

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

ReadProperties Event

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

Occurs when loading an old instance of an object that has a saved state.

Syntax

Sub *object*_ReadProperties(*pb* As PropertyBag)

The **ReadProperties** event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>pb</i>	An object of the type PropertyBag class that contains the saved data to load.

Remarks

When this event occurs, the object author can load in the saved state from *pb*, by calling the **ReadProperty** method of the **PropertyBag** object for each value that is to be loaded. This event occurs after the Initialize event.

Always include error trapping when handling the ReadProperties event, to protect the control from invalid property values that may have been entered by users editing the file containing the saved data with text editors. However, you should not raise an error in an event, since doing so may be fatal to the container, so any error trapping in the ReadProperties event procedure should not include raising errors.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: DataRepeater Control

Visual Studio 6.0

RepeatedControlLoaded, RepeatedControlUnloaded Events

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

RepeatedControlLoaded Occurs after the **RepeatedControl** has been created and initialized.

RepeatedControlUnLoaded Occurs when the **RepeatedControl** is about to be unloaded.

Syntax

Private Sub *object*_RepeatedControlLoaded()

Private Sub *object*_RepeatedControlUnloaded()

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

Remarks

At run time you can change the **RepeatedControl** by setting the **RepeatedControlName** property to a new programmatic ID. When doing so, use the RepeatedControlLoaded event to initialize values on the user control.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Reposition Event

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

Occurs after a record becomes the [current record](#).

Syntax

Private Sub *object*.**Reposition** (*[index As Integer]*)

The Reposition event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	Identifies the control if it's in a control array.

Remarks

When a **Data** control is loaded, the first record in the **Recordset** object becomes the current record, causing the Reposition event. Whenever a user clicks any button on the **Data** control, moving from record to record, or if you use one of the Move methods, such as **MoveNext**, the Find methods, such as **FindFirst**, or any other property or method that changes the current record, the Reposition event occurs after each record becomes current.

In contrast, the **Validate** event occurs before moving to a different record.

You can use this event to perform calculations based on data in the current record or to change the form in response to data in the current record.

© 2018 Microsoft

Visual Basic Reference

Reposition Event Example

This example uses the **Reposition** event to update a list of **Titles** from the selected **Publisher** in the **Biblio.mdb** sample database. First place a **DataGrid**, **TextBox**, and two **Data** controls on a form. Set the **DatabaseName** properties of both **Data** controls to the **Biblio.mdb** sample database. Set the **RecordSource** property of **Data1** to **Publishers**. Set the **DataSource** property of **Text1** to **Data1** and the **DataField** property to **Name**. Set the **DataSource** property of **DataGrid1** to **Data2**. Add the following code:

```
Private Sub Data1_Reposition()  
    ' Select all Titles that are published by  
    ' the current record in Data1  
    Data2.RecordSource = "Select * from Titles where PubID = " & Data1.Recordset("PubID")  
    Data2.Refresh ' Rebuild the recordset  
End Sub
```

The list of **Titles** in the **DataGrid** control will automatically be updated as you move through the **Publishers** recordset with **Data1**.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

Reposition Event (Remote Data)

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

Occurs after a row becomes the [current row](#).

Syntax

Private Sub *object*.**Reposition** (*[index As Integer]*)

The Reposition event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	Identifies the control if it's in a control array.

Remarks

When a [RemoteData control](#) is loaded, the first row in the **rdoResultset** object becomes the current row, causing the Reposition event to fire. The Reposition event fires after each row becomes current:

Whenever a user clicks any button on the **RemoteData control to move** from row to row.

- You use one of the *Move* methods, such as **MoveNext**.
- You use any other property or method that changes the current row

In contrast, the Validate event occurs before moving to a different row. The RowCurrencyChange event associated with the **rdoResultset** also fires when the current result set row changes.

You can use the Reposition event to perform calculations based on data in the current row or to change the form in response to data in the current row.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Extensibility Reference

Visual Studio 6.0

RequestChangeFileName Event

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

Occurs after specifying a new filename for a component or project, and the name change is completed.

Syntax

Sub RequestChangeFileName(*vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *newname* **As String**, *oldname* **As String**, *cancel* **As Boolean**)

The RequestChangeFileName event syntax has these parts:

Part	Description
<i>vbproject</i>	A VBProject object specifying the name of the project the new file will be added to.
<i>filetype</i>	An enumerated value (vbext_FileType) specifying the type of file that was written, as listed in Settings.
<i>newname</i>	A string expression specifying the name given to the renamed file.
<i>oldname</i>	A string expression specifying containing the name of the file before it was renamed.
<i>cancel</i>	A Boolean expression that determines the default Visual Basic action, as described in Settings.

Settings

The settings for *filetype* are:

Constant	Value	Description
vbext_ft_Form	0	File type is a form.
vbext_ft_Module	1	File type is a basic module.
vbext_ft_Class	2	File type is a class module.
vbext_ft_Project	3	File type is a project.
vbext_ft_Exe	4	File type is an executable file.

vbext_ft_Res	6	File type is a resource file.
vbext_ft_UserControl	7	File type is a User control.
vbext_ft_PropertyPage	8	File type is a Property Page .
vbext_ft_DocObject	9	File type is a User Document .
vbext_ft_Binary	10	File type is a binary file.
vbext_ft_GroupProject	11	File type is a group project.
vbext_ft_Designer	12	File type is a designer object.

