This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

Ghosted Property

See Also Example Applies To

Returns or sets a value that determines whether a **ListItem** object in a **ListView** control is unavailable (it appears dimmed).

Syntax

object.**Ghosted** [= *boolean*]

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

Part	**Description**
object	An object expression that evaluates to a **ListItem** object.
boolean	A Boolean expression specifying if the icon or small icon is ghosted, as described in Settings.

Settings

The settings for *boolean* are:

Setting	**Description**
True	The **ListItem** object is unavailable to the user.
False	(Default) The **ListItem** is available.

Remarks

The **Ghosted** property is typically used to show when a **ListItem** is cut, or disabled for some reason.

When a ghosted **ListItem** is selected, the label is highlighted but its image is not.

Visual Basic: Windows Controls

Ghosted, MultiSelect Properties Example

This example populates a **ListView** control with the contents of the Authors table from the Biblio.mdb database, and lets you use **OptionButton** controls to set **MultiSelect** property options. You can select any item, or hold down the SHIFT Key and select multiple items. Clicking on the **CommandButton** sets the **Ghosted** property of the selected items to **True**. To try the example, place a control array of two **OptionButton** controls, a **ListView** control, an **ImageList** control, and a **CommandButton** control 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, select a **MultiSelect** option by clicking an **OptionButton**, click on items to select them and click the **CommandButton** to ghost them.

```
Private Sub Command1_Click()
   Dim x As Object
   Dim i As Integer
   ' Ghost selected ListItem.
If ListView1.SelectedItem Is Nothing Then Exit Sub
   For i = 1 To ListView1.ListItems.Count
      If ListView1.ListItems(i).Selected = True Then
         ListView1.ListItems(i).Ghosted = True
      End If
   Next i
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)

   ' Label OptionButton controls with MultiSelect options.
      Option1(0).Caption = "No MultiSelect"
      Option1(1).Caption = "MultiSelect"
      ListView1.MultiSelect = 1 ' Set MultiSelect to True

   ListView1.BorderStyle = ccFixedSingle ' Set BorderStyle property.
   ListView1.View = lvwReport   ' Set View property to Report.
   ' Add one image to ImageList control.
   Dim imgX As ListImage
   Set imgX = ImageList1.ListImages. _
   Add(, , LoadPicture("icons\mail\mail01a.ico"))
   ListView1.Icons = ImageList1

   ' 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 icon to the ImageList icon.

      ' 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

   ListView1.View = lvwIcon  ' Show Icons view.
   Command1.Caption = "Cut"  ' Set caption of the CommandButton.
   ' Add a caption to the form.
   Me.Caption = "Select any item(s) and click 'Cut'."
End Sub

Private Sub Option1_Click(Index as Integer)
   ListView1.MultiSelect = Index
End Sub
```

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

GrandTotalName Property

See Also Example Applies To

Specifies the name used for a grand total Recordset when the **DECommand** object has an associated grand total **DEAggregate** object defined.

Syntax

object.**GrandTotalName** [=*string*]

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

Part	Description
object	An object expression that evaluates to an item in the Applies To list.
string	A string expression that specifies the name for the Grand Total Recordset.

> This documentation is archived and is not being maintained.

Visual Basic: PictureClip Control

Visual Studio 6.0

GraphicCell Property (PictureClip Control)

See Also Example Applies To

A one-dimensional array of pictures representing all of the picture cells. This property is not available at design time and is read-only at run time.

Syntax

[*form*.]*PictureClip*.**GraphicCell** (*Index%*)

Remarks

- Use the **Rows** and **Cols** properties to divide a picture into a uniform matrix of graphic cells.
- The cells specified by **GraphicCell** are indexed, beginning with 0, and increase from left to right and top to bottom.
- When reading this property, an error is generated when there is no picture or when the **Rows** or **Cols** property is set to 0.

Data Type

Integer

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

Green Property

See Also Example Applies To

Returns or sets the green component of the RGB value in a chart.

Syntax

object.**Green [=***g***]**

The **Green p**roperty syntax has these parts:

Part	**Description**
object	An object expression that evaluates to an object in the Applies To list.
g	Integer. The green value.

