Click Event Example

The following example illustrates how you can set up code for a Click event procedure using **WithEvents** and **Set**. Note that the object reference `ce` is used in place of the menu name **Tools** in the name of the Click event.

```
Private WithEvents ce As CommandBarEvents

Sub Test()
    Dim c As CommandBarControl
    Set c = Application.VBE.CommandBars("Tools").Controls(1)
    Set ce = Application.VBE.Events.CommandBarEvents(c)
End Sub

Private Sub ce_Click(ByVal CommandBarControl As Object, Handled As Boolean,
CancelDefault As Boolean)
    ' Put event-handling code here
End Sub
```

Click Event

Occurs when the **OnAction** property of a corresponding command bar control is set.

Syntax

Sub *object*_**Click (ByVal** ***ctrl*** **As Object, ByRef** ***handled*** **As Boolean, ByRef** ***canceldefault*** **As Boolean)**

The Click event syntax has these ***named arguments***:

Part	Description
ctrl	Required; **Object**. Specifies the object that is the source of the Click event.
handled	Required; **Boolean**. If **True**, other add-ins should handle the event. If **False**, the action of the command bar item has not been handled.
canceldefault	Required; **Boolean**. If **True**, default behavior is performed unless canceled by a downstream add-in. If **False**, default behavior is not performed unless restored by a downstream add-in.

Remarks

The Click event is specific to the **CommandBarEvents** object. Use a variable declared using the **WithEvents** keyword to receive the Click event for a **CommandBar** control. This variable should be set to the return value of the **CommandBarEvents** property of the **Events** object. The **CommandBarEvents** property takes the **CommandBar** control as an argument. When the **CommandBar** control is clicked (for the variable you declared using the **WithEvents** keyword), the code is executed.

ItemAdded Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaevtItemAddedC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaevtItemAddedX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaevtItemAddedA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaevtItemAddedS"}

Occurs after a reference is added.

Syntax

Sub *object*_**ItemAdded(ByVal** *item* **As Reference)**

The required *item* argument specifies the item that was added.

Remarks

The ItemAdded event occurs when a **Reference** is added to the **References** collection.

ItemRemoved Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaevtItemRemovedC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaevtItemRemovedX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaevtItemRemovedA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaevtItemRemovedS"}

Occurs after a reference is removed from a project.

Syntax

Sub *object*_**ItemRemoved(ByVal** *item* **As Reference)**

The required *item* argument specifies the **Reference** that was removed.

Add Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddinAddC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddinAddX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddinAddA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddinAddS"}

Adds an object to a collection.

Syntax

object.**Add(***component***)**

The **Add** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
component	Required. For the **LinkedWindows** collection, an object. For the **VBComponents** collection, an enumerated constant representing a class module, a form, or a standard module.

You can use one of the following constants for the *component* argument:

Constant	Description
vbext_ct_ClassModule	Adds a class module to the collection.
vbext_ct_MSForm	Adds a form to the collection.
vbext_ct_StdModule	Adds a standard module to the collection.

Remarks

For the **LinkedWindows** collection, the **Add** method adds a window to the collection of currently linked windows.

Note You can add a window that is a pane in one linked window frame to another linked window frame; the window is simply moved from one pane to the other. If the linked window frame that the window was moved from no longer contains any panes, it's destroyed.

For the **VBComponents** collection, the **Add** method creates a new standard component and adds it to the project.

For the **VBComponents** collection, the **Add** method returns a **VBComponent** object. For the **LinkedWindows** collection, the **Add** method returns **Nothing**.

AddFromFile Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddFromFileC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddFromFileX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddFromFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddFromFileS"}

For the **References** collection, adds a reference to a project from a file. For the **CodeModule** object, adds the contents of a file to a module.

Syntax

object.**AddFromFile(***filename***)**

The **AddFromFile** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
filename	Required. A string expression specifying the name of the file you want to add to the project or module. If the file name isn't found and a path name isn't specified, the directories searched by the **Windows OpenFile** function are searched.

Remarks

For the **CodeModule** object, the **AddFromFile** method inserts the contents of the file starting on the line preceding the first procedure in the code module. If the module doesn't contain procedures, **AddFromFile** places the contents of the file at the end of the module.

AddFromGuid Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddFromGuidC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddFromGuidX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddFromGuidA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddFromGuidS"}

Adds a reference to the **References** collection using the globally unique identifier (GUID) of the reference.

Syntax

object.**AddFromGuid(***guid***,** *major***,** *minor***) As Reference**

The **AddFromGuid** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
guid	Required. A string expression representing the GUID of the reference.
major	Required. A **Long** specifying the major version number of the reference.
minor	Required. A **Long** specifying the minor version number of the reference.

Remarks

The **AddFromGuid** method searches the registry to find the reference you want to add. The GUID can be a type library, control, class identifier, and so on.

AddFromString Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddFromStringC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddFromStringX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddFromStringA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddFromStringS"}

Adds text to a module.

Syntax

object.**AddFromString**

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

Remarks

The **AddFromString** method inserts the text starting on the line preceding the first procedure in the module. If the module doesn't contain procedures, **AddFromString** places the text at the end of the module.

Close Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthCloseC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthCloseX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthCloseS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthCloseA"}

Closes and destroys a window.

Syntax

object.**Close**

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

Remarks

The following types of windows respond to the **Close** method in different ways:

- For a window that is a code pane, **Close** destroys the code pane.
- For a window that is a designer, **Close** destroys the contained designer.
- For windows that are always available on the **View** menu, **Close** hides the window.

CreateEventProc Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthCreateEventProcC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthCreateEventProcX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthCreateEventProcA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthCreateEventProcS"}

Creates an event procedure.

Syntax

object.**CreateEventProc(***eventname*, *objectname*) **As Long**

The **CreateEventProc** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
eventname	Required. A string expression specifying the name of the event you want to add to the module.
objectname	Required. A string expression specifying the name of the object that is the source of the event.

Remarks

Use the **CreateEventProc** method to create an event procedure. For example, to create an event procedure for the Click event of a **Command Button** control named `Command1` you would use the following code, where `CM` represents a object of type **CodeModule**:

```
TextLocation = CM.CreateEventProc("Click", "Command1")
```

The **CreateEventProc** method returns the line at which the body of the event procedure starts. **CreateEventProc** fails if the arguments refer to a nonexistent event.

DeleteLines Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthDeleteLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthDeleteLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthDeleteLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthDeleteLinesS"}

Deletes a single line or a specified range of lines.

Syntax

object.**DeleteLines (***startline* [, *count*]**)**

The **DeleteLines** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
startline	Required. A **Long** specifying the first line you want to delete.
count	Optional. A **Long** specifying the number of lines you want to delete.

Remarks

If you don't specify how many lines you want to delete, **DeleteLines** deletes one line.

Export Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthExportC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthExportX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthExportA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthExportS"}

Saves a component as a separate file or files.

Syntax

object.**Export(***filename***)**

The **Export** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
filename	Required. A **String** specifying the name of the file that you want to export the component to.

Remarks

When you use the **Export** method to save a component as a separate file or files, use a file name that doesn't already exist; otherwise, an error occurs.

Find Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthFindC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthFindX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthFindS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthFindA"}

Searches the active module for a specified string.

Syntax

object.**Find(***target***,** *startline***,** *startcol***,** *endline***,** *endcol* [, *wholeword*] [, *matchcase*] [, *patternsearch*]**) As Boolean**

The **Find** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
target	Required. A **String** containing the text or pattern you want to find.
startline	Required. A **Long** specifying the line at which you want to start the search; will be set to the line of the match if one is found.
startcol	Required. A **Long** specifying the column at which you want to start the search; will be set to the column containing the match if one is found.
endline	Required. A **Long** specifying the last line of the match if one is found.
endcol	Required. A **Long** specifying the last line of the match if one is found.
wholeword	Optional. A **Boolean** value specifying whether to only match whole words. If **True**, only matches whole words. **False** is the default.
matchcase	Optional. A **Boolean** value specifying whether to match case. If **True**, the search is case sensitive. **False** is the default.
patternsearch	Optional. A **Boolean** value specifying whether or not the target string is a regular expression pattern. If **True**, the target string is a regular expression pattern. **False** is the default.

Remarks

Find returns **True** if a match is found and **False** if a match isn't found.

The *matchcase* and *patternmatch* arguments are mutually exclusive; if both arguments are passed as **True**, an error occurs.

The content of the **Find** dialog box isn't affected by the **Find** method.

GetSelection Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthGetSelectionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthGetSelectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthGetSelectionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthGetSelectionS"}

Returns the selection in a code pane.

Syntax

object.**GetSelection(***startline*, *startcol*, *endline*, *endcol***)**

The **GetSelection** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
startline	Required. A **Long** that returns a value specifying the first line of the selection in the code pane.
startcol	Required. A **Long** that returns a value specifying the first column of the selection in the code pane.
endline	Required. A **Long** that returns a value specifying the last line of the selection in the code pane.
endcol	Required. A **Long** that returns a value specifying the last column of the selection in the code pane.

Remarks

When you use the **GetSelection** method, information is returned in output arguments. As a result, you must pass in variables because the variables will be modified to contain the information when returned.

Import Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthImportC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthImportX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthImportA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthImportS"}

Adds a component to a project from a file; returns the newly added component.

Syntax

object.**Import(***filename***) As VBComponent**

The **Import** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
filename	Required. A **String** specifying path and file name of the component that you want to import the component from.

Remarks

You can use the **Import** method to add a component, form, module, class, and so on, to your project.

InsertLines Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthInsertLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthInsertLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthInsertLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthInsertLinesS"}

Inserts a line or lines of code at a specified location in a block of code.

Syntax

object.**InsertLines**(*line*, *code*)

The **InsertLines** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
line	Required. A **Long** specifying the location at which you want to insert the code.
code	Required. A **String** containing the code you want to insert.

Remarks

If the text you insert using the **InsertLines** method is carriage return–linefeed delimited, it will be inserted as consecutive lines.

Item Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddinItemC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddinItemX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddinItemA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddinItemS"}

Returns the indexed member of a collection.

Syntax

object.**Item(***index***)**

The **Item** method syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
index	Required. An expression that specifies the position of a member of the collection. If a numeric expression, *index* must be a number from 1 to the value of the collection's **Count** property. If a string expression, *index* must correspond to the ***key*** argument specified when the member was added to the collection.

The following table lists the collections and their corresponding ***key*** arguments for use with the **Item** method. The string you pass to the **Item** method must match the collection's ***key*** argument.

Collection	Key argument
Windows	**Caption** property setting
LinkedWindows	**Caption** property setting
CodePanes	No unique string is associated with this collection.
VBProjects	**Name** property setting
VBComponents	**Name** property setting
References	**Name** property setting
Properties	**Name** property setting

Remarks

The *index* argument can be a numeric value or a string containing the title of the object.

Lines Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthLinesS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthLinesA"}

Returns a specified line of code.

Syntax

object.**Lines(***startline*, *count***) As String**

The **Lines** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
startline	Required. A **Long** specifying the first line of the code you want to return.
count	Required. A **Long** specifying the number of lines you want to return.

Remarks

The line numbers in a code module begin at 1.

ProcBodyLine Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthProcBodyLineC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthProcBodyLineX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthProcBodyLineA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthProcBodyLineS"}

Returns the first line of a procedure.

Syntax

object.**ProcBodyLine(***procname***,** *prockind***) As Long**

The **ProcBodyLine** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
procname	Required. A **String** containing the name of the procedure.
prockind	Required. Specifies the kind of procedure to locate. Because property procedures can have multiple representations in the module, you must specify the kind of procedure you want to locate. All procedures other than property procedures (that is, **Sub** and **Function** procedures) use **vbext_pk_Proc**.

You can use one of the following constants for the *prockind* argument:

Constant	Description
vbext_pk_Get	Specifies a procedure that returns the value of a property.
vbext_pk_Let	Specifies a procedure that assigns a value to a property.
vbext_pk_Set	Specifies a procedure that sets a reference to an object.
vbext_pk_Proc	Specifies all procedures other than property procedures.

Remarks

The first line of a procedure is the line on which the **Sub**, **Function**, or **Property** statement appears.

ProcCountLines Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthProcCountLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthProcCountLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthProcCountLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthProcCountLinesS"}

Returns the number of lines in the specified procedure.

Syntax

object.**ProcCountLines(***procname***,** *prockind***) As Long**

The **ProcCountLines** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
procname	Required. A **String** containing the name of the procedure.
prockind	Required. Specifies the kind of procedure to locate. Because property procedures can have multiple representations in the module, you must specify the kind of procedure you want to locate. All procedures other than property procedures (that is, **Sub** and **Function** procedures) use **vbext_pk_Proc**.

You can use one of the following constants for the *prockind* argument:

Constant	Description
vbext_pk_Get	Specifies a procedure that returns the value of a property.
vbext_pk_Let	Specifies a procedure that assigns a value to a property.
vbext_pk_Set	Specifies a procedure that sets a reference to an object.
vbext_pk_Proc	Specifies all procedures other than property procedures.

Remarks

The **ProcCountLines** method returns the count of all blank or comment lines preceding the procedure declaration and, if the procedure is the last procedure in a code module, any blank lines following the procedure.

ProcOfLine Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthProcOfLineC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthProcOfLineX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthProcOfLineA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthProcOfLineS"}

Returns the name of the procedure that the specified line is in.

Syntax

object.**ProcOfLine**(*line*, *prockind*) **As String**

The **ProcOfLine** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
line	Required. A **Long** specifying the line to check.
prockind	Required. Specifies the kind of procedure to locate. Because property procedures can have multiple representations in the module, you must specify the kind of procedure you want to locate. All procedures other than property procedures (that is, **Sub** and **Function** procedures) use **vbext_pk_Proc**.

You can use one of the following constants for the *prockind* argument:

Constant	Description
vbext_pk_Get	Specifies a procedure that returns the value of a property.
vbext_pk_Let	Specifies a procedure that assigns a value to a property.
vbext_pk_Set	Specifies a procedure that sets a reference to an object.
vbext_pk_Proc	Specifies all procedures other than property procedures.

Remarks

A line is within a procedure if it's a blank line or comment line preceding the procedure declaration and, if the procedure is the last procedure in a code module, a blank line or lines following the procedure.

ProcStartLine Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthProcStartLineC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthProcStartLineX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthProcStartLineA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthProcStartLineS"}

Returns the line at which the specified procedure begins.

Syntax

object.**ProcStartLine(***procname***,** *prockind***) As Long**

The **ProcStartLine** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
procname	Required. A **String** containing the name of the procedure.
prockind	Required. Specifies the kind of procedure to locate. Because property procedures can have multiple representations in the module, you must specify the kind of procedure you want to locate. All procedures other than property procedures (that is, **Sub** and **Function** procedures) use **vbext_pk_Proc**.