The settings for *cancel* are:

Setting	Description
True	Cancel the renaming of the file. This event won't be triggered for any subsequent add-ins connected to the FileControl object.
False	Continue triggering this event for subsequent add-ins connected to the FileControl object.

Remarks

This event allows all the add-ins to examine the new filename that is proposed to be added to the project, and decide whether to accept or cancel the name change.

This event occurs in all add-ins that are connected to the **FileControl** object. The add-in cannot prevent the file from being written to disk because the operation is complete. However, you can use this event to perform other tasks, such as:

- Log information about the event.
- Update information about the file.
- Back up the file.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Extensibility Reference

Visual Studio 6.0

RequestWriteFile Event

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

Occurs prior to saving any project component with unsaved changes.

Syntax

Sub RequestWriteFile(*vbproject* **As VBProject**, *filename* **As String**, *cancel* **As Boolean**)

The RequestWriteFile event syntax has these parts:

Part	Description
<i>vbproject</i>	A VBProject object specifying the name of the project containing the component.
<i>filename</i>	A string expression containing the name of the file to be saved.
<i>cancel</i>	A Boolean expression used as a flag to cancel the action, as described in Settings.

Settings

The settings for *cancel* are:

Setting	Description
True	Does not write the file to disk. This event is not triggered for any subsequent add-ins connected to the FileControl object.
False	Continues triggering this event for subsequent add-ins connected to the FileControl object.

Remarks

The RequestWriteFile event occurs once for each saved component, and once for each associated binary data file (such as .Frx or .Pgx files).

This event allows add-ins to prepare the specified file for writing. For example, you could use it to enable an add-in to check out a file from a source code control project prior to writing to it.

This event occurs in all add-ins that are connected to the **FileControl** object. The add-in cannot prevent the file from being written to disk because the operation is complete. However, you can use this event to perform other tasks, such as:

- Log information about the event.
- Update information about the file.
- Back up the file.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Resize Event

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

Occurs when an object is first displayed or when the window state of an object changes. (For example, a form is maximized, minimized, or restored.)

Syntax

```
Private Sub Form_Resize( )
```

```
Private Sub object_Resize(height As Single, width As Single)
```

The Resize event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>height</i>	Number specifying the new height of the control.
<i>width</i>	Number specifying the new width of the control.

Remarks

Use a Resize event procedure to move or resize controls when the parent form is resized. You can also use this event procedure to recalculate variables or properties, such as **ScaleHeight** and **ScaleWidth**, that may depend on the size of the form. If you want graphics to maintain sizes proportional to the form when it's resized, invoke the Paint event by using the **Refresh** method in a Resize event procedure.

Whenever the **AutoRedraw** property is set to **False** and the form is resized, Visual Basic also calls the related events, Resize and Paint, in that order. When you attach procedures for these related events, be sure their actions don't conflict.

When an **OLE** container control's **SizeMode** property is set to 2 (Autosize), the control is automatically sized according to the display size of the object contained in the control. If the display size of the object changes, the control is automatically resized to fit the object. When this occurs, the Resize event is invoked for the object before the **OLE** container control is resized. The *height* and *width* parts indicate the optimal size for displaying the object (this size is determined by the application that created the object). You can size the control differently by changing the values of the *height* and *width* parts in the Resize event.

© 2018 Microsoft

Visual Basic Reference

Resize Event Example

This example automatically resizes a **TextBox** control to fill the form whenever the form is resized. To try this example, paste the code into the Declarations section of a form that contains a **TextBox**. Set the **TextBox** control's **MultiLine** property to **True**, its **ScrollBars** property to 3, and its **BorderStyle** property to 0, and then press F5 and resize the form.

```
Private Sub Form_Load ()  
    Text1.Text = "" ' Clear text.  
End Sub  
  
Private Sub Form_Resize ()  
    Text1.Move 0,0, ScaleWidth, ScaleHeight  
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Respond Event

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

Occurs when a **WebItem** object is activated by a user request, if no event is found that directly corresponds to the selected element.

Syntax

Private Sub *object*.Respond()

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

Remark

Requests from the browser often correspond to a template event for the **WebItem**. For example, when the user selects Hyperlink1, the webclass fires a Hyperlink1 event. However, if no direct correlation is found between the selected element and an event for the **WebItem**, the system fires the Respond event. The Respond event is also fired when you navigate from one **WebItem** to another using the **NextWebItem** method.

The Respond event is the default event for a **WebItem** object. When this event is fired, the code must create the response to be sent to the browser using the **Response** object.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

ResultsChanged Event

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

Occurs when a new result set is made available after the **MoreResults** method is executed.

Private Sub *object*.ResultsChanged()

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

Remarks

This event is raised after the **MoreResults** method completes and a new set of rows is loaded into the result set. This event is fired even if there are no more sets and the **MoreResults** method returns **False**. In this case, both the EOF and BOF properties will be **True**, indicating that the result set is empty.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

RetainedProject Property

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

Returns a value that determines whether a project will remain loaded in memory.

Syntax

object.**RetainedProject** [= *boolean*]

The **RetainedProject** 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 a project will remain loaded in memory.

Settings

The settings for *boolean* are:

Setting	Description
True	A project remains loaded in memory.
False	(Default) A project does not remain loaded in memory.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

RollbackTrans Event

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

Occurs after the **RollbackTrans** method has completed.

Private Sub *object*.RollbackTrans()

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

Remarks

You can respond to this event to synchronize some other process with the transaction.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

RowColChange Event (MSHFlexGrid)

SeeAlso Example [Applies To](#)

Occurs when the currently active cell changes to a different cell.

Syntax

Private Sub *object_RowColChange()*

The RowColChange event syntax has one part:

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

Remarks

The following order of events occurs when the active cell changes: first LeaveCell, next EnterCell, finally RowColChange. The RowColChange event occurs when a user clicks a new cell. It does not occur while a user drags a selection across the **MSHFlexGrid**.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

RowColChange Event

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

Occurs when the current cell changes to a different cell.

Syntax