Remarks

RGB specifies the relative intensity of red, green, and blue to cause a specific color to be displayed. The valid range for a normal RGB color is 0 to 16,777,215. The value for any argument to RGB that exceeds 255 is assumed to be 255.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

GridColor, GridColorBand, GridColorFixed, GridColorHeader, GridColorIndent, GridColorUnpopulated Properties

See Also Example Applies To

Returns or sets the line color used between the cells, bands, headers, indents, or unpopulated areas of the **MSHFlexGrid**.

Note If using the **MSFlexGrid**, only the **GridColor** and **GridColorFixed** properties are available.

Syntax

object.**GridColor** [= *color*]
object.**GridColorBand(***BandNumber***)** [= *color*]
object.**GridColorFixed** [= *color*]
object.**GridColorHeader(***BandNumber***)** [= *color*]
object.**GridColorIndent(***BandNumber***)** [= *color*]
object.**GridColorUnpopulated** [= *color*]

Syntax for the **GridColor**, **GridColorBand**, **GridColorFixed**, **GridColorHeader**, **GridColorIndent**, and **GridColorUnpopulated** properties has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
BandNumber	Required. A Long value that specifies the band being affected.
color	A value or constant that determines the color used to paint the gridlines in the scrollable or fixed areas of the **MSFlexGrid**.

Remarks

The **GridColor** property can only be used when the **GridLines** property is set to 1 (Lines).

The **GridColorFixed** property can only be used when **GridLinesFixed** is set to 1 (Lines).

Raised and inset gridlines are always drawn in black and white.

Visual Basic: Windows Controls

Visual Studio 6.0

GridLines Property (ListView Control)

See Also Example Applies To

Returns or sets a value the determines if the ListView control, in Report view, displays gridlines.

Syntax

object.**GridLines** [= *boolean*]

The **GridLines** 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 the appearance of gridlines, as shown in Settings.

Settings

The settings for *boolean* are:

Constant	Description
False	(Default) No gridlines are displayed.
True	The control displays gridlines.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

GridLines, GridLinesBand, GridLinesFixed, GridLinesHeader, GridLinesIndent, GridLinesUnpopulated Properties (MSHFlexGrid)

See Also Example Applies To

Returns or sets a value that determines whether lines are drawn between cells, bands, headers, indents, or unpopulated areas. These properties also determine the type of lines that are drawn in the **MSHFlexGrid**.

Note If using the **MSFlexGrid**, only the **GridLines** and **GridLinesFixed** properties are available.

Syntax

object.**GridLines** [= *value*]
object.**GridLinesBand(***BandNumber***)** [= *value*]
object.**GridLinesFixed** [= *value*]
object.**GridLinesHeader(***BandNumber***)** [= *value*]
object.**GridLinesIndent(***BandNumber***)** [= *value*]
object.**GridLinesUnpopulated** [= *value*]

Syntax for the **GridLines**, **GridLinesBand**, **GridLinesFixed**, **GridLinesHeader**, **GridLinesIndent**, and **GridLinesUnpopulated** properties has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
BandNumber	Required. A Long value that specifies the band being affected.
value	An integer or constant that specifies the type of lines drawn, as described in Settings.

Settings

The settings for *value* are:

Constant	Value	Description
flexGridNone	0	There are no lines between cells. In the **MSFlexGrid**, this is the default for **GridLines**.
flexGridFlat	1	The line style between cells is set to normal, flat, lines. In the **MSHFlexGrid**, this is the default for **GridLines**.
flexGridInset	2	The line style between cells is set to inset lines. In the **MSFlexGrid**, this is the default for **GridLinesFixed**.
flexGridRaised	3	The line style between cells is set to raised lines. In the **MSHFlexGrid**, this is the default for **GridLinesFixed**.
flexGridDashes	4	The line style between cells is set to dashed lines.
flexGridDots	5	The line style between cells is set to dotted lines.

Remarks

When the **GridLines** property is set to 1 (Lines), the color of the lines is determined by the **GridColor** property.