You can use one of the following constants for the *prockind* argument:

Constant	Description
vbext_pk_Get	Specifies a procedure that returns the value of a property.
vbext_pk_Let	Specifies a procedure that assigns a value to a property.
vbext_pk_Set	Specifies a procedure that sets a reference to an object.
vbext_pk_Proc	Specifies all procedures other than property procedures.

Remarks

A procedure starts at the first line below the **End Sub** statement of the preceding procedure. If the procedure is the first procedure, it starts at the end of the general Declarations section.

Remove Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthAddinRemoveC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthAddinRemoveX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthAddinRemoveA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthAddinRemoveS"}

Removes an item from a collection.

Syntax

object.**Remove(***component***)**

The **Remove** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
component	Required. For the **LinkedWindows** collection, an object. For the **References** collection, a reference to a type library or a project. For the **VBComponents** collection, an enumerated constant representing a class module, a form, or a standard module.

Remarks

When used on the **LinkedWindows** collection, the **Remove** method removes a window from the collection of currently linked windows. The removed window becomes a floating window that has its own linked window frame.

ReplaceLine Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthReplaceLineC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthReplaceLineX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthReplaceLineA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthReplaceLineS"}

Replaces an existing line of code with a specified line of code.

Syntax

object.**ReplaceLine**(*line*, *code*)

The **ReplaceLine** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
line	Required. A **Long** specifying the location of the line you want to replace.
code	Required. A **String** containing the code you want to insert.

SetFocus Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthSetFocusC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthSetFocusX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthSetFocusA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthSetFocusS"}

Moves the focus to the specified window.

Syntax

object.**SetFocus**

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

Remarks

Use the **SetFocus** method on windows that are already visible.

SetSelection Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthSetSelectionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthSetSelectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthSetSelectionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthSetSelectionS"}

Sets the selection in the code pane.

Syntax

object.**SetSelection(***startline*, *startcol*, *endline*, *endcol***)**

The **SetSelection** syntax has these parts:

Part	Description
object	Required. An object expression that evaluates to an object in the Applies To list.
startline	Required. A **Long** specifying the first line of the selection.
startcol	Required. A **Long** specifying the first column of the selection.
endline	Required. A **Long** specifying the last line of the selection.
endcol	Required. A **Long** specifying the last column of the selection.

Show Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vamthShowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vamthShowX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vamthShowA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vamthShowS"}

Makes the specified code pane the visible code pane in its window.

Syntax

object.**Show**

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

Remarks

The **Show** method makes the specified code pane the pane with the focus in its window.

Add Method Example

The following example uses the **Add** method to add one standard module to the **VBComponents** collection.

```
Application.VBE.VBProjects(1).VBComponents.Add(vbext_ct_StdModule)
```

AddFromFile Method Example

The following example uses the **AddFromFile** method to add the contents of a file to a specified code pane.

```
Application.VBE.CodePanes(3).CodeModule.AddFromFile "c:\Code
Files\book2.frm"
```

AddFromGUID Method Example

The following example uses the **AddFromGUID** method to add a reference to the current project, identifying the reference using the globally unique ID value of the **Reference** object.

```
Application.VBE.ActiveVBProject.References.AddFromGuid("{000204EF-0000-
0000-C000-000000000046}", 5, 0)
```

AddFromString Method Example

The following example uses the **AddFromString** method to add a line, “Dim intJack As Integer,” to the specified code pane.

```
Application.VBE.CodePanes(3).CodeModule.AddFromString "Dim intJack As
Integer"
```

Close Method Example

The following example uses the **Close** method to close a specified member of the **Windows** collection.

```
Application.VBE.Windows(9).Close
```

CreateEventProc Method Example

The following example uses the **CreateEventProc** method to create the Button_Click procedure.

```
Debug.Print
Application.VBE.SelectVBComponents.CodeModule.CreateEventProc("Click",
"Button")
```

DeleteLines Method Example

The following example has two steps. The first **For…Next** loop uses the **InsertLines** method to insert into `CodePanes(1)` 26 ever-longer initial segments of the alphabet, starting with “a.” The last line inserted is the entire alphabet.

The second **For…Next** loop uses the **DeleteLines** method to delete the odd-numbered lines. Although it seems that the second loop should simply delete every other line, note that after each deletion the lines get renumbered. Therefore the deletion is advancing by two lines at each step, one line because I is increasing by one and another line because the larger line numbers are each decreasing by one.

```
For I = 1 to 26
   Application.VBE.SelectedVBComponent.CodeModule.InsertLines i, Mid$
("abcdefghijklmnopqrstuvwxyz", 1, I)
Next
For I = 1 to 13
   Application.VBE.SelectedVBComponent.CodeModule.DeleteLines I
Next
```

Export Method Example

The following example creates a file named `test.bas` and uses the **Export** method to copy the contents of the `VBComponents(1)` code module into the file.

```
Application.VBE.ActiveVBProject.VBComponents(1).Export("test.bas")
```

Find Method Example

The following example uses the **Find** method to verify that the specified block of lines, lines 1261 through 1279, of a particular code pane does contain the string “Tabs.Clear.”

```
Application.VBE.CodePanes(2).CodeModule.Find ("Tabs.Clear", 1261, 1, 1280,
1, False, False)
```

GetSelection Method Example

The following example returns the locations of the starting and ending points of the current selection in `CodePanes(1)`. The last line in the example uses the **GetSelection** method to place the four values in the four variables.

```
Dim m As Long
Dim n As Long
Dim x As Long
Dim y As Long
Application.VBE.CodePanes(1).GetSelection m, n, x, y
```

Import Method Example

The following example uses the **Import** method on the **VBComponents** collection to copy the contents of the test.bas file into the a code module.

```
Application.VBE.ActiveVBProject.VBComponents.Import("test.bas")
```

InsertLines Method Example

The following example uses the **InsertLines** method to insert a line, “Option Explicit,” in the specified code pane.

```
Application.VBE.CodePanes(1).CodeModule.InsertLines 1, "Option Explicit"
```

Item Method Example

The following example contains two ways to display a specific member of the **CodePanes** collection, one using the **Item** method.

```
Application.VBE.CodePanes.Item(2).Show
Application.VBE.CodePanes(2).Show
```

Lines Method Example

The following example uses the **Lines** method to return a specific block of code, lines 1 through 4, in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(1).CodeModule.Lines( 1, 4)
```

ProcBodyLine Method Example

The following example uses the **ProcBodyLine** method to return the line number of the first line of code in the specified procedure, SetupTabs, in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(3).CodeModule.ProcBodyLine
("SetupTabs", vbext_pk_Proc)
```

ProcCountLines Method Example

The following example uses the **ProcCountLines** method to return the number of lines of code in the specified procedure, SetupTabs, in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(3).CodeModule.ProcCountLines
("SetupTabs", vbext_pk_Proc)
```

ProcOfLine Method Example

The following example uses the **ProcOfLine** method to return the name of the procedure containing the specified line number in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(3).CodeModule.ProcOfLine (1270,
vbext_pk_Proc)
```

ProcStartLine Method Example

The following example uses the **ProcStartLine** method to return the line at which the specified procedure begins in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(3).CodeModule.ProcStartLine
("SetupTabs", vbext_pk_Proc)
```

Remove Method Example

The example verifies that a particular member of the **VBComponents** collection is a module, and then it uses the **Remove** method to remove the module.

```
Debug.Print Application.VBE.ActiveVBProject.VBComponents(4).Name
Application.VBE.ActiveVBProject.VBComponents.Remove
Application.VBE.ActiveVBProject.VBComponents(4)
```

ReplaceLine Method Example

The following example has two steps. The first **For…Next** loop uses the **InsertLines** method to insert into `CodePanes(1)` 26 ever-longer initial segments of the alphabet, starting with “a.” The last line inserted is the entire alphabet.

The second **For…Next** loop uses the **ReplaceLine** method to replace each even-numbered line with the last letter in the string that previously occupied that line. Odd-numbered lines are unchanged.

```
For I = 1 to 26
   Application.VBE.CodePanes(1).CodeModule.InsertLines I, Mid$
("abcdefghijklmnopqrstuvwxyz", 1, I)
Next I
For I = 1 to 13
   Application.VBE.CodePanes(1).CodeModule.ReplaceLine 2*I, Mid$
("abcdefghijklmnopqrstuvwxyz", 1, I)
Next I
```

SetFocus Method Example

The following example uses the **SetFocus** method to move the focus to a particular member of the **Windows** collection; that is, it makes that window behave as if you had clicked its title bar with your mouse.

```
Application.VBE.Windows(9).SetFocus
```

SetSelection Method Example

The following example uses the **SetSelection** method to select the text whose first character is the one immediately after the fourth character on the second line of `CodePanes(1)` and whose last character is the fifteenth character on the third line.

```
Application.VBE.CodePanes(1).SetSelection 2,4,3,15
```

Show Method Example

The following example uses the **Show** method to move the specified code pane to the foreground.