```
Private Sub object_RowColChange ([index As Integer, Lastrow As String, Lastcol As Integer])
```

The RowColChange event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that uniquely identifies a control if it is in a control array.
<i>lastrow</i>	(For DataGrid control) A string expression specifying the previous row position.
<i>lastcol</i>	(For DataGrid control) An integer specifying the previous column position.

Remarks

This event occurs whenever the user clicks a cell other than the current cell or when you programmatically change the current cell within a selection using the **Col** and **Row** properties.

The SelChange event also occurs when a user clicks a new cell, but doesn't occur when you programmatically change the selected range without changing the current cell.

For the **DataGrid** control, the position of the current cell is provided by the **Bookmark** and **ColIndex** properties. The previous cell position is specified by *lastrow* and *lastcol*. If you edit data and then move the current cell position to a new row, the update events for the original row are completed before another cell becomes the current cell.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

RowCurrencyChange Event

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

Occurs after the result set has repositioned to a new row, BOF or EOF.

Private Sub *object*.RowCurrencyChange()

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

Remarks

This event is raised after the result set has repositioned to a new row, or has moved to either BOF or EOF. Any of the *Move* methods, the **AbsolutePosition**, **PercentPosition**, or **Bookmark** properties, or the **Requery**, **MoreResults**, or **Update** (after an **AddNew**) methods can also cause a the current row pointer to be repositioned and cause the RowCurrencyChange event to fire. The current position can be determined by accessing the **AbsolutePosition**, **PercentPosition**, or **Bookmark** properties of the object.

The RowCurrencyChange event can be used to execute a detail query when an associated master row currency changes. For example, if you setup a form containing a master customer record, and a set of rows corresponding to customer orders, you can use the RowCurrencyChange event to launch a query that returns all associated order information each time the user chooses another master customer record.

Note The order in which the RowCurrencyChange and Reposition events fire cannot be predicted.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: DataGrid Control

Visual Studio 6.0

RowResize Event

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

Occurs when a user resizes a row in a **DataGrid** control.

Syntax

Private Sub *object_RowResize* ([*index As Integer*,] *cancel As Integer*)

The RowResize event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that identifies a control if it is in a control array.
<i>cancel</i>	A Boolean expression that determines if a change is canceled, as described in Settings.

Settings

The settings for *cancel* are:

Setting	Description
True	Cancels the change and restores row to its original height.
False	(Default) Continues with change of row height.

Remarks

The user can resize the **DataGrid** control rows using the mouse. When the user changes the height, the RowResize event is triggered. Your event procedure can accept the change, alter the degree of change, or cancel the change completely.

The **DataGrid** control's **RowHeight** property determines the height for all rows in the control.

If you set the *cancel* argument to **True**, the row height is restored. To alter the degree of change, set the **RowHeight** property to the desired value.

Executing the **Refresh** method within the procedure causes the control to be repainted even if the *cancel* argument is **True**.

© 2018 Microsoft

Visual Basic: DataGrid Control

RowResize Event Example

This example ensures that there are at least five visible rows in the grid.

```
Private Sub DataGrid1_RowResize (Cancel As Integer)
    If DataGrid1.VisibleRows < 5 Then Cancel = True
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RDO Data Control

Visual Studio 6.0

RowStatusChanged Event

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

Occurs after the data state of the current row changes due to an edit, delete or insert.

Private Sub *object*.RowStatusChange()

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

Remarks

This event is raised after the status of the current row data changes. The status of a row can change due to an **Delete**, or **Update** operation. The current status for the row can be determined using the **Status** property of the object.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

Scroll Event (MSHFlexGrid)

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

Occurs when the contents of the **MSHFlexGrid** are scrolled. This can be done using the scroll bars, keyboard, or programmatically by changing the **TopRow** or **LeftCol** properties.