Raised and inset gridlines are always black and white.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

GridLineWidth Property

See Also Example Applies To

Returns or sets the width in pixels of the gridlines for an **MSFlexGrid** control.

Syntax

object.**GridLineWidth** [= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	An integer specifying the gridline width. The minimum setting is 1 (default); the maximum setting is 10.

This documentation is archived and is not being maintained.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

GridLineWidth, GridLineWidthBand, GridLineWidthFixed, GridLineWidthHeader, GridLineWidthIndent, GridLineWidthUnpopulated Properties (MSHFlexGrid)

SeeAlso Example Applies To

Returns or sets the width, in pixels, of the lines displayed between cells, bands, headers, indents, or unpopulated areas.

Note For the **MSFlexGrid** control, only the **GridLineWidth** property is available.

Syntax

object.**GridLineWidth** [= *value*]
object.**GridLineWidthBand(***BandNumber***)** [= *value*]
object.**GridLineWidthFixed** [= *value*]
object.**GridLineWidthHeader(***BandNumber***)** [= *value*]
object.**GridLineWidthIndent(***BandNumber***)** [= *value*]
object.**GridLineWidthUnpopulated** [= *value*]

Syntax for the **GridLineWidth**, **GridLineWidthBand**, **GridLineWidthFixed**, **GridLineWidthHeader**, **GridLineWidthIndent**, and **GridLineWidthUnpopulated** properties has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
BandNumber	Required. A Long value that specifies the band being affected.
value	A numeric expression that specifies the preferred width, in pixels, for the current line.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

GridX, GridY Properties

See Also Example Applies To

Specifies the divisions of the alignment grid at design time (ignored at run time).

Syntax

object.**GridX**

object.**GridY**

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

Remarks

The **GridX** and **GridY** properties are used only at design time as an aid in aligning controls on the **DataReport** object.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

GroupCommand Property

See Also Example Applies To

Specifies whether a **DECommand** object is grouped.

Syntax

object.**GroupCommand** [=*Boolean*]

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

Part	Description
object	An object expression that evaluates to an item in the Applies To list.
Boolean	A Boolean expression that defines whether the DECommand object is grouped. The default is **False**, or ungrouped.

Remarks

This property is used when you define a group-based hierarchy. If set to **True**, the **DECommand** object is grouped. Use the **DEGroupingFields** collection to define the Field objects by which the **DECommand** object is grouped.

This documentation is archived and is not being maintained.

Visual Basic Extensibility Reference

Visual Studio 6.0

GUID Property

See Also Example Applies To Specifics

Returns a String containing the class identifier of an object. Read-only.

Visual Basic Extensibility Reference

GUID Property Example

The following example uses the **GUID** property to return the globally unique ID number for the specified **Reference** object in the specified project.

```
Debug.Print Application.VBE.VBProjects(1).References(1).GUID
```

Visual Studio 6.0

Visual Basic: MSChart Control

GuidelinePen Property

See Also Example Applies To

Returns a reference to a **Pen** object that describes the pattern of line and color used to display guidelines.

Syntax

object.**GuidelinePen**

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

Remarks

Setting this property automatically sets the **ShowGuideLines** property to **True**.

Visual Basic: MSChart Control

Pen Object, GuideLinePen Property Example

The following example sets the pen attributes for a two-dimensional xy chart series. The **GuideLinePen** property returns a reference to a **Pen** object.

```
Private Sub Command1_Click()
   ' Set Guide Lines for 2D XY chart Series 1.
   MSChart1.ChartType = VtChChartType2dXY
   MSChart1.Plot.SeriesCollection.Item(1) _
   .ShowGuideLine(VtChAxisIdX) = True
   With _
   MSChart1.Plot.SeriesCollection.Item(1).GuideLinePen
      ' Set Pen attributes.
      .VtColor.Set 255, 255, 0
      .Width = 10
      .Style = VtPenStyleDashDot
      .Join = VtPenJoinRound
      .Cap = VtPenCapRound
   End With
End Sub
```