```
Application.VBE.CodePanes(2).Show
```

CodeModule Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCodeModuleC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjCodeModuleX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjCodeModuleP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjCodeModuleM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjCodeModuleE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjCodeModuleS"}

Represents the code behind a component, such as a form, class, or document.

Remarks

You use the **CodeModule** object to modify (add, delete, or edit) the code associated with a component.

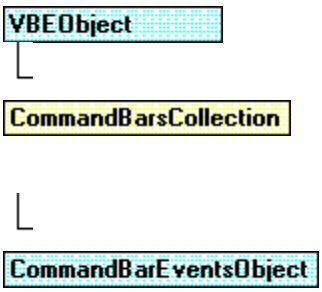
Each component is associated with one **CodeModule** object. However, a **CodeModule** object can be associated with multiple code panes.

The methods associated with the **CodeModule** object enable you to manipulate and return information about the code text on a line-by-line basis. For example, you can use the **AddFromString** method to add text to the module. **AddFromString** places the text just above the first procedure in the module or places the text at the end of the module if there are no procedures.

Use the **Parent** property to return the **VBComponent** object associated with a code module.

CommandBarEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCommandBarEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjCommandBarEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjCommandBarEventsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjCommandBarEventsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjCommandBarEventsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjCommandBarEventsS"}



Returned by the **CommandBarEvents** property. The **CommandBarEvents** object triggers an event when a control on the command bar is clicked.

Remarks

The **CommandBarEvents** object is returned by the **CommandBarEvents** property of the **Events** object. The object that is returned has one event in its interface, the Click event. You can handle this event using the **WithEvents** object declaration.

CommandBars Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCommandBarsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjCommandBarsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjCommandBarsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjCommandBarsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjCommandBarsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjCommandBarsS"}

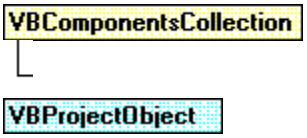
Contains all of the command bars in a project, including command bars that support shortcut menus.

Remarks

Use the **CommandBars** collection to enable add-ins to add command bars and controls or to add controls to existing, built-in, command bars.

VBComponent Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjVBComponentC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjVBComponentX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjVBComponentP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjVBComponentM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjVBComponentE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjVBComponentS"}



Represents a component, such as a class module or standard module, contained in a project.

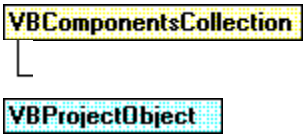
Remarks

Use the **VBComponent** object to access the code module associated with a component or to change a component's property settings.

You can use the **Type** property to find out what type of component the **VBComponent** object refers to. Use the **Collection** property to find out what collection the component is in.

VBComponents Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjVBComponentsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjVBComponentsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjVBComponentsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjVBComponentsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjVBComponentsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjVBComponentsS"}



Represents the components contained in a project.

Remarks

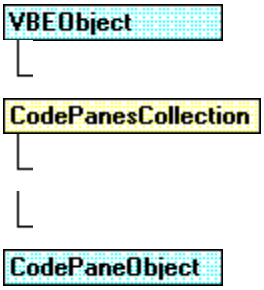
Use the **VBComponents** collection to access, add, or remove components in a project. A component can be a form, module, or class. The **VBComponents** collection is a standard collection that can be used in a **For Each** block.

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

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

CodePane Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCodePaneC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjCodePaneX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjCodePaneP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjCodePaneM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjCodePaneE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjCodePaneS"}



Represents a code pane.

Remarks

Use the **CodePane** object to manipulate the position of visible text or the text selection displayed in the code pane.

You can use the **Show** method to make the code pane you specify visible. Use the **SetSelection** method to set the selection in a code pane and the **GetSelection** method to return the location of the selection in a code pane.

CodePanes Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCodePanesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjCodePanesX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjCodePanesP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjCodePanesM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjCodePanesE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjCodePanesS"}

CodePanesCollection

Contains the active code panes in the **VBE** object.

Remarks

Use the **CodePanes** collection to access the open code panes in a project.

You can use the **Count** property to return the number of active code panes in a collection.

Events Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjEventsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjEventsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjEventsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjEventsS"}

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

Remarks

The **Events** object provides properties that return event source objects. Use the properties to return event source objects that notify you of changes in the Visual Basic for Applications environment.

The properties of the **Events** object return objects of the same type as the property name. For example, the **CommandBarEvents** property returns the **CommandBarEvents** object.

LinkedWindows Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjLinkedWindowsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjLinkedWindowsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjLinkedWindowsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjLinkedWindowsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjLinkedWindowsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjLinkedWindowsS"}

WindowObject

Contains all linked windows in a linked window frame.

Remarks

Use the **LinkedWindows** collection to modify the docked and linked state of windows in the development environment.

The **LinkedWindowFrame** property of the **Window** object returns a **Window** object that has a valid **LinkedWindows** collection.

Linked window frames contain all windows that can be linked or docked. This includes all windows except code windows, designers, the **Object Browser** window, and the **Search and Replace** window.

If all the panes from one linked window frame are moved to another window, the linked window frame with no panes is destroyed. However, if all the panes are removed from the main window, it isn't destroyed.

Use the **Visible** property to check or set the visibility of a window.

You can use the **Add** method to add a window to the collection of currently linked windows. A window that is a pane in one linked window frame can be added to another linked window frame. Use the **Remove** method to remove a window from the collection of currently linked windows; this results in the window being unlinked or undocked.

The **LinkedWindows** collection is used to dock and undock windows from the main window frame.

Property Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjPropertyP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjPropertyM"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjPropertyS"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjPropertyE"}

Represents the properties of an object that are visible in the **Properties** window for any given component.

Remarks

Use **Value** property of the **Property** object to return or set the value of a property of a component.

At a minimum, all components have a **Name** property. Use the **Value** property of the **Property** object to return or set the value of a property. The **Value** property returns a **Variant** of the appropriate type. If the value returned is an object, the **Value** property returns the **Properties** collection that contains **Property** objects representing the individual properties of the object. You can access each of the **Property** objects by using the **Item** method on the returned **Properties** collection.

If the value returned by the **Property** object is an object, you can use the **Object** property to set the **Property** object to a new object.

Properties Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjPropertiesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjPropertiesX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjPropertiesP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjPropertiesM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjPropertiesE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjPropertiesS"}

PropertiesCollection

PropertyObject

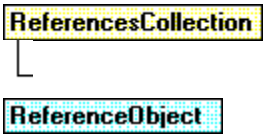
Represents the properties of an object.

Remarks

Use the **Properties** collection to access the properties displayed in the **Properties** window. For every property listed in the **Properties** window, there is an object in the **Properties** collection.

Reference Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjReferenceC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjReferenceX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjReferenceP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjReferenceM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjReferenceE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjReferenceS"}



Represents a reference to a type library or a project.

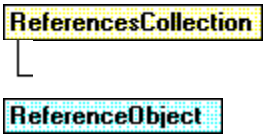
Remarks

Use the **Reference** object to verify whether a reference is still valid.

The **IsBroken** property returns **True** if the reference no longer points to a valid reference. The **BuiltIn** property returns **True** if the reference is a default reference that can't be moved or removed. Use the **Name** property to determine if the reference you want to add or remove is the correct one.

References Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjReferencesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjReferencesX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjReferencesP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjReferencesM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjReferencesE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjReferencesS"}



Represents the set of references in the project.

Remarks

Use the **References** collection to add or remove references. The **References** collection is the same as the set of references selected in the **References** dialog box.

ReferencesEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjReferencesEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjReferencesEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjReferencesEventsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjReferencesEventsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjReferencesEventsP"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjReferencesEventsS"}

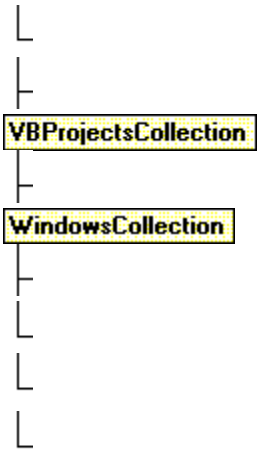
Returned by the **ReferencesEvents** property.

Remarks

The **ReferencesEvents** object is the source of events that occur when a reference is added to or removed from a project. The ItemAdded event is triggered after a reference is added to a project. The ItemRemoved event is triggered after a reference is removed from a project.

VBE Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjVBEC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjVBEX":1}
{ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjVBEP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjVBEM"}
{ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjVBEE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjVBES"}



The root object that contains all other objects and collections represented in Visual Basic for Applications.

Remarks

You can use the following collections to access the objects contained in the **VBE** object:

- Use the **VBProjects** collection to access the collection of projects.
- Use the **Windows** collection to access the collection of windows.
- Use the **CodePanes** collection to access the collection of code panes.
- Use the **CommandBars** collection to access the collection of command bars.

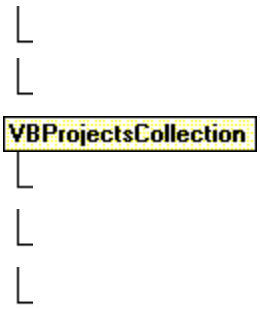
Use the **Events** object to access properties that enable add-ins to connect to all events in Visual Basic for Applications. The properties of the **Events** object return objects of the same type as the property name. For example, the **CommandBarEvents** property returns the **CommandBarEvents** object.

You can use the **SelectedVBComponent** property to return the active component. The active component is the component that is being tracked in the **Project** window. If the selected item in the **Project** window isn't a component, **SelectedVBComponent** returns **Nothing**.

Note All objects in this object model have a **VBE** property that points to the **VBE** object.

VBProject Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjVBProjectC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjVBProjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjVBProjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjVBProjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjVBProjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjVBProjects"}



Represents a project.

Remarks

Use the **VBProject** object to set properties for the project, to access the **VBComponents** collection, and to access the **References** collection.

VBProjects Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjVBProjectsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjVBProjectsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjVBProjectsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjVBProjectsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjVBProjectsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjVBProjectsS"}

Represents all the projects that are open in the development environment.

Remarks

Use the **VBProjects** collection to access specific projects in an instance of the development environment. **VBProjects** is a standard collection that can be used in a **For Each** block.

Window Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjWindowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjWindowX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjWindowP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjWindowM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjWindowE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjWindowS"}

WindowsCollection

Represents a window in the development environment.

Remarks

Use the **Window** object to show, hide, or position windows.

You can use the **Close** method to close a window in the **Windows** collection. The **Close** method affects different types of windows as follows:

Window	Result of using Close method
Code window	Removes the window from the **Windows** collection.
Designer	Removes the window from the **Windows** collection.
Window objects of type linked window frame	Windows become unlinked separate windows.

Note Using the **Close** method with code windows and designers actually closes the window. Setting the **Visible** property to **False** hides the window but doesn't close the window. Using the **Close** method with development environment windows, such as the **Project** window or **Properties** window, is the same as setting the **Visible** property to **False**.

You can use the **SetFocus** method to move the focus to a window.

You can use the **Visible** property to return or set the visibility of a window.

To find out what type of window you are working with, you can use the **Type** property. If you have more than one window of a type, for example, multiple designers, you can use the **Caption** property to determine the window you're working with. You can also find the window you want to work with using the **DesignerWindow** property of the **VBComponent** object or the **Window** property of the **CodePane** object.

Windows Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjWindowsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaobjWindowsX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaobjWindowsP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vaobjWindowsM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vaobjWindowsE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaobjWindowsS"}

WindowsCollection

Contains all open or permanent windows.

Remarks

Use the **Windows** collection to access **Window** objects.

The **Windows** collection has a fixed set of windows that are always available in the collection, such as the **Project** window, the **Properties** window, and a set of windows that represent all open code windows and designer windows. Opening a code or designer window adds a new member to the **Windows** collection. Closing a code or designer window removes a member from the **Windows** collection. Closing a permanent development environment window doesn't remove the corresponding object from this collection, but results in the window not being visible.

ActiveCodePane Property Example

The following example uses the **ActiveCodePane** property and **TopLine** properties to obtain the number of the top line in the active code pane.

```
Debug.Print Application.VBE.ActiveCodePane.TopLine
```

ActiveVBProject Property Example

The following example uses the **ActiveVBProject** property to return the name of the active project.

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

ActiveWindow Property Example

The following example uses the **ActiveWindow** property to return the caption of the active window.

```
Debug.Print Application.VBE.ActiveWindow.Caption
```

BuiltIn Property Example

The following example uses the **BuiltIn** property to return a **Boolean** indicating whether or not a particular reference in the active project is built-in.

```
Debug.Print Application.VBE.ActiveVBProject.References(1).BuiltIn
```

Caption Property Example

The following example uses the **Caption** property to display the caption of the active window.

```
Debug.Print Application.VBE.ActiveWindow.Caption
```

CodeModule Property Example

The following example uses the **CodeModule** and **CountOfLines** properties to return the number of lines in a particular code module.

```
Debug.Print
Application.VBE.ActiveVBProject.VBComponents(6).CodeModule.CountOfLines
```

CodePane Property Example

The following example uses the **CodePane** and **TopLine** properties to display the number of the top line in the code module of the selected **VBComponent** object.

```
Debug.Print Application.VBE.SelectedVBComponent.CodeModule.CodePane.TopLine
```

CodePanes Property Example

The following example uses the **CodePanes** and **TopLine** properties to display the line number of the top line in the specified code pane.

```
Debug.Print Application.VBE.CodePanes(3).TopLine
```

CodePaneView Property Example

The following example uses the **CodePaneView** property to return a value indicating whether the specified code pane is in procedure view or full module view.

```
Debug.Print Application.VBE.CodePanes(3).CodePaneView
```

Collection Property Example

The following example uses the **Collection** and **Count** properties to return the number of objects the active project contains, when viewed as a collection of objects.

```
Debug.Print Application.VBE.ActiveVBProject.Collection.Count
```

CommandBarEvents Property Example

The following example uses code including the **CommandBarEvents** property to support any code to handle a mouse click on a command bar.

```
Private WithEvents ce As CommandBarEvents

Sub Test()
    Dim c As CommandBarControl
    Set c = Application.VBE.CommandBars("Tools").Controls(1)
    Set ce = Application.VBE.Events.CommandBarEvents(c)
End Sub

Private Sub ce_Click(ByVal CommandBarControl As Object, Handled As Boolean,
CancelDefault As Boolean)
    ' Put event-handling code here
End Sub
```

Count Property Example

The following example uses the **Count** property to return the number of **VBComponent** objects in a particular project.

```
Debug.Print Application.VBE.VBProjects(1).VBComponents.Count
```

CountOfDeclarationLines Property Example

The following example uses the **CountOfDeclarationLines** property to return the number of declaration lines in a particular code pane.

```
Debug.Print Application.VBE.CodePanes(2).CodeModule.CountOfDeclarationLines
```

CountOfLines Property Example

The following example uses the **CountOfLines** property to return the total number of lines in a particular code pane.

```
Application.VBE.CodePanes(2).CodeModule.CountOfLines
```

CountOfVisibleLines Property Example

The following example uses the **CountOfVisibleLines** property to return the number of lines visible at one time in a particular code pane, based on the height of the pane.

```
Debug.Print Application.VBE.Codepanes(3).CountOfVisibleLines
```

Description Property Example

The first of the following examples uses the **Description** property to assign a description to a particular project; the example also prints the description to verify that the assignment was successful.

The second example uses the **Description** property to return the descriptive names of the specified **Reference** objects in a particular project.

```
Application.VBE.VBProjects(1).Description = "Hot Sauce"
Debug.Print Application.VBE.VBProjects(1).Description

Debug.Print Application.VBE.VBProjects(1).References(1).Description

Debug.Print Application.VBE.VBProjects(1).References(2).Description
```

Designer Property Example

The following example uses the **Designer** and **Count** properties to return the number of controls on a form. Note that the window containing the form must be selected. The **Designer** object is the form itself.

```
Debug.Print Application.VBE.SelectVBComponent.Designer.Controls.Count
```

DesignerWindow Property Example

The following example uses the **DesignerWindow** and **Visibile** properties to find out whether or not a particular designer is visible. Note that the **VBComponent** object must be a form.

```
Debug.Print
Application.VBE.VBProjects(1).VBComponents(1).DesignerWindow.Visible
```

FullPath Property Example

The following example uses the **FullPath** property to return the full path of the object library for the specified reference.

```
Debug.Print Application.VBE.ActiveVBProject.References(1).FullPath
```

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
```

HasOpenDesigner Property Example

The following example uses the **HasOpenDesigner** property to return whether or not the specified component, in this case a form, of a particular project has an open designer.

```
Debug.Print Application.VBE.VBProjects(1).VBComponents(1).HasOpenDesigner
```

Height, Width Properties Example

The following example uses the **Height** and **Width** properties to return the height and width of the specified window, in twips. These property settings change after a window is linked or docked because then they refer to the **Window** object to which the original window is linked or docked.

```
Debug.Print Application.VBE.Windows(9).Height
Debug.Print Application.VBE.Windows(9).Width
```

HelpContextID Property Example

The following example uses the **HelpContextID** property to return the context ID for the Help file corresponding to a project.

```
Debug.Print Application.VBE.VBProjects(1).HelpContextID
```

HelpFile Property Example

The following example uses the **HelpFile** property to assign a Help file to a project; the example verifies that the assignment was successful by printing the full path of the Help file.

```
Application.VBE.VBProjects(1).HelpFile = "C:\HelpStuff\veenob3.hlp"
Debug.Print Application.VBE.VBProjects(1).HelpFile
```

IsBroken Property Example

The following example uses the **IsBroken** property to return a value indicating whether or not the specified **Reference** object in a particular project is broken.

```
Debug.Print Application.VBE.vbprojects(1).References(1).IsBroken
```

Left, Top Properties Example

The following example uses the **Left** and **Top** properties to return the coordinates of the upper-left corner of a particular window, in twips. These property settings change after a window is linked or docked because then they refer to the **Window** object to which the original window is linked or docked.

```
Debug.Print Application.VBE.Windows(9).Left
Debug.Print Application.VBE.Windows(9).Top
```

MainWindow Property Example

The following example uses the **MainWindow** property to return the **Window** object representing the main window, and then prints the caption of the main window.

```
Debug.Print Application.VBE.MainWindow.Caption
```

Major Property Example

The following example uses the **Major** property to return the major version number of the specified **Reference** object in a particular project.

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

Minor Property Example

The following example uses the **Minor** property to return the minor version number of the specified **Reference** object in a particular project.

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

Mode Property Example

The following example uses the **Mode** property to return the mode of the active project. The value returned is a predefined constant representing the project's mode.

```
Debug.Print Application.VBE.ActiveVBProject.Mode
```

Name Property Example

The following example uses the **Name** property to return the name of the specified member of the **VBComponents** collection in a particular project.

```
Debug.Print Application.VBE.VBProjects(1).VBComponents(1).Name
```

NumIndices Property Example

The following example uses the **NumIndices** property to return the number of indexes belonging to the specified property of a particular **VBComponent** object

```
Debug.Print
Application.VBE.VBProjects(1).VBComponents(1).Properties(40).NumIndices
```

Object Property Example

The following example loads the name of an icon into the icon list for the specified object, which must be a form.

```
Set
Application.VBE.ActiveVBProject.VBComponents(1).Properties("Icon").Object =
LoadPicture("Baseball.ico")
```

Parent Property Example

The following example uses the **Parent** property to return the name of an object's parent in the object hierarchy.