Syntax

Private Sub *object_Scroll()*

The Scroll event syntax has one parts:

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

Remarks

If the **ScrollTrack** property is **True**, this event occurs while the user is dragging or scrolling the mouse. If the **ScrollTrack** property is **False**, this event occurs once the drag is complete.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

Scroll Event (Slider Control)

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

Occurs when you move the slider on a **Slider** control, either by clicking on the control or using keyboard commands.

Syntax

Private Sub *object_Scroll()*

The object placeholder represents an object expression that evaluates to a **Slider** control.

Remarks

The Scroll Event occurs before the Click event.

The Scroll Event continuously returns the value of the **Value** property as the slider is moved. You can use this event to perform calculations to manipulate controls that must be coordinated with ongoing changes in the **Slider** control. In contrast, use the Change event when you want an update to occur only once, after a **Slider** control's **Value** property has changed.

Note Avoid using a **MsgBox** statement or function in this event.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Scroll Event

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

Occurs when the scroll box on a **ScrollBar** control, or an object which contains a scrollbar, is repositioned or scrolled horizontally or vertically.

Syntax

```
Private Sub DataGrid_Scroll([cancel As Integer])
```

```
Private Sub object_Scroll( )
```

The Scroll event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>cancel</i>	Determines whether the scroll operation succeeds and the ScrollBar or DataGrid are repainted, as described in Remarks.

Remarks

For a **DataGrid** control, this event occurs when the user scrolls the grid horizontally or vertically but before the grid is repainted to display the results of the scroll operation.

For a **ComboBox** control, this event occurs only when the scrollbars in the dropdown portion of the control are manipulated.

Setting *cancel* to **True** causes the **DataGrid** scroll operation to fail, and no repaint operation occurs. If the **Refresh** method is invoked within this event, the grid is repainted in its new (scrolled) arrangement even if *cancel* is set to **True**. However, in this case, the grid is repainted again because the scroll operation fails and it snaps back to its previous position.

You can use this event to perform calculations or to manipulate controls that must be coordinated with ongoing changes in scroll bars. In contrast, use the Change event when you want an update to occur only once, after a **ScrollBar** control changes.

Note Avoid using a **MsgBox** statement or function in this event.

© 2018 Microsoft

Visual Basic Reference

Scroll Event Example

This example changes the size of a **Shape** control to correspond to the value of a horizontal scroll bar (**HScrollBar**) as you drag the scroll box on the scroll bar. To try this example, paste the code into the Declarations section of a form that contains a **Shape** control, a **Label** control, and an **HScrollBar** control. Set the **Index** property for the **Shape** control to 0 to create a control array. Then press F5 and move the scroll bar.

```
Private Sub Form_Load ()
    ' Move and size the first Shape control.
    Shape1(0).Move HScroll11.Left, HScroll11.Top * 1.5, HScroll11.Width, HScroll11.Height
    Label1.Caption = "" ' Set the Label caption.
    Load Shape1(1) ' Create the second Shape.
    ' Move and size the second Shape control.
    Shape1(1).Move Shape1(0).Left, Shape1(0).Top, 1, Shape1(0).Height
    Shape1(1).BackStyle = 1 ' Set BackStyle to Opaque.
    Shape1(1).Visible = True ' Display the second Shape.
    HScroll11.Min = 1 ' Set values of the scroll bar.
    HScroll11.Max = HScroll11.Width
End Sub

Private Sub HScroll11_Change ()
    Label1.Caption = "Changed" ' Display message after change.
End Sub

Private Sub HScroll11_Scroll ()
    Shape1(1).BackColor = &HFF0000 ' Set Shape color to Blue.
    Label1.Caption = "Changing" ' Display message while scrolling.
    Shape1(1).Width = HScroll11.Value ' Size Shape to Scroll Value.
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

SelectionChanged Event

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

Occurs when the selection of controls on the form has changed.

Syntax

Sub *object* **SelectionChanged()**

The SelectionChanged event syntax has these parts:

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

Remarks

The firing of this event notifies the property page that the selection of controls has changed, and therefore the display of current property values may need to be updated. The **SelectedControls** property should be read to find out the new set of selected controls.

The SelectionChanged event is also raised when the property page is first brought up for a control.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: DataGrid Control

Visual Studio 6.0

SelChange Event (DataGrid Control)

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

Occurs when the user selects a different range of rows or columns.

Syntax

Private Sub *object_SelChange*([*index As Integer*,] *cancel As Integer*)

The SelChange event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that identifies a control if it is in a control array.
<i>cancel</i>	A Boolean expression that determines if a change is canceled, as described in Settings.

Settings

The settings for *cancel* are:

Setting	Description
True	If your event procedure sets the Cancel argument to True , the previous row and column selections (if any) are restored, and the SelStartCol , SelEndCol , and SelBookmarks properties revert to their previous values
False	(Default) Continues with change.

Remarks

This event is triggered under several circumstances:

- When the user selects a single row by clicking its record selector.
- When the user adds a row to the list of selected rows by clicking its record selector while holding down the CTRL key.

- When the user selects a single column by clicking its header.
- When the user changes the range of selected columns by dragging to an adjacent column within the header row.
- When the user extends the range of selected columns by holding down the SHIFT key and clicking on an unselected column header.
- When the user clears the current row or column selection by clicking an individual cell, this event will fire before the RowColChange event.

The current range of selected columns is provided by the **SelStartCol** and **SelEndCol** properties. The bookmarks of the selected rows are available in the collection provided by the **SelBookmarks** property. Within this event procedure, these properties reflect the user's pending selection(s).

This event is only triggered by user interaction with the grid. It cannot be triggered by code.

Note When the user selects a column, any row selections are cleared. Similarly, when the user selects a row, any column selections are cleared.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Windows Controls

Visual Studio 6.0

SelChange Event (MonthView Control)

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

Occurs when the user selects a new date or range of dates.

Syntax

Private Sub *object_SelChange*(*StartDate As Date*, *EndDate As Date*, *Cancel As Boolean*)

The SelChange Event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>StartDate</i>	The first date in the selection.
<i>EndDate</i>	The last date in the selection
<i>Cancel</i>	Determines whether the selection is canceled.

© 2018 Microsoft

Visual Basic: Windows Controls

SelChange Event Example

The example prints out the range of dates selected by the user. To try the example, place a **MonthView** control on a form and paste the code into the Declarations section of the code module. Start the project and select various date ranges (set the **MultiSelect** property to **True**).

