   'Page grid to the right.
   If DataGrid1.LeftCol + DataGrid1.VisibleCols < _
       DataGrid1.Columns.Count Then
      DataGrid1.LeftCol = DataGrid1.LeftCol + _
       DataGrid1.VisibleCols
   End If
End Sub

Private Sub PageLeft_Click ()
   'Page grid to the left.
   If DataGrid1.LeftCol - DataGrid1.VisibleCols >= 0 Then
      DataGrid1.LeftCol = DataGrid1.LeftCol - _
       DataGrid1.VisibleCols
   End If
End Sub
```

Visual Basic: DataCombo/DataList Controls

Visual Studio 6.0

VisibleCount Property

See Also Example Applies To

Returns a value indicating the number of visible items in the list portion of the **DataCombo** or **DataList** control.

Syntax

object.**VisibleCount**

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

Remarks

The **VisibleCount** property returns an integer from 0 to the number of items visible in the control. An item is considered visible even if only a portion of the text is visible, as when the **IntegralHeight** property setting is **False**.

Note VisibleCount may be set to 0 before the first time the list portion of the **DataCombo** control is displayed.

Data Type

Integer

Visual Basic: DataCombo/DataList Controls

VisibleCount, VisibleItems Properties Example

The following sample code uses the **VisibleCount** and **VisibleItems** properties to display the fields in all the visible records of a **DataList** control:

```
Private Sub Command1_Click()
Dim I As Integer, fld As Field, msg As Variant

For I = 0 To DataList1.VisibleCount - 1
  Data1.Recordset.Bookmark = DataList1.VisibleItems(I)
  msg = ""
  For Each fld In Data1.Recordset.Fields
    msg = msg & fld.Value & "-"
  Next
  MsgBox msg
Next I

End Sub
```

Visual Basic: Windows Controls

Visual Studio 6.0

VisibleDays Property

See Also Example Applies To

Returns an array containing the dates that are currently visible.

Syntax

object.**VisibleDays**(*index*)

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	An integer which specifies a displayed date on the calendar.

Remarks

Index can be any value from 1 to 42. A value of 1 indicates the first date that is currently displayed.

Only dates that are currently displayed can be found in the **VisibleDays** property. In addition, the number of visible days can changes depending on the settings of the **MonthColumns** and **MonthRows** properties.

As you move from month to month, the information in this property is not preserved.

Visual Basic: Windows Controls

VisibleDays Property Example

The following code prints the dates of all days displayed on the calendar. To try the example, place a **MonthView** control on a form and paste the code into the Declarations section. Run the project and double-click the form.

```
Private Sub Form_DblClick()
   Dim i As Integer
   For i = 1 To 42
      Debug.Print MonthView1.VisibleDays(i)
   Next i
End Sub
```

Visual Basic: DataCombo/DataList Controls

Visual Studio 6.0

VisibleItems Property

See Also Example Applies To

Returns an array of bookmarks, one for each visible item in the **DataCombo** or **DataList** control's list.

Syntax

object.**VisibleItems(***Index***)**

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
Index	An Integer expression that specifies the element in the array. This value can be 0 to **VisibleCount** - 1.

Remarks

These bookmarks may be used to fetch individual records from the recordset used to fill the list.

Data Type

Variant

Visual Basic: DataRepeater Control

Visual Studio 6.0

VisibleRecords Property

See Also Example Applies To

Returns or sets the bookmark of the record to be displayed at the specified row index.

Syntax

object.**VisibleRecords(***rowindex***)** [=*bookmark*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
rowindex	Required. A Long value that specifies which row to display a record in.
bookmark	Optional. Sets the bookmark of the record to display in the *rowindex* row.

Return Type

Variant

Remarks

The *rowindex* value cannot evaluate to a position beyond the EOF or BOF of the recordset.

The *rowindex* value must be between 1 (the top row) and the value of the **VisibleRows** property.

Visual Basic: DataRepeater Control

CurrentRecord, VisibleRecords Properties Example

The example causes the current record to become the first record visible. If the user scrolls the **DataRepeater** control so that the current record is hidden, invoking the procedure causes the current record to reappear.

```
Private Sub MakeFirstVisibleRecord()
   DataRepeater1.VisibleRecords(1)=DataRepeater1.CurrentRecord
End Sub
```

Visual Basic: DataGrid Control

Visual Studio 6.0

VisibleRows Property

See Also Example Applies To

Returns a value indicating the number of visible rows in the **DataGrid** control. This property is read-only at run time.

Syntax

object.**VisibleRows**

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

Remarks

The **VisibleRows** property returns an integer ranging from 0 to the number of rows in the **DataGrid** control. The **VisibleRows** property includes either fully or partially visible **DataGrid** control rows.

Visual Basic: DataGrid Control

VisibleRows Property Example

This example selects all the rows that are currently visible on the grid.

```
Private Sub SelectVisible_Click ()
   Dim I
   For I = 0 To DataGrid1.VisibleRows - 1
      DataGrid1.SelBookmarks.Add DataGrid1.RowBookmark(I)
   Next I
End Sub
```

Visual Basic: DataRepeater Control

Visual Studio 6.0

VisibleRows Property (DataRepeater Control)

See Also Example Applies To

Returns the number of fully visible rows in the control.

Syntax

object.**VisibleRows**

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

Return Type

Long

Remarks

Use the **VisibleRows** property together with the **VisibleRecords** property to specify where the current record displays. For example, to display the current record in the bottommost row, use the code below:

```
DataRepeater1.VisibleRecords(DataRepeater1.VisibleRows) = _
DataRepeater1.CurrentRecord
```

Visual Basic for Applications Reference

Visual Studio 6.0

VolumeName Property

See Also Example Applies To Specifics

Description

Sets or returns the volume name of the specified drive. Read/write.

Syntax

object.**VolumeName** [= *newname*]

The VolumeName property has these parts:

Part	Description
object	Required. Always the name of a **Drive** object.
newname	Optional. If provided, *newname* is the new name of the specified *object*.

Remarks

The following code illustrates the use of the **VolumeName** property:

```
Sub ShowVolumeInfo(drvpath)
    Dim fs, d, s
    Set fs = CreateObject("Scripting.FileSystemObject")
    Set d = fs.GetDrive(fs.GetDriveName(fs.GetAbsolutePathName(drvpath)))
    s = "Drive " & d.DriveLetter & ": - " & d.VolumeName
    MsgBox s
End Sub
```

Visual Basic: MSChart Control

VtColor Property

See Also Example Applies To

Returns a reference to a **VtColor** object that describes a drawing color in a chart.

Syntax

object.**VtColor**

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

Visual Studio 6.0

Visual Basic: MSChart Control

VtFont Property

See Also Example Applies To

Returns a reference to a **VtFont** object that describes the font used to display chart text.

Syntax

object.**VtFont**

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