```
Private Sub MonthView1_SelChange(ByVal StartDate As Date, ByVal EndDate As Date, Cancel As Boolean)
    Dim d As Date
    d = StartDate
    Debug.Print "Start"
    While d <= EndDate
        Debug.Print d
        d = d + 1
    Wend
    Debug.Print "End"
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: MSFlexGrid/MSHFlexGrid Controls

Visual Studio 6.0

SelChange Event (MSHFlexGrid)

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

Occurs when the selected range changes to a different cell or range of cells.

Syntax

Private Sub *object*_SelChange()

The SelChange event syntax has these parts:

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

Remarks

The SelChange event occurs when the user clicks a new cell, drags to select a new range of cells, or selects a range of cells by pressing and holding SHIFT while using the ARROW keys. This event does not occur while a user drags the selection across the **MSHFlexGrid**.

This event can occur programmatically by changing the selected region using the **Row**, **Col**, **RowSel**, or **ColSel** properties.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: RichTextBox Control

Visual Studio 6.0

SelChange Event (RichTextBox Control)

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

Occurs when the current selection of text in the **RichTextBox** control has changed or the insertion point has moved.

Syntax

Private Sub *object_SelChange*(*[index As Integer]*)

The SelChange event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that uniquely identifies a control if it's in a control array.

Remarks

You can use the SelChange event to check the various properties that give information about the current selection (such as **SelBold**) so you can update buttons in a toolbar, for example.

© 2018 Microsoft

Visual Basic: RichTextBox Control

SelChange Event Example

This example checks the size of the current selection to see if the menu commands for cutting or copying text to the Clipboard should be enabled. To try this example, put a **RichTextBox** control and three **Menu** controls on a form to create a menu with commands to cut and copy. Paste this code into the SelChange event of the **RichTextBox** control. Then run the example.

```
Private Sub RichTextBox1_SelChange()  
    If RichTextBox1.SelectionLength > 0 Then  
        EditCutMenu.Enabled = True  
        EditCopyMenu.Enabled = True  
    Else  
        EditCutMenu.Enabled = False  
        EditCopyMenu.Enabled = False  
    End If  
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Winsock Control

Visual Studio 6.0

SendComplete Event

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

Occurs when a send operation is completed.

Syntax

object.**SendComplete**

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

Arguments

None

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Winsock Control

Visual Studio 6.0

SendProgress Event

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

Occurs while data is being sent.

Syntax

object.**SendProgress** (*bytesSent* **As Long**, *bytesRemaining* **As Long**)

The SendProgress event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>bytesSent</i>	The number of bytes that have been sent since the last time this event was activated.
<i>bytesRemaining</i>	The number of bytes in the send buffer waiting to be sent.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

SeriesActivated Event

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

Occurs when the user double clicks a chart series. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Private Sub *object_SeriesActivated* (*series* **As Integer**, *mouseFlags* **As Integer**, *cancel* **As Integer**)

The SeriesActivated event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>series</i>	Integer. Identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.
<i>mouseFlags</i>	Integer. Indicates whether a key is held down when the mouse button is clicked, as described in Settings.
<i>cancel</i>	Integer. This argument is not used at this time.

Settings

The event handler determines if a key is held down when the mouse button is clicked and sets *mouseFlags* to:

Constants	Description
VtChMouseFlagShiftKeyDown	If the SHIFT key is held down.
VtChMouseFlagControlKeyDown	If the CONTROL key is held down.

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

SeriesSelected Event

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

Occurs when the user clicks a chart series.

Syntax

Private Sub *object_SeriesSelected* (*series* **As Integer**, *mouseFlags* **As Integer**, *cancel* **As Integer**)

The SeriesSelected event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>series</i>	Integer. Identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.
<i>mouseFlags</i>	Integer. Indicates whether a key is held down when the mouse button is clicked, as described in Settings.
<i>cancel</i>	Integer. This argument is not used at this time.

Settings

The event handler determines if a key is held down when the mouse button is clicked and sets *mouseFlags* to:

Constants	Description
VtChMouseFlagShiftKeyDown	If the SHIFT key is held down.
VtChMouseFlagControlKeyDown	If the CONTROL key is held down.

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

SeriesUpdated Event

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

Occurs when a chart series has changed.

Syntax

Private Sub *object_SeriesUpdated* (*series* **As Integer**, *updateFlags* **As Integer**)

The SeriesUpdated event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>series</i>	Integer. Identifies the series containing the data point. Series are numbered in the order that their columns appear in the data grid, beginning with 1.
<i>updateFlags</i>	Integer. Provides information about the update of the series, as described in Settings.

Settings

The following table lists the constants for *updateFlags*.

Constant	Description
VtChNoDisplay	Absence of update flags; the chart display is not affected. (Defined as 0.)
VtChDisplayPlot	Update will cause the plot to repaint.
VtChLayoutPlot	Update will cause the plot to lay out.
VtChDisplayLegend	Update will cause the legend to repaint.
VtChLayoutLegend	Update will cause the legend to lay out.
VtChLayoutSeries	Update will cause the series to lay out.
VtChPositionSection	A chart section has been moved or resized.

This documentation is archived and is not being maintained.

Visual Basic: SysInfo Control

Visual Studio 6.0

SettingChanged Event

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

Occurs when an application changes a systemwide parameter.

Syntax

```
Private Sub object_SettingChanged([index As Integer, ByVal item As Integer, ByVal section As Integer])
```

The SettingChanged event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	A numeric expression that evaluates to the index of a control if it is in a control array.
<i>item</i>	A long integer that contains system parameter information that has changed. If the message is not sent from USER32, then that system parameter (which is the WPARAM from the WM_SETTINGCHANGE and corresponds to a parameter for the win32 API routine, SystemParameterInfo) is always zero. Many of the various control panel applets cause this message to be sent, but they do it themselves, not from USER32. Under these conditions, you may get 0 as the value in this event.
<i>section</i>	A string expression that evaluates to the name of the section containing the system parameter that changed.

Remarks

You can use the SettingChanged event to detect a number of useful system changes. One of the most useful changes you can detect is when a user moves or resizes the Windows taskbar.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Show Event (UserControl Object)

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

Occurs when the objects **Visible** property changes to **True**.

Syntax

Sub *object_Show()*

The **Show** event syntax has these parts:

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

Remarks

In order to draw to the screen in Windows, any object must have a window, temporarily or permanently. Visual Basic ActiveX controls have permanent windows. Before a control has been sited on a form, its window is not on the container. The control receives Show events when the window is added.

While the controls window is on the form, the control receives a Show event when the controls **Visible** property changes to **True**.

The control does *not* receive Show events if the form is hidden and then shown again, or if the form is minimized and then restored. The controls window remains on the form during these operations, and its **Visible** property doesnt change.

If the control is being shown in an internet browser, a Show event occurs if the user returns to the page containing the control.

Note If the control is used with earlier versions of Visual Basic than 5.0, the control will not receive Show events at design time. This is because earlier versions of Visual Basic did not put any visible windows on a form at design time.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Show Event (UserDocument Object)

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

Occurs when the objects **Visible** property changes to **True**.

Syntax

Sub *object_Show()*

The Show event syntax has these parts:

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

Remarks

In order to draw to the screen in Windows, any object must have a window, temporarily or permanently; Visual Basic ActiveX documents have permanent windows. Before *object* has been sited on a form, its window is not on the container. The UserDocument object receives Show events when the window is added.

While *objects* window is on the container, *object* receives a Show event when *objects* **Visible** property changes to **True**.

Object does *not* receive Show events if the container is hidden and then shown again, or if the container is minimized and then restored. *Objects* window remains on the container during these operations, and its **Visible** property doesn't change.

If *object* is being shown in an internet browser, a Show event occurs when the user navigates to the page.

Note If *object* is used with earlier versions of Visual Basic than 5.0, *object* will not receive Show events at design time. This is because earlier versions of Visual Basic did not put any visible windows on a form at design time.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: DataGrid Control

Visual Studio 6.0

SplitChange Event

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

Occurs when the current cell changes to a different cell in another split.

Syntax

Private Sub *object_SplitChange*([*index* **As Integer**])

The SplitChange event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that identifies a control if it is in a control array.

Remarks

This event is triggered under several circumstances:

- When the grid is first displayed.
- When the user clicks a cell in another split (subject to the setting of the **AllowFocus** property).
- When the user presses a navigation key to cross a split boundary (subject to the setting of the **TabAcrossSplits** property).
- When the **Split** property is changed in code to a different value.
- When a new split is inserted before the current split via code or user interaction.
- When the current split is removed via code or user interaction.

If the user edits data and then moves the current cell position to a new row in another split, the update events for the original row are completed before the SplitChange event is executed.

If a split change also results in a change to the current row or column, then the SplitChange event will always precede the RowColChange event.

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Start Event

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

Occurs when a user launches a **WebClass** object from the browser, if no webitem and event are specified in the request. The Start event follows the BeginRequest event in the application's lifetime.

Syntax

Private Sub *object_Start()*

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

Remark

The Start event occurs when the user accesses your application by navigating to the Active Server Page that forms the applications base URL. You can write code for the Start event in order for the system to successfully run your application in this situation.

Note If the user includes a webitem in the URL they use to launch the webclass, the Start event does not fire. Instead, the run-time fires the BeginRequest event, then launches the specified webitem and its default event or the event the user specified.

Generally, you should use the Start event to navigate to the first **WebItem** in your **WebClass** object. The Start event code can use the **NextItem** property to navigate to the specified **WebItem** in the **Webclass**.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Internet Control

Visual Studio 6.0

StateChanged Event

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

Occurs whenever there is a state change in the connection.

Syntax

object **StateChanged**(ByVal *State* As Integer)

The StateChanged event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>State</i>	Integer. Specifies the state, as shown in Settings below.

Settings

The settings for *State* are:

Constant	Value	Description
icNone	0	No state to report.
icHostResolvingHost	1	The control is looking up the IP address of the specified host computer.
icHostResolved	2	The control successfully found the IP address of the specified host computer.
icConnecting	3	The control is connecting to the host computer.
icConnected	4	The control successfully connected to the host computer.
icRequesting	5	The control is sending a request to the host computer.
icRequestSent	6	The control successfully sent the request.
icReceivingResponse	7	The control is receiving a response from the host computer.
icResponseReceived	8	The control successfully received a response from the host computer.

icDisconnecting	9	The control is disconnecting from the host computer.
icDisconnected	10	The control successfully disconnected from the host computer.
icError	11	An error occurred in communicating with the host computer.
icResponseCompleted	12	The request has completed and all data has been received.

Remarks

In general, you will use the StateChanged event to determine when to retrieve data using the **GetChunk** method. To do this, use a **Select Case** statement and test for **icResponseReceived** (8) or **icResponseCompleted** (12).

Note, however, that the **icResponseReceived** state may occur when the control has completed an operation that hasn't resulted in any data in the buffer. For example, when connecting to an FTP site, the control will perform a "handshake" with the site that doesn't result in any data transfer, yet the **icResponseReceived** state will occur.

On the other hand, the **icResponseCompleted** state occurs after an operation has completed in its entirety. For example, if you are using the **Execute** method with the GET operation to retrieve a file, the **icResponseCompleted** event will occur only once after the file has been totally retrieved.

In practice, using the **icResponseReceived** state allows you to parse the data until you have retrieved only the information you need (for example, when retrieving an HTML file, retrieving only the headers). Once you have the information, you can cancel the retrieval. On the other hand, if you are intent on retrieving the whole file, the **icResponseCompleted** state will notify you that the transfer is completed, allowing you to proceed.

© 2018 Microsoft

Visual Basic: Internet Control

GetChunk Method, StateChanged Event Example

The example uses the **GetChunk** method in the StateChanged event to retrieve a chunk of data. The example uses a **Select Case** statement to determine what to do with every possible state. The example assumes a **TextBox** control named **txtData** exists on the form.

```
Private Sub Inet1_StateChanged(ByVal State As Integer)
    ' Retrieve server response using the GetChunk
    ' method when State = 12. This example assumes the
    ' data is text.

    Select Case State
        ' ... Other cases not shown.

    Case icResponseReceived ' 12
        Dim vtData As Variant ' Data variable.
        Dim strData As String: strData = ""
        Dim bDone As Boolean: bDone = False

        ' Get first chunk.
        vtData = Inet1.GetChunk(1024, icString)
        DoEvents
        Do While Not bDone
            strData = strData & vtData
            DoEvents
            ' Get next chunk.
            vtData = Inet1.GetChunk(1024, icString)
            If Len(vtData) = 0 Then
                bDone = True
            End If
        Loop

        txtData.Text = strData
    End Select

End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: Multimedia MCI Control

Visual Studio 6.0

StatusUpdate Event (Multimedia MCI Control)

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

Occurs automatically at intervals given by the **UpdateInterval** property.

Syntax

Private Sub *MMControl_StatusUpdate* ()

Remarks

This event allows an application to update the display to inform the user about the status of the current MCI device. The application can obtain status information from properties such as **Position**, **Length**, and **Mode**.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: SysInfo Control

Visual Studio 6.0

SysColorsChanged Event

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

Occurs when a system color setting changes, either by an application or through the Control Panel.

Syntax

Private Sub *object_SysColorsChanged*([(index **As Integer**)])

The SysColorsChanged event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	A numeric expression that evaluates to the index of a control if it is in a control array.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Terminate Event

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

Occurs when all references to an instance of a **Form**, **MDIForm**, **User** control, **Property Page**, **Webclass**, **DHTML Page Designer**, or class are removed from memory by setting all the variables that refer to the object to **Nothing** or when the last reference to the object falls out of scope.

Syntax

Private Sub *object*_Terminate()

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

Remarks

For all objects except classes, the Terminate event occurs after the Unload event.

The Terminate event isn't triggered if the instances of the form or class were removed from memory because the application terminated abnormally. For example, if your application invokes the **End** statement before removing all existing instances of the class or form from memory, the Terminate event isn't triggered for that class or form.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic: SysInfo Control

Visual Studio 6.0

TimeChanged Event

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

Occurs when the system time changes, either by an application or through the Control Panel.

Syntax

Private Sub *object*.TimeChanged(*[index As Integer]*)

The TimeChanged event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	A numeric expression that evaluates to the index of a control if it is in a control array.

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Basic Reference

Visual Studio 6.0

Timer Event

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

Occurs when a preset interval for a **Timer** control has elapsed. The interval's frequency is stored in the control's **Interval** property, which specifies the length of time in milliseconds.

Syntax

```
Private Sub object_Timer([index As Integer])
```

The Timer event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>index</i>	An integer that uniquely identifies a control if it's in a control array.

Remarks

Use this event procedure to tell Visual Basic what to do after each **Timer** control interval has elapsed. When you're working with the Timer event:

- The **Interval** property specifies the interval between Timer events in milliseconds.
- Whenever the **Timer** control's **Enabled** property is set to **True** and the **Interval** property is greater than 0, the Timer event waits for the period specified in the **Interval** property.

© 2018 Microsoft

Visual Basic Reference

Timer Event Example

This example demonstrates a digital clock. To try this example, paste the code into the Declarations section of a form that contains a **Label** control and a **Timer** control, and then press F5.

```
Private Sub Form_Load ()
    Timer1.Interval = 1000 ' Set Timer interval.
End Sub
```

```
Private Sub Timer1_Timer ()
    Label1.Caption = Time ' Update time display.
End Sub
```

This example moves a **PictureBox** control across a form. To try this example, paste the code into the Declarations section of a form that contains a **Timer** control and a **PictureBox** control, and then press F5. For a better visual effect you can assign a bitmap to the **PictureBox** using the **Picture** property.

```
Dim DeltaX, DeltaY As Integer ' Declare variables.
Private Sub Timer1_Timer ()
    Picture1.Move Picture1.Left + DeltaX, Picture1.Top + DeltaY
    If Picture1.Left < ScaleLeft Then DeltaX = 100
    If Picture1.Left + Picture1.Width > ScaleWidth + ScaleLeft Then
        DeltaX = -100
    End If
    If Picture1.Top < ScaleTop Then DeltaY = 100
    If Picture1.Top + Picture1.Height > ScaleHeight + ScaleTop Then
        DeltaY = -100
    End If
End Sub

Private Sub Form_Load ()
    Timer1.Interval = 1000 ' Set Interval.
    DeltaX = 100 ' Initialize variables.
    DeltaY = 100
End Sub
```

© 2018 Microsoft

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

TitleActivated Event

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

Occurs when the user double clicks the chart title. You can replace the standard user interface by canceling the event and displaying your own dialog box.

Syntax

Private Sub *object*_**TitleActivated** (*mouseFlags* **As Integer**, *cancel* **As Integer**)

The TitleActivated event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>mouseFlags</i>	Integer. Indicates whether a key is held down when the mouse button is clicked, as described in Settings.
<i>cancel</i>	Integer. This argument is not used at this time.

Settings

The event handler determines if a key is held down when the mouse button is clicked and sets *mouseFlags* to:

Constants	Description
VtChMouseFlagShiftKeyDown	If the SHIFT key is held down.
VtChMouseFlagControlKeyDown	If the CONTROL key is held down.

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

TitleSelected Event

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

Occurs when the user clicks the chart title.

Syntax

Private Sub *object*_**TitleSelected** (*mouseFlags* **As Integer**, *cancel* **As Integer**)

The TitleSelected event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>mouseFlags</i>	Integer. Indicates whether a key is held down when the mouse button is clicked, as described in Settings.
<i>cancel</i>	Integer. This argument is not used at this time.

Settings

The event handler determines if a key is held down when the mouse button is clicked and sets *mouseFlags* to:

Constants	Description
VtChMouseFlagShiftKeyDown	If the SHIFT key is held down.
VtChMouseFlagControlKeyDown	If the CONTROL key is held down.

This documentation is archived and is not being maintained.

Visual Studio 6.0

Visual Basic: MSChart Control

TitleUpdated Event

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

Occurs when the chart title has changed.

Syntax

Private Sub *object*_**TitleUpdated** (*updateFlags* **As Integer**)

The TitleUpdated event syntax has these parts:

Part	Description
<i>object</i>	An object expression that evaluates to an object in the Applies To list.
<i>updateFlags</i>	Provides information about the update of the title, as described in Settings.

Settings

The following table lists the constants for *updateFlags*.

Constant	Description
VtChNoDisplay	Absence of update flags; the chart display is not affected. (Defined as 0.)
VtChDisplayPlot	Update will cause the plot to repaint.
VtChLayoutPlot	Update will cause the plot to lay out.
VtChDisplayLegend	Update will cause the legend to repaint.
VtChLayoutLegend	Update will cause the legend to lay out.
VtChLayoutSeries	Update will cause the series to lay out.
VtChPositionSection	A chart section has been moved or resized.