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

Protection Property Example

The following example uses the **Protection** property to return a value indicating whether or not a project is protected. The value returned is a number that corresponds to a predefined constant representing the project's status.

```
Debug.Print Application.VBE.ActiveVBProject.Protection
```

ReferencesEvents Property Example

The following example uses code including the **ReferencesEvents** property to support event-handling code for adding or removing references.

```
Private WithEvents X As ReferencesEvents

Sub Test()
    Set X = Application.VBE.Events.ReferencesEvents
End Sub

Private Sub X_ItemAdded(ByVal Reference As VBIDE.Reference)
    ' Put code to support item addition here
End Sub

Private Sub X_ItemRemoved(ByVal Reference As VBIDE.Reference)
    ' Put code to support item removal here
End Sub
```

Saved Property Example

The following example uses the **Saved** property to return a **Boolean** value indicating whether or not the specified project has been saved in its current state.

```
Debug.Print Application.VBE.VBProjects(1).Saved
```

SelectedVBComponent Property Example

The following example uses the **SelectedVBComponent** property to return the selected component.

```
Debug.Print Application.VBE.SelectedVBComponent.Name
```

TopLine Property Example

The following example uses the **TopLine** property to return the line number of the top line in the specified code pane.

```
Debug.Print Application.VBE.CodePanes(3).TopLine
```

Type Property Example

The following example uses the **Type** property to return a value indicating the type of the specified member of the **VBComponents** collection in a particular project. The value returned is a number that corresponds to a predefined constant for one of the component object types.

```
Debug.Print Application.VBE.VBProjects(1).VBComponents(1).Type
```

Value Property Example

The following example uses the **Value** property to return the value of the specified property of a member of the **VBComponents** collection.

```
Debug.Print Application.VBE.
ActiveVBProject.VBComponents(1).Properties("AcceptLabelsInFormulas").Value
```

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
```

Version Property Example

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

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

Visible Property Example

The following example uses the **Visible** property to return a **Boolean** value indicating whether or not the specified window is visible.

```
Debug.Print Application.VBE.Windows(9).Visible
```

Window Property Example

The following example uses the **Window** and **Caption** properties to return the caption of the specified code pane.

```
Debug.Print Application.VBE.CodePanes(1).Window.Caption
```

WindowState Property Example

The following example uses the **WindowState** property to return the visual state of the specified window. The value returned is a number that corresponds to a predefined constant that specifies the visual state of a window.

```
Debug.Print Application.VBE.Windows(9).WindowState
```

SelectedVBComponent Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproSelectedVBComponentC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproSelectedVBComponentX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproSelectedVBComponentA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproSelectedVBComponentS"}

Returns the selected component. Read-only.

Remarks

The **SelectedVBComponent** property returns the selected component in the **Project** window. If the selected item in the **Project** window isn't a component, **SelectedVBComponent** returns **Nothing**.

ActiveVBProject Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproActiveVBProjectC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproActiveVBProjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproActiveVBProjectA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproActiveVBProjectS"}

Returns the active project in the **Project** window. Read-only.

Remarks

The **ActiveVBProject** property returns the project that is selected in the **Project** window or the project in which the components are selected. In the latter case, the project itself isn't necessarily selected. Whether or not the project is explicitly selected, there is always an active project .

ActiveWindow Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproActiveWindowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproActiveWindowX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproActiveWindowA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproActiveWindowS"}

Returns the active window in the development environment. Read-only.

Remarks

When more than one window is open in the development environment, the **ActiveWindow** property setting is the window with the focus. If the main window has the focus, **ActiveWindow** returns **Nothing**.

Caption Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCaptionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCaptionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCaptionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCaptionS"}

Returns a **String** containing the title of the active window. Read-only.

Remarks

The title of the active window is the text displayed in the window's title bar.

CodeModule Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCodeModuleC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCodeModuleX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCodeModuleA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCodeModuleS"}

Returns an object representing the code behind the component. Read-only.

Remarks

The **CodeModule** property returns **Nothing** if the component doesn't have a code module associated with it.

Note The **CodePane** object represents a visible code window. A given component can have several **CodePane** objects. The **CodeModule** object represents the code within a component. A component can only have one **CodeModule** object.

CodePane Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCodePaneC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCodePaneX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCodePaneA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCodePaneS"}

Returns a **CodePane** object. Read-only.

Remarks

If a code pane exists, it becomes the active code pane, and the window that contains it becomes the active window. If a code pane doesn't exist for the module, the **CodePane** property creates one.

CodePaneView Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCodePaneViewC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCodePaneViewX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCodePaneViewA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCodePaneViewS"}

Returns a value indicating whether the code pane is in Procedure view or Full Module view. Read-only.

Return Values

The **CodePaneView** property return values are:

Constant	Description
vbext_cv_ProcedureView	The specified code pane is in Procedure view.
vbext_cv_FullModuleView	The specified project is in Full Module view.

Collection Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCollectionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCollectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCollectionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCollectionS"}

Returns the collection that contains the object you are working with. Read-only.

Remarks

Most objects in this object model have either a **Parent** property or a **Collection** property that points to the object's parent object.

Use the **Collection** property to access the properties, methods, and controls of the collection to which the object belongs.

CommandBarEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCommandBarEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCommandBarEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCommandBarEventsA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCommandBarEventsS"}

Returns the **CommandBarEvents** object. Read-only.

Settings

The setting for the argument you pass to the **CommandBarEvents** property is:

Argument	Description
vbcontrol	Must be an object of type **CommandBarControl**.

Remarks

Use the **CommandBarEvents** property to return an event source object that triggers an event when a command bar button is clicked. The argument passed to the **CommandBarEvents** property is the command bar control for which the Click event will be triggered.

Count Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproAddinCountC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproAddinCountX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproAddinCountA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproAddinCountS"}

Returns a **Long** containing the number of items in a collection. Read-only.

CountOfLines Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCountOfLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCountOfLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCountOfLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCountOfLinesS"}

Returns a **Long** containing the number of lines of code in a code module. Read-only.

CountOfDeclarationLines Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCountOfDeclarationLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCountOfDeclarationLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCountOfDeclarationLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCountOfDeclarationLinesS"}

Returns a **Long** containing the number of lines of code in the Declarations section of a code module. Read-only.

CountOfVisibleLines Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCountOfVisibleLinesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCountOfVisibleLinesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCountOfVisibleLinesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCountOfVisibleLinesS"}

Returns a **Long** containing the number of lines visible in a code pane. Read-only.

ActiveCodePane Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproActiveCodePaneC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproActiveCodePaneX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproActiveCodePaneA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproActiveCodePaneS"}

Returns the active or last active **CodePane** object or sets the active **CodePane** object. Read/write.

Remarks

You can set the **ActiveCodePane** property to any valid **CodePane** object, as shown in the following example:

```
Set MyApp.VBE.ActiveCodePane = MyApp.VBE.CodePanes(1)
```

The preceding example sets the first code pane in a collection of code panes to be the active code pane. You can also activate a code pane using the **Set** method.

Description Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproAddinDescriptionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproAddinDescriptionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproAddinDescriptionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproAddinDescriptionS"}

Returns or sets a string expression containing a descriptive string associated with an object. For the **VBProject** object, read/write; for the **Reference** object, read-only.

Remarks

For the **VBProject** object, the **Description** property returns or sets a descriptive string associated with the active project.

For the **Reference** object, the **Description** property returns the descriptive name of the reference.

Designer Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproDesignerC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproDesignerX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproDesignerA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproDesignerS"}

Returns the object that enables you to access the design characteristics of a component.

Remarks

If the object has an open designer, the **Designer** property returns the open designer; otherwise a new designer is created. The designer is a characteristic of certain **VBComponent** objects. For example, when you create certain types of **VBComponent** object, a designer is created along with the object. A component can have only one designer, and it's always the same designer. The **Designer** property enables you to access a component-specific object. In some cases, such as in standard modules and class modules, a designer isn't created because that type of **VBComponent** object doesn't support a designer.

The **Designer** property returns **Nothing** if the **VBComponent** object doesn't have a designer.

DesignerWindow Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproDesignerWindowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproDesignerWindowX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproDesignerWindowA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproDesignerWindowS"}

Returns the **Window** object that represents the component's designer.

Remarks

If the component supports a designer but doesn't have an open designer, accessing the **DesignerWindow** property creates the designer, but it isn't visible. To make the window visible, set the **Window** object's **Visible** property to **True**.

CodePanes Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproCodePanesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproCodePanesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproCodePanesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproCodePanesS"}

Returns the collection of active **CodePane** objects. Read-only.

GUID Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproGUIDC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproGUIDX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproGUIDA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproGUIDS"}

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

HasOpenDesigner Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproHasOpenDesignerC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproHasOpenDesignerX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproHasOpenDesignerA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproHasOpenDesignerS"}

Returns a **Boolean** value indicating whether or not the **VBComponent** object has an open designer. Read-only.

Return Values

The **HasOpenDesigner** property returns these values:

Value	Description
True	The **VBComponent** object has an open **Design** window.
False	The **VBComponent** object doesn't have an open **Design** window.

Height Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproHeightC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproHeightX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproHeightS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproHeightA"}

Returns or sets a **Single** containing the height of the window in twips. Read/write.

Remarks

Changing the **Height** property setting of a linked window or docked window has no effect as long as the window remains linked or docked.

HelpContextID Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproAddinHelpContextIDC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproAddinHelpContextIDX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproAddinHelpContextIDA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproAddinHelpContextIDS"}

Returns or sets a **String** containing the context ID for a topic in a Microsoft Windows Help file. Read/write.

HelpFile Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproAddinHelpFileC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproAddinHelpFileX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproAddinHelpFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproAddinHelpFileS"}

Returns or sets a **String** specifying the Microsoft Windows Help file for a project. Read/write.

IndexedValue Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproIndexedValueC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproIndexedValueX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproIndexedValueA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproIndexedValueS"}

Returns or sets a value for a member of a property that is an indexed list or an array.

Remarks

The value returned or set by the **IndexedValue** property is an expression that evaluates to a type that is accepted by the object. For a property that is an indexed list or array, you must use the **IndexedValue** property instead of the **Value** property. An indexed list is a numeric expression specifying index position.

IndexedValue accepts up to 4 indices. The number of indices accepted by **IndexedValue** is the value returned by the **NumIndices** property.

The **IndexedValue** property is used only if the value of the **NumIndices** property is greater than zero. Values in indexed lists are set or returned with a single index.

IsBroken Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproIsBrokenC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproIsBrokenX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproIsBrokenA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproIsBrokenS"}

Returns a **Boolean** value indicating whether or not the **Reference** object points to a valid reference in the registry. Read-only.

Return Values

The **IsBroken** property returns these values:

Value	Description
True	The **Reference** object no longer points to a valid reference in the registry.
False	The **Reference** object points to a valid reference in the registry.

BuiltIn Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproBuiltInC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproBuiltInX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproBuiltInS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproBuiltInA"}

Returns a **Boolean** value indicating whether or not the reference is a default reference that can't be removed. Read-only.

Return Values

The **BuiltIn** property returns these values:

Value	Description
True	The reference is a default reference that can't be removed.
False	The reference isn't a default reference; it can be removed.

Saved Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproSavedC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproSavedX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproSavedS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproSavedA"}

Returns a **Boolean** value indicating whether or not the object was edited since the last time it was saved. Read/write.

Return Values

The **Saved** property returns these values:

Value	Description
True	The object has not been edited since the last time it was saved.
False	The object has been edited since the last time it was saved.

Remarks

The **SaveAs** method sets the **Saved** property to **True**.

Note If you set the **Saved** property to **False** in code, it returns **False**, and the object is marked as if it were edited since the last time it was saved.

Left Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproLeftC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproLeftX":1}
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproLeftA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproLeftS"}

Returns or sets a **Single** containing the location of the left edge of the window on the screen in twips. Read/write.

Remarks

The value returned by the **Left** property depends on whether or not the window is linked or docked.

Note Changing the **Left** property setting of a linked or docked window has no effect as long as the window remains linked or docked.

LinkedWindowFrame Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproLinkedWindowFrameC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproLinkedWindowFrameX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproLinkedWindowFrameA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproLinkedWindowFrameS"}

Returns the **Window** object representing the frame that contains the window. Read-only.

Remarks

The **LinkedWindowFrame** property enables you to access the object representing the linked window frame, which has properties distinct from the window or windows it contains. If the window isn't linked, the **LinkedWindowFrame** property returns **Nothing**.

FullPath Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproFullPathC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproFullPathX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproFullPathA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproFullPathS"}

Returns a **String** containing the path and file name of the referenced type library. Read-only.

MainWindow Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproMainWindowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproMainWindowX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproMainWindowA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproMainWindowS"}

Returns a **Window** object representing the main window of the Visual Basic development environment. Read-only.

Remarks

You can use the **Window** object returned by the **MainWindow** property to add or remove docked windows. You can also use the **Window** object returned by the **MainWindow** property to maximize, minimize, hide, or restore the main window of the Visual Basic development environment.

Major Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproMajorC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproMajorX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproMajorS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproMajorA"}

Returns a **Long** containing the major version number of the referenced type library. Read-only.

Remarks

The number returned by the **Major** property corresponds to the major version number stored in the type library to which you have set the reference.

Minor Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproMinorC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproMinorX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproMinorS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproMinorA"}

Returns a **Long** indicating the minor version number of the referenced type library. Read-only.

Remarks

The number returned by the **Minor** property corresponds to the minor version number stored in the type library to which you have set the reference.

Mode Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproModeC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproModeX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproModeA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproModeS"}

Returns a value containing the mode of the specified project. Read-only.

Return Values

The **Mode** property return values are:

Constant	Description
vbext_vm_Run	The specified project is in run mode.
vbext_vm_Break	The specified project is in break mode.
vbext_vm_Design	The specified project is in design mode.

Name Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproNameC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproNameX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproNameA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproNameS"}

Returns or sets a **String** containing the name used in code to identify an object. For the **VBProject** object and the **VBComponent** object, read/write; for the **Property** object and the **Reference** object, read-only.

Remarks

The following table describes how the **Name** property setting applies to different objects.

Object	Result of Using Name Property Setting
VBProject	Returns or sets the name of the active project.
VBComponent	Returns or sets the name of the component. An error occurs if you try to set the **Name** property to a name already being used or an invalid name.
Property	Returns the name of the property as it appears in the **Property Browser**. This is the value used to index the **Properties** collection. The name can't be set.
Reference	Returns the name of the reference in code. The name can't be set.

The default name for new objects is the type of object plus a unique integer. For example, the first new Form object is Form1, a new Form object is Form1, and the third TextBox control you create on a form is TextBox3.

An object's **Name** property must start with a letter and can be a maximum of 40 characters. It can include numbers and underline (_) characters but can't include punctuation or spaces. Forms and modules can't have the same name as another public object such as **Clipboard**, **Screen**, or **App**. Although the **Name** property setting can be a keyword, property name, or the name of another object, this can create conflicts in your code.

NumIndices Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproNumIndicesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproNumIndicesX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproNumIndicesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproNumIndicesS"}

Returns the number of indices on the property returned by the **Property** object.

Remarks

The value of the **NumIndices** property can be an integer from 0 – 4. For most properties, **NumIndices** returns 0. Conventionally indexed properties return 1. Property arrays might return 2.

Object Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproObjectC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproObjectS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproObjectA"}

Returns or sets the value of an object returned by a property. Read/write.

Remarks

If a property returns an object, you must use the **Object** property to return or set the value of that object.

Parent Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproParentC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproParentX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproParentS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproParentA"}

Returns the object or collection that contains another object or collection. Read-only.

Remarks

Most objects have either a **Parent** property or a **Collection** property that points to the object's parent object in this object model. The **Collection** property is used if the parent object is a collection.

Use the **Parent** property to access the properties, methods, and controls of an object's parent object.

Protection Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproProtectionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproProtectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproProtectionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproProtectionS"}

Returns a value indicating the state of protection of a project. Read-only.

Return Values

The **Protection** property return values are:

Constant	Description
vbext_pp_locked	The specified project is locked.
vbext_pp_none	The specified project isn't protected.

ReferencesEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproReferencesEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproReferencesEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproReferencesEventsA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproReferencesEventsS"}

Returns the **ReferencesEvents** object. Read-only.

Settings

The setting for the argument you pass to the **ReferencesEvents** property is:

Argument	Description
vbproject	If *vbproject* points to **Nothing**, the object that is returned will supply events for the **References** collections of all **VBProject** objects in the **VBProjects** collection. If *vbproject* points to a valid **VBProject** object, the object that is returned will supply events for only the **References** collection for that project.

Remarks

The **ReferencesEvents** property takes an argument and returns an event source object. The **ReferencesEvents** object is the source for events that are triggered when references are added or removed.

Top Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproTopC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproTopX":1}
{ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproTopA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproTopS"}

Returns or sets a **Single** specifying the location of the top of the window on the screen in twips. Read/write.

Remarks

The value returned by the **Top** property depends on whether or not the window is docked, linked, or in docking view.

Note Changing the **Top** property setting of a linked or docked window has no effect as long as the window remains linked or docked.

TopLine Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproTopLineC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproTopLineX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproTopLineA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproTopLineS"}

Returns a **Long** specifying the line number of the line at the top of the code pane or sets the line showing at the top of the code pane. Read/write.

Remarks

Use the **TopLine** property to return or set the line showing at the top of the code pane. For example, if you want line 25 to be the first line showing in a code pane, set the **TopLine** property to 25.

The **TopLine** property setting must be a positive number. If the **TopLine** property setting is greater than the actual number of lines in the code pane, the setting will be the last line in the code pane.

Type Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproTypeC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproTypeX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproTypeS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproTypeA"}

Returns a numeric or string value containing the type of object. Read-only.

Return Values

The **Type** property settings for the **Window** object are described in the following table:

Constant	Value	Description
vbext_wt_CodeWindow	0	**Code** window
vbext_wt_Designer	1	Designer
vbext_wt_Browser	2	**Object Browser**
vbext_wt_Watch	3	Watch pane
vbext_wt_Locals	4	Locals
vbext_wt_Immediate	5	**Immediate** window
vbext_wt_ProjectWindow	6	**Project** window
vbext_wt_PropertyWindow	7	**Properties** window
vbext_wt_Find	8	**Find** dialog box
vbext_wt_FindReplace	9	**Search and Replace** dialog box
vbext_wt_LinkedWindowFrame	11	Linked window frame
vbext_wt_MainWindow	12	Main window

The **Type** property settings for the **VBComponent** object are described in the following table:

Constant	Description
vbext_ct_ClassModule	Class module
vbext_ct_MSForm	Microsoft Form
vbext_ct_StdModule	Standard module
vbext_ct_Document	Document module

The **Type** property settings for the **Reference** object are described in the following table:

Constant	Description
vbext_rk_TypeLib	Type library
vbext_rk_Project	Project

Value Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproValueC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproValueX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproValueA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproValueS"}

Returns or sets a **Variant** specifying the value of the property. Read/write.

Remarks

Because the **Value** property returns a **Variant**, you can access any property. To access a list, use the **IndexedValue** property.

If the property that the **Property** object represents is read/write, the **Value** property is read/write. If the property is read-only, attempting to set the **Value** property causes an error. If the property is write-only, attempting to return the **Value** property causes an error.

The **Value** property is the default property for the **Property** object.

VBE Property

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.

Version Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproVersionC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproVersionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproVersionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproVersionS"}

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.

Visible Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproVisibleC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproVisibleX":1} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproVisibleS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproVisibleA"}

For the **Window** object, returns or sets a **Boolean** value that specifies the visibility of a window. Read/write. For the **CodePane** object, returns a **Boolean** value that indicates whether or not the code pane is visible in the window. Read-only.

Return Values

The **Visible** property returns the following values:

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

Width Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproWidthC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproWidthX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproWidthA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproWidthS"}

Returns or sets a **Single** containing the width of the window in twips. Read/write.

Remarks

Changing the **Width** property setting of a linked window or docked window has no effect as long as the window remains linked or docked.

Window Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproWindowC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproWindowX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproWindowA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproWindowS"}

Returns the window in which the code pane is displayed. Read-only.

WindowState Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaproWindowStateC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vaproWindowStateX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaproWindowStateA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vaproWindowStateS"}

Returns or sets a numeric value specifying the visual state of the window. Read/write.

Settings

The **WindowState** property returns or sets the following values:

Constant	Value	Description
vbext_ws_Normal	0	(Default) Normal
vbext_ws_Minimize	1	Minimized (minimized to an icon)
vbext_ws_Maximize	2	Maximized (enlarged to maximum size)

Change Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtChangeEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtChangeEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtChangeEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtChangeEventControlsPlaceholderS"}

Click Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtClickEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtClickEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtClickEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtClickEventControlsPlaceholderS"}

DblClick Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtDblClickEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtDblClickEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtDblClickEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtDblClickEventControlsPlaceholderS"}

DragDrop Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtDragDropEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtDragDropEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtDragDropEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtDragDropEventControlsPlaceholderS"}

DragOver Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtDragOverEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtDragOverEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtDragOverEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtDragOverEventControlsPlaceholderS"}

KeyDown Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtKeyDownEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtKeyDownEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtKeyDownEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtKeyDownEventControlsPlaceholderS"}

KeyPress Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtKeyPressEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtKeyPressEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtKeyPressEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtKeyPressEventControlsPlaceholderS"}

KeyUp Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtKeyUpEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtKeyUpEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtKeyUpEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtKeyUpEventControlsPlaceholderS"}

MouseDown Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtMouseDownEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtMouseDownEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtMouseDownEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtMouseDownEventControlsPlaceholderS"}

MouseMove Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtMouseMoveEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtMouseMoveEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtMouseMoveEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtMouseMoveEventControlsPlaceholderS"}

MouseUp Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtMouseUpEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtMouseUpEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtMouseUpEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtMouseUpEventControlsPlaceholderS"}

OLECompleteDragEvent (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLECompleteDragEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLECompleteDragEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLECompleteDragEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLECompleteDragEventControlsPlaceholderS"}

OLEDragDrop Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLEDragDropEventPHolderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLEDragDropEventPHolderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLEDragDropEventPHolderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLEDragDropEventPHolderS"}

OLEDragOver Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLEDragOverEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLEDragOverEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLEDragOverEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLEDragOverEventControlsPlaceholderS"}

OLEGiveFeedback Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLEGiveFeedbackEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLEGiveFeedbackEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLEGiveFeedbackEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLEGiveFeedbackEventControlsPlaceholderS"}

OLESetData Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLESetDataEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLESetDataEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLESetDataEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLESetDataEventControlsPlaceholderS"}

OLEStartDrag Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtOLEStartDragEventControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtOLEStartDragEventControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtOLEStartDragEventControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtOLEStartDragEventControlsPlaceholderS"}

Refresh Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRefreshMethodControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRefreshMethodControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRefreshMethodControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRefreshMethodControlsPlaceholderS"}

OLEDrag Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthOLEDragMethodControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthOLEDragMethodControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthOLEDragMethodControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthOLEDragMethodControlsPlaceholderS"}

Appearance Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproAppearancePropertyControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproAppearancePropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproAppearancePropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproAppearancePropertyControlsPlaceholderS"}

BackColor Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBackColorPropertyControlsPlaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproBackColorPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproBackColorPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproBackColorPropertyControlsPlaceholderS"}

Caption Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCaptionPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproCaptionPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproCaptionPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCaptionPropertyControlsPlaceholderS"}

Enabled Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproEnabledPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproEnabledPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproEnabledPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproEnabledPropertyControlsPlaceholderS"}

Font Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFontPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFontPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFontPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFontPropertyControlsPlaceholderS"}

ForeColor Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproForeColorPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproForeColorPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproForeColorPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproForeColorPropertyControlsPlaceholderS"}

IntegralHeight Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIntegralHeightPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIntegralHeightPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproIntegralHeightPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIntegralHeightPropertyControlsPlaceholderS"}

Locked Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproLockedPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproLockedPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproLockedPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproLockedPropertyControlsPlaceholderS"}

MouseIcon Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproMouseIconPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMouseIconPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMouseIconPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMouseIconPropertyControlsPlaceholderS"}

MousePointer Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproMousePointerPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMousePointerPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMousePointerPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMousePointerPropertyControlsPlaceholderS"}

OLEDragMode Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproOLEDragModePropertyControlsPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproOLEDragModePropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproOLEDragModePropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproOLEDragModePropertyControlsPlaceholderS"}

OLEDropMode Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproOLEDropModePropertyControlsPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproOLEDropModePropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproOLEDropModePropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproOLEDropModePropertyControlsPlaceholderS"}

SelLength Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproSelLengthPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproSelLengthPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproSelLengthPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproSelLengthPropertyControlsPlaceholderS"}

SelStart Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproSelStartPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproSelStartPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproSelStartPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproSelStartPropertyControlsPlaceholderS"}

SelText Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproSelTextPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproSelTextPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproSelTextPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproSelTextPropertyControlsPlaceholderS"}

Style Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproStylePropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproStylePropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproStylePropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproStylePropertyControlsPlaceholderS"}

Text Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTextPropertyControlsPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproTextPropertyControlsPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproTextPropertyControlsPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTextPropertyControlsPlaceholderS"}

Keyword Not Found

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic"}

The keyword you've selected can't be found in Visual Basic Help. It's possible that:

- The keyword is misspelled
- You selected too much or too little text
- You asked for help on a word that is not a valid Visual Basic keyword.

The easiest way to get help on a specific keyword is to position the insertion point anywhere within the keyword you want help on and press F1. You do not need to select the keyword. In fact, if you select only a portion of the keyword, or more than a single word, Help will not find what you're looking for.

To use the built-in Help search dialog box, press the **Help Topics** button on the toolbar.

One option to possibly find what you're looking for is to view the ReadMe file that comes with Visual Basic. This document contains information regarding last-minute changes, additions, and deletions that did not make it into the final documentation.

Left, Top Properties (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproLeftTopPropertiesPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproLeftTopPropertiesPlaceholderS"}

Height, Width Properties (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHeightWidthPropertiesPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHeightWidthPropertiesPlaceholderS"}

Tag Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTagPropertyPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTagPropertyPlaceholderS"}

ToolTipText Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproToolTipTextPropertyPlaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproToolTipTextPropertyPlaceholderS"}

Count Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCountPropertyPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCountPropertyPlaceholderS"}

Item Method (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproItemMethodPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthItemMethodPlaceholderS;vbproItemMethodPlaceholderS"}

Hwnd Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHwndPropertyPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHwndPropertyPlaceholderS"}

TabStop Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTabStopPropertyPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproTabStopPropertyPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproTabStopPropertyPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTabStopPropertyPlaceholderS"}

Visible Property (Placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproVisiblePropertyPlaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVisiblePropertyPlaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproVisiblePropertyPlaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVisiblePropertyPlaceholderS"}

Picture Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproPicturePropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproPicturePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproPicturePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproPicturePropertyplaceholderS"}

Max, Min Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproMaxMinPropertiesplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMaxMinPropertiesplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMaxMinPropertiesplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMaxMinPropertiesplaceholderS"}

Add Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthAddMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddMethodplaceholderS"}

Clear Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthClearMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthClearMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthClearMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthClearMethodplaceholderS"}

Container Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproContainerPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproContainerPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproContainerPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproContainerPropertyplaceholderS"}

Controls Collection (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolControlsCollectionplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolControlsCollectionplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbcolControlsCollectionplaceholderP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbcolControlsCollectionplaceholderM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbcolControlsCollectionplaceholderE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolControlsCollectionplaceholderS"}

Copies Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCopiesPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproCopiesPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproCopiesPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCopiesPropertyplaceholderS"}

DataBinding Object (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjDataBindingObjectplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjDataBindingObjectplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjDataBindingObjectplaceholderP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjDataBindingObjectplaceholderM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjDataBindingObjectplaceholderE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjDataBindingObjectplaceholderS"}

DataBindings Collection (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolDataBindingsCollectionplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolDataBindingsCollectionplaceholderX":1} {ewc
HLP95EN.DLL,DYNALINK,"Properties":"vbcolDataBindingsCollectionplaceholderP"} {ewc
HLP95EN.DLL,DYNALINK,"Methods":"vbcolDataBindingsCollectionplaceholderM"} {ewc
HLP95EN.DLL,DYNALINK,"Events":"vbcolDataBindingsCollectionplaceholderE"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbcolDataBindingsCollectionplaceholderS"}

Filename Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFilenamePropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFilenamePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFilenamePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFilenamePropertyplaceholderS"}

FontBold, FontItalic, FontStrikeThru Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFontBoldFontItalicFontStrikeThruPropertiesplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFontBoldFontItalicFontStrikeThruPropertiesplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFontBoldFontItalicFontStrikeThruPropertiesplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFontBoldFontItalicFontStrikeThruPropertiesplaceholderS"}

FontName Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFontNamePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFontNamePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFontNamePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFontNamePropertyplaceholderS"}

FontSize Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFontSizePropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFontSizePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFontSizePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFontSizePropertyplaceholderS"}

GetData Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthGetDataMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthGetDataMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthGetDataMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthGetDataMethodplaceholderS"}

GetFormat Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthGetFormatMethodplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthGetFormatMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthGetFormatMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthGetFormatMethodplaceholderS"}

HelpFile Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHelpFilePropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHelpFilePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHelpFilePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHelpFilePropertyplaceholderS"}

Item Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproItemPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproItemPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproItemPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproItemPropertyplaceholderS"}

Remove Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRemoveMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRemoveMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRemoveMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRemoveMethodplaceholderS"}

SetData Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthSetDataMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthSetDataMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthSetDataMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthSetDataMethodplaceholderS"}

Align Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproAlignPropertyplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproAlignPropertyplaceholderS"}

DragIcon Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDragIconPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDragIconPropertyplaceholderS"}

DragMode Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDragModePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDragModePropertyplaceholderS"}

HelpContextID Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHelpContextIDPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHelpContextIDPropertyplaceholderS"}

TabIndex Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTabIndexPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTabIndexPropertyplaceholderS"}

Alignment Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstAlignmentConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Border Property Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstBorderPropertyConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

BorderStyle Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstBorderStyleConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

BorderStyle Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproBorderStylePropertyActiveXControlsplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproBorderStylePropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproBorderStylePropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproBorderStylePropertyActiveXControlsplaceholderS"}

Clear Method (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthClearMethodActiveXControlsplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthClearMethodActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthClearMethodActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthClearMethodActiveXControlsplaceholderS"}

Clipboard Object Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstClipboardObjectConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Color Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstColorConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

CommonDialog Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstCommonDialogControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

CommonDialog Error Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstCommonDialogErrorConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

DataBindings Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDataBindingsPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDataBindingsPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDataBindingsPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDataBindingsPropertyplaceholderS"}

DDE Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstDDEConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Drag-and-Drop Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstDragandDropConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Drawing Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstDrawingConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

FetchVerbs Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthFetchVerbsMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthFetchVerbsMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthFetchVerbsMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthFetchVerbsMethodS"}

Form Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstFormConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Graphics Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstGraphicsConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Grid Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstGridControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Help Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstHelpConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

HideSelection Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHideSelectionPropertyActiveXControlsplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHideSelectionPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHideSelectionPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHideSelectionPropertyActiveXControlsplaceholderS"}

HideSelection Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHideSelectionPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHideSelectionPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHideSelectionPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHideSelectionPropertyplaceholderS"}

Image Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproImagePropertyActiveXControlsplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproImagePropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproImagePropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproImagePropertyActiveXControlsplaceholderS"}

ImageList Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproImageListPropertyActiveXControlsplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproImageListPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproImageListPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproImageListPropertyActiveXControlsplaceholderS"}

Index Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIndexPropertyActiveXControlsplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIndexPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproIndexPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIndexPropertyActiveXControlsplaceholderS"}

Index Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIndexPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIndexPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproIndexPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIndexPropertyplaceholderS"}

Key Code Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstKeyCodeConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Key Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproKeyPropertyActiveXControlsplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproKeyPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproKeyPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproKeyPropertyActiveXControlsplaceholderS"}

Menu Accelerator Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstMenuAcceleratorConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Menu Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstMenuControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Miscellaneous Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstMiscellaneousConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

MousePointer Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstMousePointerConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

OLE Container Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstOLEContainerControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Picture Object Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstPictureObjectConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Printer Object Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstPrinterObjectConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

RasterOp Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstRasterOpConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Remove Method (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRemoveMethodActiveXControlsplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRemoveMethodActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthRemoveMethodActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRemoveMethodActiveXControlsplaceholderS"}

ShowInTaskbar Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstShowInTaskbarPropertyplaceholderC;vbproBooksOnlineJumpTopic"}

ShowTips Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproShowTipsPropertyActiveXControlsplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproShowTipsPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproShowTipsPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproShowTipsPropertyActiveXControlsplaceholderS"}

ShowWhatsThis Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthShowWhatsThisMethodplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthShowWhatsThisMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthShowWhatsThisMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthShowWhatsThisMethodplaceholderS"}

Text Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTextPropertyActiveXControlsplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproTextPropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproTextPropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTextPropertyActiveXControlsplaceholderS"}

Value Property (ActiveX Controls) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproValuePropertyActiveXControlsplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproValuePropertyActiveXControlsplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproValuePropertyActiveXControlsplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproValuePropertyActiveXControlsplaceholderS"}

Variant Type Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstVariantTypeConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

Visual Basic Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstVisualBasicConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

WhatsThisButton Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproWhatsThisButtonPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproWhatsThisButtonPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproWhatsThisButtonPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproWhatsThisButtonPropertyplaceholderS"}

WhatsThisHelp Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproWhatsThisHelpPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproWhatsThisHelpPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproWhatsThisHelpPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproWhatsThisHelpPropertyplaceholderS"}

WhatsThisHelpID Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproWhatsThisHelpIDPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproWhatsThisHelpIDPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproWhatsThisHelpIDPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproWhatsThisHelpIDPropertyplaceholderS"}

WhatsThisMode Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthWhatsThisModeMethodplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthWhatsThisModeMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthWhatsThisModeMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthWhatsThisModeMethodplaceholderS"}

Windows 95 Control Constants (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcstWindows95ControlConstantsplaceholderC;vbproBooksOnlineJumpTopic"}

ColIsVisible Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproColIsVisiblePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproColIsVisiblePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproColIsVisiblePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproColIsVisiblePropertyplaceholderS"}

ColPos Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproColPosPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproColPosPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproColPosPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproColPosPropertyplaceholderS"}

DataObject Object (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDataObjectObjectplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDataObjectObjectplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDataObjectObjectplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjDataObjectObjectplaceholderS;vbproDataObjectObjectplaceholderS"}

DataObjectFiles Collection (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolDataObjectFilesCollectionplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolDataObjectFilesCollectionplaceholderX":1} {ewc
HLP95EN.DLL,DYNALINK,"Properties":"vbcolDataObjectFilesCollectionplaceholderP"} {ewc
HLP95EN.DLL,DYNALINK,"Methods":"vbcolDataObjectFilesCollectionplaceholderM"} {ewc
HLP95EN.DLL,DYNALINK,"Events":"vbcolDataObjectFilesCollectionplaceholderE"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbcolDataObjectFilesCollectionplaceholderS"}

DataSource Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDataSourcePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDataSourcePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDataSourcePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDataSourcePropertyplaceholderS"}

Drag Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthDragMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthDragMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthDragMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthDragMethodplaceholderS"}

DrawMode Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDrawModePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDrawModePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproDrawModePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDrawModePropertyplaceholderS"}

FixedAlignment Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFixedAlignmentPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFixedAlignmentPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproFixedAlignmentPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFixedAlignmentPropertyplaceholderS"}

GotFocus Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtGotFocusEventplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtGotFocusEventplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtGotFocusEventplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtGotFocusEventplaceholderS"}

GridLineWidth Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproGridLineWidthPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproGridLineWidthPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproGridLineWidthPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproGridLineWidthPropertyplaceholderS"}

Index Property (Control Array) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIndexPropertyControlArrayplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIndexPropertyControlArrayplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproIndexPropertyControlArrayplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIndexPropertyControlArrayplaceholderS"}

LostFocus Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtLostFocusEventplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtLostFocusEventplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtLostFocusEventplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtLostFocusEventplaceholderS"}

Move Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthMoveMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthMoveMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthMoveMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthMoveMethodplaceholderS"}

Name Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproNamePropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproNamePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproNamePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproNamePropertyplaceholderS"}

Object Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproObjectPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproObjectPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproObjectPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproObjectPropertyplaceholderS"}

Parent Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproParentPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproParentPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproParentPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproParentPropertyplaceholderS"}

RowColChange Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtRowColChangeEventplaceholderC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtRowColChangeEventplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtRowColChangeEventplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtRowColChangeEventplaceholderS"}

RowIsVisible Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproRowIsVisiblePropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproRowIsVisiblePropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproRowIsVisiblePropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRowIsVisiblePropertyplaceholderS"}

RowPos Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproRowPosPropertyplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproRowPosPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproRowPosPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRowPosPropertyplaceholderS"}

SetFocus Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthSetFocusMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthSetFocusMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthSetFocusMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthSetFocusMethodplaceholderS"}

ZOrder Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthZOrderMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthZOrderMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthZOrderMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthZOrderMethodplaceholderS"}

Files Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthFilesMethodplaceholderC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthFilesMethodplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthFilesMethodplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthFilesMethodplaceholderS"}

Connect Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproConnectEventplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproConnectEventplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproConnectEventplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproConnectEventplaceholderS"}

HScrollSmallChange, VScrollSmallChange Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproHScrollSmallChangeVScrollSmallChangePropertiesplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproHScrollSmallChangeVScrollSmallChangePropertiesplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproHScrollSmallChangeVScrollSmallChangePropertiesplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproHScrollSmallChangeVScrollSmallChangePropertiesplaceholderS"}

MinHeight, MinWidth Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproMinHeightMinWidthPropertiesplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMinHeightMinWidthPropertiesplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMinHeightMinWidthPropertiesplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMinHeightMinWidthPropertiesplaceholderS"}

Property Pages Dialog Box (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproPropertryPagesDialogBoxplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproPropertryPagesDialogBoxplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproPropertryPagesDialogBoxplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproPropertryPagesDialogBoxplaceholderS"}

RemoteHost Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproRemoteHostPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproRemoteHostPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproRemoteHostPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRemoteHostPropertyplaceholderS"}

RemotePort Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproRemotePortPropertyplaceholderC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproRemotePortPropertyplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproRemotePortPropertyplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproRemotePortPropertyplaceholderS"}

ViewportHeight, ViewportLeft, ViewportTop, ViewportWidth Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproViewportHeightViewportLeftViewportTopViewportWidthPropertiesplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproViewportHeightViewportLeftViewportTopViewportWidthPropertiesplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproViewportHeightViewportLeftViewportTopViewportWidthPropertiesplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproViewportHeightViewportLeftViewportTopViewportWidthPropertiesplaceholderS"}

Resync Method (Remote Data) (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthResyncMethodRemoteDataplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthResyncMethodRemoteDataplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthResyncMethodRemoteDataplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthResyncMethodRemoteDataplaceholderS"}

ContainedVBControls Collection (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolContainedVBControlsCollectionplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolContainedVBControlsCollectionplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbcolContainedVBControlsCollectionplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolContainedVBControlsCollectionplaceholderS"}

Member Object (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproMemberObjectplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMemberObjectplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMemberObjectplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMemberObjectplaceholderS"}

Members Collection (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproMembersCollectionplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproMembersCollectionplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproMembersCollectionplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproMembersCollectionplaceholderS"}

VBComponentsEvents Object (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproVBComponentsEventsObjectplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVBComponentsEventsObjectplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproVBComponentsEventsObjectplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVBComponentsEventsObjectplaceholderS"}

VBProjectsEvents Object (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproVBProjectsEventsObjectplaceholderC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVBProjectsEventsObjectplaceholderX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproVBProjectsEventsObjectplaceholderA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVBProjectsEventsObjectplaceholderS"}

Files Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscFilesPropertyplaceholderC"}

Alignment Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscAlignmentPropertyplaceholderC"}

AboutBox Method (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscAboutBoxMethodplaceholderC"}

Col, Row Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscColRowPropertiesplaceholderC"}

RowHeight Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscRowHeightPropertyplaceholderC"}

Scroll Event (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscScrollEventplaceholderC"}

ScrollBars Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscScrollBarsPropertyplaceholderC"}

SelEndCol, SelStartCol, SelEndRow, SelStartRow Properties (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscSelEndColPropertiesplaceholderC"}

ScrollBars Property (placeholder)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmscScrollBarsPropertyplaceholderC"}

AddIn Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjAddInObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjAddInObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproCollection;vaproDescription;vaproGUID;vaproVBE;vbobjAddInObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjAddInObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjAddInObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjAddInObjectS"}

The **AddIn** object provides information about an add-in to other add-ins.

Syntax

AddIn

Remarks

An **AddIn** object is created for every add-in that appears in the Vbaddin.Ini file.

AddIns Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolAddInsCollectionC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolAddInsCollectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproVBE;vbcolAddInsCollectionP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbcolAddInsCollectionM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbcolAddInsCollectionE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolAddInsCollectionS"}

Returns a collection of add-ins listed in the Vbaddin.Ini file.

Syntax

AddIns

Remarks

The **AddIns** collection is accessed through the **VBE** object. Every add-in listed in the Add-In Manager in an instance of Visual Basic has an object in the **AddIns** collection.

This collection replaces the **ExternalObjects** collection used in Visual Basic version 4.0.

CommandBar Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjCommandBarObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjCommandBarObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjCommandBarObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjCommandBarObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjCommandBarObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjCommandBarObjectS"}

The **CommandBar** object contains other **CommandBar** objects which can act as either buttons or menu commands.

Syntax

CommandBar

FileControlEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjFileControlEventsObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjFileControlEventsObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjFileControlEventsObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjFileControlEventsObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjFileControlEventsObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjFileControlEventsObjectS"}

Represents all events supplied by Visual Basic which support file control.

Syntax

FileControlEvents

Remarks

The **FileControlEvents** object replaces the **FileControl** object in Visual Basic version 4.0. It works the same as before, only its events have been changed to allow multiple project support.

Member Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vaobjCodeModule;vaobjCodePane;vbobjMemberObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjMemberObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproCollection;vaproDescription;vaproVBE;vbobjMemberObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjMemberObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjMemberObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjMemberObjectS"}

The **Member** object represents a mixture of code-based properties and type library-based attributes of members.

Syntax

Member

Remarks

Code-based properties like **Name** are read-only, so the add-in must modify the code to change these properties.

Members Collection

{ewc HLP95EN.DLL,DYNALINK,"See
Also":"vaobjCodeModule;vaobjCodePane;vbcolMembersCollectionC;vbproBooksOnlineJumpTopic"} {ewc
HLP95EN.DLL,DYNALINK,"Example":"vbcolMembersCollectionX":1} {ewc
HLP95EN.DLL,DYNALINK,"Properties":"vaproVBE;vbcolMembersCollectionP"} {ewc
HLP95EN.DLL,DYNALINK,"Methods":"vbcolMembersCollectionM"} {ewc
HLP95EN.DLL,DYNALINK,"Events":"vbcolMembersCollectionE"} {ewc
HLP95EN.DLL,DYNALINK,"Specifics":"vbcolMembersCollectionS"}

Returns a collection of code module-level members.

Syntax

Members

Remarks

A member of a code module is an identifier that has module-level scope and which can be considered a property, method, or event of that code module.

Properties Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolPropertiesCollectionC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolPropertiesCollectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproVBE;vbcolPropertiesCollectionP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbcolPropertiesCollectionM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbcolPropertiesCollectionE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolPropertiesCollectionS"}

Returns the available properties of a control or component.

Syntax

Properties

Remarks

This object or collection includes all the properties that can normally be accessed at design time.

The default value for the **Properties** collection is determined by the **Item** method.

Use the **Properties** collection to access the properties displayed in the **Properties** window. For every property listed in the **Properties** window, there is an object in the **Properties** collection.

SelectedVBControls Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolSelectedVBControlsCollectionC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolSelectedVBControlsCollectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproVBE;vbcolSelectedVBControlsCollectionP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vamthClear;vamthcut;vbcolSelectedVBControlsCollectionM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbcolSelectedVBControlsCollectionE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolSelectedVBControlsCollectionS"}

Returns a collection of currently selected controls on a component.

Syntax

SelectedVBControls

Remarks

You can use this collection to access all currently selected controls on a form. The code can step through the collection of controls or request a specific control.

This collection has the same specifications as the **VBControls** collection, except this collection doesn't implement the **Add** method. The default method for the **SelectedVBControls** collection is the **Item** method and is indexed with integers.

This collection replaces the **SelectedControlTemplates** collection from Visual Basic version 4.0.

SelectedVBControlsEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjSelectedVBControlsEventsObjectC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjSelectedVBControlsEventsObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjSelectedVBControlsEventsObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjSelectedVBControlsEventsObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjSelectedVBControlsEventsObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjSelectedVBControlsEventsObjectS"}

Represents a source of events supported by all currently selected controls.

Syntax

SelectedVBControlsEvents

VBComponentsEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjVBComponentsEventsObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjVBComponentsEventsObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjVBComponentsEventsObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjVBComponentsEventsObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjVBComponentsEventsObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjVBComponentsEventsObjectS"}

Represents a source of events that occur when an object is added, removed, selected, renamed, or activated in a Visual Basic project.

Syntax

VBComponentsEvents

VBControl Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjVBControlObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjVBControlObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproCollection;vaprocontainedvbcontrols;vaproVBE;vbobjVBControlObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjVBControlObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjVBControlObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjVBControlObjectS"}

Represents a control on a component in a project.

Syntax

VBControl

Remarks

A program can access a control through the **VBForm** object. Using the **VBForm** object, you can:

- Access all the design time properties of a control.
- Identify the container of the control.
- Change the Z-order of the control.

The **VBControl** object replaces the **ControlTemplate** object from Visual Basic version 4.0.

VBControls Collection

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbcolVBControlsCollectionC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbcolVBControlsCollectionX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaproVBE;vbcolVBControlsCollectionP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbcolVBControlsCollectionM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbcolVBControlsCollectionE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbcolVBControlsCollectionS"}

Returns a collection all components on a form.

Syntax

VBControls

Remarks

A program can access controls through the **VBControls** collection. Using the **VBControls** collection, you can:

- Access all the controls on a component.
- Step through the collection of controls.
- Return a specific control.
- Add controls to a component.

The **Item** method determines the default value of the **VBControls** collection.

The **VBControls** collection replaces the **ControlTemplates** collection from Visual Basic version 4.0.

VBControlsEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjVBControlsEventsObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjVBControlsEventsObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjVBControlsEventsObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjVBControlsEventsObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjVBControlsEventsObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjVBControlsEventsObjectS"}

Represents a source of events that occur when a control is added, removed, selected, renamed, or activated in a Visual Basic project.

Syntax

VBControlsEvents

VBForm Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjVBFormObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjVBFormObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vaprovbcontrols;vaproVBE;vbobjVBFormObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjVBFormObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjVBFormObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjVBFormObjectS"}

Returns a component in a project.

Syntax

VBForm

Remarks

The **ClassName** property determines the default value of the **VBForm** object.

The **VBForm** object replaces the **ControlTemplate** object from Visual Basic version 4.0.

VBProjectsEvents Object

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjVBProjectsEventsObjectC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjVBProjectsEventsObjectX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjVBProjectsEventsObjectP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjVBProjectsEventsObjectM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjVBProjectsEventsObjectE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjVBProjectsEventsObjectS"}

Represents a source of events that occur when projects are added, removed, renamed, or activated in a Visual Basic project.

Syntax

VBProjectsEvents

AfterAddFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtAfterAddFileC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtAfterAddfileX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtAfterAddFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtAfterAddfileS"}

Occurs after a component is added to the current Visual Basic project with the **Add File** command in the **Project** menu.

Syntax

Sub *object*_**AfterAddFile(***vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *filename* **As String)**

The AfterAddFile event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project in which the file was added.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was added, as listed in Settings.
filename	A string expression specifying the name of the file that was added.

Settings

The enumerated values for **vbext_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.

Remarks

Visual Basic triggers this event only for files you can add from the **Project** menu. (That is, forms, classes, **User** controls, **Property Pages**, and modules). The AfterAddFile event does not occur if you select **Add** *object* from the **Project** menu. It also does not occur when an .Frx file is created for the first time, and doesn't occur twice when a form is added.

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.

- Backup the file.

AfterChangeFileName Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtAfterChangeFileNameEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtAfterChangeFileNameEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtAfterChangeFileNameEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtAfterChangeFileNameEventS"}

Occurs after a file in the current project has been saved for the first time, or saved with a new name. It also occurs when the project is first compiled to an .Exe file, or when compiled to a new .Exe name.

Syntax

Sub *object*_**AfterChangeFileName (***vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *newname* **As String**, *oldname* **As String)**

The AfterChangeFileName event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project in which the file was changed.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was changed, as listed in Settings.
newname	A string expression specifying the new name of the file.
oldname	A string expression specifying the old name of the file.

Settings

The enumerated values for **vbext_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.

AfterCloseFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtAfterCloseFileC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtAfterCloseFileX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtAfterCloseFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtAfterCloseFileS"}

Occurs after a project has been closed, either directly by the user, or by Visual Basic when the user quits the program.

Syntax

Sub *object*_**AfterCloseFile(***vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *filename* **As String,** *wasdirty* **As Boolean)**

The AfterCloseFile event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project in which the file was closed.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was closed, as listed in Settings.
filename	A string expression specifying the name of the file that was closed.
wasdirty	A Boolean expression that specifies whether changes were saved to a file prior to it being closed, as listed in Settings.

Settings

The enumerated values for **vbext_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 *wasdirty* are:

Setting	Description
True	The file was dirty when it was closed. (That is, the user elected to not save changes made to the file prior to closing it.)
False	The file was not dirty when it was closed (That is, the user selected to save changes made to the file prior to closing it.)

Remarks

This event can occur once for each add-in connected to the **FileControl** object in each project; once for each form, module, class, and control file, and once for the project file.

The AfterCloseFile event does not occur if the form is dirty and the user selects **No** on the **Save changes to the following files** dialog box. Also, this event does not occur for .Frx files when a project is closed. It occurs when the .Frm file is saved.

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.
- Compare versions of the executable (.EXE) file.

AfterRemoveFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtAfterRemoveFileC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtAfterRemoveFileX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtAfterRemoveFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtAfterRemoveFileS"}

Occurs after a file is removed from the active Visual Basic project.

Syntax

Sub *object*_**AfterRemoveFile(***vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *filename* **As String)**

The AfterRemoveFile event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project from which the file was removed.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was removed, as listed in Settings.
filename	A string expression specifying the name of the file that was removed.

Remarks

The AfterRemoveFile event does not occur for components that are removed before they have been saved.

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.

AfterWriteFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtAfterWriteFileC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtAfterWriteFileX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtAfterWriteFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtAfterWriteFileS"}

Occurs after a file is written to disk.

Syntax

Sub *object*_**AfterWriteFile(***vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *filename* **As String,** *result* **As Integer)**

The AfterWriteFile event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project to which the file was written.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was written, as listed in Settings.
filename	A string expression specifying the name of the file that was written.
result	A numeric expression that specifies the result of the write operation, as listed in Settings.

Settings

The enumerated values for **vbext_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 *result* are:

Value	Description
0	Write was successful.
1	Write was canceled.
2	Write failed.

Remarks

The AfterWriteFile event occurs when the binary data file associated with a component (such as

an .Frx file) is saved for the first time, and 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.
- Compare versions of the executable (.EXE) file.

BeforeLoadFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtBeforeLoadFileC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtBeforeLoadFileX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtBeforeLoadFileA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtBeforeLoadFileS"}

Occurs when a component is added (not opened) to a project, or when a component's associated binary file (such as an .Frx file) is accessed.

Syntax

Sub *object*_**BeforeLoadFile(***vbproject* **As VBProject**, *filenames()* **As String)**

The BeforeLoadFile event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project in which the file is to be loaded.
filenames	A string expression specifying the names of the files to be loaded.

Remarks

This event occurs in all add-ins that are connected to the **FileControl** object. This event occurs several times for a project: once for the project file; once for all the forms, modules, classes, **User** controls, **Property Pages**, and control files; and once for each of the .Frx files. This event occurs if a form file with an associated .Frx file is saved, because the .Frx is loaded when the .Frm file is saved.

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.

DoGetNewFileName Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtDoGetNewFileNameC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtDoGetNewFileNameX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtDoGetNewFileNameA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtDoGetNewFileNameS"}

Occurs whenever a Save As operation is performed on any component or project, whether manually performed from the **File** menu, or programmatically performed.

Syntax

Sub DoGetNewFileName(*vbproject* **As VBProject**, *filetype* **As vbext_FileType**, *newname* **As String**, *oldname* **As String**, *canceldefault* **As Boolean)**

The DoGetNewFileName event syntax has these parts:

Part	Description
vbproject	A **VBProject** object specifying the name of the project which will be written.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file to be written, as listed in Settings.
newname	A string expression specifying the name of the new file. The file specification must be relative to the current **LastUsedPath** property or a fully qualified filename.
oldname	A string expression specifying the old name of the file.
canceldefault	A Boolean expression that determines the default Visual Basic action, as described in Settings.

Settings

The enumerated values for **vbext_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 *canceldefault* are:

Setting	Description
True	Stops triggering this event for any subsequent add-ins connected to the **FileControl** object. If *newname* is a zero-length string ("") when *canceldefault* is set to **True**, the event is canceled; otherwise, the name entered in *newname* is used as

the new filename.

False Continues triggering this event for subsequent add-ins connected to the **FileControl** object. If no add-in sets *canceldefault* to **True**, the **Save File As** or **Make .Exe** dialog box is displayed with the string you entered in *newname* selected.

Remarks

If the *canceldefault* parameter is set to **True**, the **Save File As** dialog box is not displayed. If *canceldefault* is set to **False**, the **Save File As** dialog box displays. If more than one add-ins is connected, and *canceldefault* is set to **True** at any time during a Save As operation, the S**ave File As** dialog box will not display for any of the add-ins until the next Save As operation is performed.

The *newname* argument is initially set to the same value as *oldname*, but any add-in that receives this event can change it. One way to do this is through a custom user interface where you obtain the new name of the file and set *newname* to the user's selection. However, if *canceldefault* is **True** (meaning that a previous add-in has set it to **True**), you shouldn't set *newname* again.

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.

IDTExtensibility Interface

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbobjIDTExtensibilityInterfaceC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbobjIDTExtensibilityInterfaceX":1} {ewc HLP95EN.DLL,DYNALINK,"Properties":"vbobjIDTExtensibilityInterfaceP"} {ewc HLP95EN.DLL,DYNALINK,"Methods":"vbobjIDTExtensibilityInterfaceM"} {ewc HLP95EN.DLL,DYNALINK,"Events":"vbobjIDTExtensibilityInterfaceE"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbobjIDTExtensibilityInterfaceS"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbobjIDTExtensibilityInterfaceA"}

The **IDTExtensibility** interface contains methods that Visual Basic calls when an add-in is connected to it, whether through the Add-In Manager, or some other manner.

The **IDTExtensibility** interface contains pre-configured procedure templates (which includes their parameter lists) that you need to manage add-ins in Visual Basic.

Syntax

Implements IDTExtensiblity

Remarks

The usage of interfaces is new to Visual Basic 5.0. Interfaces enable you to choose a pre-configured procedure template from a module's **Procedure** drop down list, eliminating parameter list entry errors and allowing you to program your applications a bit faster.

An interface's methods are exposed through the **Implements** statement. When the above syntax is entered in the Declarations section of the Class module that handles an add-in's events, the interface's methods become available for your use through the module's **Procedure** and **Object** drop down boxes. To add the code to the module, simply select it from the drop down box.

The **IDTExtensiblity** interface currently contains four methods:

- **OnAddinsUpdate** Method
- **OnConnection** Method
- **OnDisconnection** Method
- **OnStartupComplete** Method

While these are methods to the **IDTExtensibility** interface, to you as a Visual Basic programmer, though, they act and behave like events. In other words, when an add-in is connected to Visual Basic, the **OnConnection** method is called automatically, similar to an event firing. When it is disconnected, the **OnDisconnection** method is called automatically, and so forth.

Important Since an interface is a contract between an object and Visual Basic, you must be sure to implement *all* of the methods in the interface. This means that all four **IDTExtensibility** interface methods are present in your Class module, each containing at least one executable statement. This can consist of as little as a single remark statement, but they must each contain at least one executable statement to prevent the compiler from removing them as empty procedures.

ItemActivated Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtItemActivatedEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtItemActivatedEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProjects;vbevtItemActivatedEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtItemActivatedEventS"}

Occurs when a component is double-clicked in the Project window, and when a project is single-clicked in a project window when there are multiple projects loaded in the IDE.

Syntax

Sub *object*_**ItemActivated(***vbcomponent* **As VBComponent)**
Sub *object*_**ItemActivated(***vbproject* **As VBProject)**

The ItemActivated event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbcomponent	A **VBComponent** object specifying the name of the component that was double-clicked.
vbproject	A **VBProject** object specifying the name of the project which was double-clicked.

Remarks

The ItemActivated event does not occur when a component is double-clicked in the Project window.

ItemAdded Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtItemAddedEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtItemAddedEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjReferencesEvents;vaobjVBProjects;vbevtItemAddedEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtItemAddedEventS"}

Occurs after a project, control, or component is added to the current project.

Syntax

Sub *object*_**ItemAdded (***vbproject* **As VBProject)**
Sub *object*_**ItemAdded (***vbcomponent* **As VBComponent)**
Sub *object*_**ItemAdded (***vbcontrol* **As VBControl)**

The ItemAdded event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project that was loaded.
vbcomponent	A **VBComponent** object specifying the name of the component that was loaded.
vbcontrol	A **VBControl** object specifying the name of the control that was loaded.

ItemRemoved Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtItemRemovedEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtItemRemovedEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjReferencesEvents;vaobjVBProjects;vbevtItemRemovedEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtItemRemovedEventS"}

Occurs after a project, control, or component is removed from the current project.

Syntax

Sub *object*_**ItemRemoved (***vbcontrol* **As VBControl)**
Sub *object*_**ItemRemoved (***vbproject* **As VBProject)**
Sub *object*_**ItemRemoved (***vbcomponent* **As VBComponent)**

The ItemRemoved event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project that was removed.
vbcomponent	A **VBComponent** object specifying the name of the component that was removed.
vbcontrol	A **VBControl** or **SelectedVBControl** object specifying the name of the component that was removed.

ItemRenamed Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtItemRenamedEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtItemRenamedEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProjects;vbevtItemRenamedEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtItemRenamedEventS"}

Occurs after a project, control, or component is renamed in the current project.

Syntax

Sub *object*_**ItemRenamed (***vbproject* **As VBProject)**
Sub *object*_**ItemRenamed (***vbcomponent* **As VBComponent)**
Sub *object*_**ItemRenamed (***vbcontrol* **As VBControl)**

The ItemRenamed event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbproject	A **VBProject** object specifying the name of the project that was renamed.
vbcomponent	A **VBComponent** object specifying the name of the component that was renamed.
vbcontrol	A **VBControl** object specifying the name of the control that was renamed.

ItemSelected Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtItemSelectedEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtItemSelectedEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtItemSelectedEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtItemSelectedEventS"}

Occurs when a component in the Project window or an open designer-window is clicked.

Syntax

Sub *object*_**ItemSelected (***vbcomponent* **As VBComponent)**

The ItemSelected event syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbcomponent	A **VBComponent** object specifying the name of the component that was selected.

RequestChangeFileName Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtRequestChangeFileNameEventC;vbproBooksOnlineJumpTopic"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtRequestChangeFileNameEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtRequestChangeFileNameEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtRequestChangeFileNameEventS"}

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
vbproject	A **VBProject** object specifying the name of the project the new file will be added to.
filetype	An enumerated value (**vbext_FileType**) specifying the type of file that was written, as listed in Settings.
newname	A string expression specifying the name given to the renamed file.
oldname	A string expression specifying containing the name of the file before it was renamed.
cancel	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.

RequestWriteFile Event

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbevtRequestWriteFileEventC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbevtRequestWriteFileEventX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbevtRequestWriteFileEventA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbevtRequestWriteFileEventS"}

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
vbproject	A **VBProject** object specifying the name of the project containing the component.
filename	A string expression containing the name of the file to be saved.
cancel	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.

Activate Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthActivateMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthActivateMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vbmthActivateMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthActivateMethodS"}

Causes the currently selected component in the project window to be activated as if it were double-clicked.

Syntax

object.**Activate**

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

Add Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthAddA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddS"}

- ContainedVBControls collection: Adds a new **VBControl** object to the ContainedVBControls collection.
- VBControls collection: Adds a new **VBControl** object to the VBControls collection.
- VBProjects collection: adds a new, empty project to the set of projects in the VBProjects collection.

Syntax

object.**Add** (*progid* **As String**, **[***relativevbcontrol* **As VBControl] [***before* **As Boolean]**) **As VBControl**
object.**Add** (*projecttype* **As vbext_ProjectType**, **[***exclusive* **As Boolean]**) **As VBProject**

The **Add** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
progid	Required. A string expression specifying the ProgID of the component to be added.
relativevbcontrol	Optional. An existing **VBControl** object specifying the point where the new component is to be inserted.
before	Optional. Default = **False**. A Boolean expression specifying whether the new **VBControl** is to be placed before or after the *relativevbcontrol*.
projecttype	Required. A **VBProject** object specifying the type of the new project. For a list of kinds of projects, see the **Kind** property.
exclusive	Optional. Default = **False**. A Boolean expression specifying whether a new project is added to an existing set of projects, or added as the only project.

Remarks

If the *exclusive* parameter is specified as **True**, then the existing group project is closed and the new project becomes the only project in the collection.

AddCustom Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddCustomMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddCustomMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthAddCustomMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddCustomMethodS"}

Returns a **VBComponent** object, or creates a new custom component and adds it to the project.

Syntax

object.**AddCustom** (**ByVal** *progid* **As String**) **As VBComponent**

The **AddCustom** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
progid	Required. The ProgID of the custom component to be created.

AddFile Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddFileMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddFileMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthAddFileMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddFileMethodS"}

Returns the newly added component.

Syntax

object.**AddFile** (**ByVal** *pathname* **As String**, [*relateddocument* **As Boolean**]) **As VBComponent**

The **AddFile** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
pathname	Required. A string expression specifying the path and filename of the file to open as a template.
relateddocument	Optional (for text files only). Default = **False**. A Boolean expression specifying whether the file is to be treated as a standard module or a document. If set to **True**, then the file added is treated as a document file.

Remarks

Files that are normally Visual Basic project components, such as forms, cause an error if the *relateddocument* parameter is set to **True**. The *relateddocument* parameter is required only when adding text files that can be treated as either standard modules or documents.

AddFromFile Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddFromfileMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddFromfileMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjCodeModule;vaobjVBProjects;vbmthAddFromfileMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddFromfileMethodS"}

Adds or opens a project or group project.

Syntax

object.**AddFromFile** (**ByVal** *pathname* **As String**, [*exclusive* **As Boolean**]) **As VBNewProjects**

The **AddFromFile** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
pathname	Required. A string expression specifying the path to the file to use as the template.
exclusive	Optional. Default = **False**. A Boolean expression. If set to **True**, then the existing group project is closed and the new project is created as the only open project.

Remarks

If the file is a group project file and *exclusive* is set to **False**, then all projects in that group project are added to the current group project. If the file is a group project file and *exclusive* is set to **True**, then the current group project is replaced by the specified one.

AddFromTemplate Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddFromTemplateMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddFromTemplateMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProjects;vbmthAddFromTemplateMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddFromTemplateMethodS"}

- VBComponents collection: Returns the newly created component, and creates a new component from a template.
- VBProjects collection: Returns a collection of all projects added as a result of a call to this method, or creates a new project using an existing project as a template.

Syntax

object.**AddFromTemplate** (*filename* **As String**) **As VBComponent**
object.**AddFromTemplate** (**ByVal** *pathname* **As String**, [*exclusive* **As Boolean**]) **As VBNewProjects**

The **AddFromTemplate** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
filename	Required. A string expression specifying the path and filename of the file to open as a template.
exclusive	Optional. Default = **False**. A Boolean expression. If set to **True**, then the existing group project is closed and the new project is created as the only open project.
pathname	Required. A string expression specifying the path to the file to use as the template.

Remarks

If the file type referenced is a group project file, and *exclusive* is set to **False**, then all projects in that file are created as templates and added to the current set of open projects. If *exclusive* is set to **True**, however, the current group project is closed and a new group project created, and all projects within the group project template are created as project templates. The object returned by the method is **Nothing**.

New project or projects are given the usual default names.

AddToolboxProgID Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddToolboxProgIDMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddToolboxProgIDMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbmthAddToolboxProgIDMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddToolboxProgIDMethodS"}

Places the control or embedded component in the toolbox and adds a control reference to the project.

Syntax

object.**AddToolboxProgID** (**ByVal** *progid* **As String**, [*filename* **As String**])

The **AddToolboxProgID** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
progid	Required. A string expression specifying the programmatic identifier (ProgID) of the compound document object to add to the Visual Basic toolbox. Either a version-independent or version-dependent ProgID can be used. If a version-independent progid is specified, the most recent version is used. If the compound document object has an associated type library, this type library is referenced as well.
filename	Optional. A string expression specifying the filename of the desired type library to be added to Visual Basic. A complete pathname can be used, but if the file isn't found, the directories searched by the Windows **OpenFile** function are searched, even if a complete pathname is specified.

Item Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthItemAddInC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthItemAddInX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjLinkedWindows;vaobjVBProjects;vbmthItemAddInA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthItemAddInS"}

Returns an item from the specified collection by either name or index.

- AddIns collection:
- ContainedVBControls collection:
- Members collection:
- SelectedVBControls collection:
- VBControls collection:
- VBNewProjects collection:

Syntax

object.**Item (***index***)**
object.**Item (***collectionindex,* **[***controlindex***]) As VBControl**
object.**Item (***var***) As Member**

The **Item** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	Required. A variant expression specifying the name or index in the collection of the object to be accessed.
collectionindex	Required. A numeric expression specifying the index
controlindex	Optional.
var	Required.

Remarks

There is no guarantee that a given index number for a collection will always point to the same item, because items may be added or deleted from the collection. Using index numbers for the collection is useful only when iterating through the whole collection and no items are added or deleted during the iteration.

MakeCompiledFile Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbfctMakeCompiledFileMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbfctMakeCompiledFileMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbfctMakeCompiledFileMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbfctMakeCompiledFileMethodS"}

Causes the current project to be written as an Exe, Dll, or control, depending on project type.

Syntax

object.**MakeCompiledFile**

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

OnAddinsUpdate Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthUpdateMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthUpdateMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthUpdateMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthUpdateMethodS"} {ewc HLP95EN.DLL,DYNALINK,"Interfaces":"vbmthUpdateMethodI"}

Occurs automatically when changes to the Vbaddin.Ini file are saved.

Syntax

object.**IDTExtensibility_OnAddinsUpdate (***custom()* **As Variant)**

The **OnAddinsUpdate** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
custom()	An array of variant expressions to hold user-defined data.

Remarks

This method is part of the IDTExtensibility interface, which you should implement in the class that provides your connection object.

Important Don't directly enter the syntax given above. Instead, use the **Implements** statement to generate the appropriate method template for the interface. To do this, in the Declarations section of the Class module that provides your add-in's connection object, enter:

```
Implements IDTExtensibility
```

After adding this line, you can select **IDTExtensibility** from the module's **Object** drop down box. Select each method from the **Procedure** drop down to get the procedure template shown above in Syntax. Notice that the necessary code is automatically added to the Class module.

An interface's methods are exposed through the **Implements** statement. When the above syntax is entered in the Declarations section of the Class module that handles an add-in's events, the interface's methods become available for your use through the module's **Procedure** and **Object** drop down boxes. To add the code to the module, select the method from the **Procedure** drop down box.

Note While the **OnAddinsUpdate** method is a method to the IDTExtensibility interface, to you as a Visual Basic programmer it acts and behaves like an event. In other words, when changes made to the Vbaddin.Ini file are saved, any code in the **OnAddinsUpdate** method automatically occurs, just as if it were an event procedure.

Important Since an interface is a contract between an object and Visual Basic, you must be sure to implement *all* of the methods in the interface. This means that all four **IDTExtensibility** interface methods are present in your Class module, each containing at least one executable statement. This can consist of as little as a single remark statement, but they must each contain at least one executable statement to prevent the compiler from removing them as empty procedures.

OnConnection Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthOnConnectionMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthOnConnectionMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthOnConnectionMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthOnConnectionMethodS"} {ewc HLP95EN.DLL,DYNALINK,"Interfaces":"vbmthOnConnectionMethodI"}

Occurs when an add-in is connected to the Visual Basic IDE, either through the **Add-In Manager** dialog box or another add-in.

Syntax

object. **IDTExtensibility_OnConnection (***vbinst* **As Object**, *connectmode* **As vbext_ConnectMode**, *addininst* **As AddIn**, *custom()* **As Variant)**

The **OnConnection** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
vbinst	An object representing the instance of the current Visual Basic session.
connectmode	An enumerated value of type **vbext_ConnectMode**, as specified in Settings.
addininst	An **AddIn** object representing the instance of the add-in.
custom()	An array of variant expressions to hold user-defined data.

Settings

The settings for *connectmode* (**vbext_ConnectMode**) are:

Constant	Value	Description
vbext_cm_AfterStartup	0	Add-in was started after the initial **Open Project** dialog box was shown.
vbext_cm_Startup	1	Add-in was started before the initial **Open Project** dialog box was shown.
vbext_cm_External	2	Add-in was started externally by another program or component.

Remarks

This method is part of the IDTExtensibility interface, which you should implement in the class that provides your connection object.

Important Don't directly enter the syntax given above. Instead, use the **Implements** statement to generate the appropriate method template for the interface. To do this, in the Declarations section of the Class module that provides your add-in's connection object, enter:

```
Implements IDTExtensibility
```

After adding this line, you can then select **IDTExtensibility** from the module's **Object** drop down box. Select each method from the **Procedure** drop down to get the procedure template shown above in Syntax. Notice that the necessary code is automatically added to the Class module.

An interface's methods are exposed through the **Implements** statement. When the above syntax is entered in the Declarations section of the Class module that handles an add-in's events, the interface's methods become available for your use through the module's **Procedure** and **Object** drop down boxes. To add the code to the module, select the method from the **Procedure** drop down box.

Note While the **OnConnection** method is a method to the IDTExtensibility interface, to you as a Visual Basic programmer it acts and behaves like an event. In other words, when an add-in is connected to the Visual Basic IDE, either through the **Add-In Manager** dialog box or another add-in, any code in the **OnConnection** method automatically occurs, just as if it were an event procedure.

Important Since an interface is a contract between an object and Visual Basic, you must be sure to implement *all* of the methods in the interface. This means that all four **IDTExtensibility** interface methods are present in your Class module, each containing at least one executable statement. This can consist of as little as a single remark statement, but they must each contain at least one executable statement to prevent the compiler from removing them as empty procedures.

OnDisconnection Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthOnDisconnectionMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthOnDisconnectionMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthOnDisconnectionMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthOnDisconnectionMethodS"} {ewc HLP95EN.DLL,DYNALINK,"Interfaces":"vbmthOnDisconnectionMethodI"}

Occurs when an add-in is disconnected from the Visual Basic IDE, either programmatically or through the **Add-In Manager** dialog box.

Syntax

object. **IDTExtensibility_OnDisconnection (***removemode* **As vbext_DisconnectMode**, *custom()***As Variant)**

The **OnDisconnection** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
removemode	An enumerated value of type **vbext_DisconnectMode**, as specified in Settings.
custom()	An array of variant expressions to hold user-defined data.

Settings

The settings for *removemode* (**vbext_ConnectMode**) are:

Constant	Value	Description
vbext_dm_HostShutdown	0	Add-in was removed by the add-in's host being closed.
vbext_dm_UserClosed	1	Add-in was removed by the user closing it.

Remarks

This method is part of the IDTExtensibility interface, which you should implement in the class that provides your connection object.

Important Don't directly enter the syntax given above. Instead, use the **Implements** statement to generate the appropriate method template for the interface. To do this, in the Declarations section of the Class module that provides your add-in's connection object, enter:

```
Implements IDTExtensibility
```

After adding this line, you can then select **IDTExtensibility** from the module's **Object** drop down box. Select each method from the **Procedure** drop down to get the procedure template shown above in Syntax. Notice that the necessary code is automatically added to the Class module.

An interface's methods are exposed through the **Implements** statement. When the above syntax is entered in the Declarations section of the Class module that handles an add-in's events, the interface's methods become available for your use through the module's **Procedure** and **Object** drop down boxes. To add the code to the module, select the method from the **Procedure** drop down box.

Note While the **OnDisconnection** method is a method to the IDTExtensibility interface, to you as a Visual Basic programmer it acts and behaves like an event. In other words, when an add-in is disconnected from the Visual Basic IDE, either programmatically or through the **Add-In Manager** dialog box, any code in the **OnDisconnection** method automatically occurs, just as if it were an event procedure.

Important Since an interface is a contract between an object and Visual Basic, you must be sure to implement *all* of the methods in the interface. This means that all four **IDTExtensibility** interface methods are present in your Class module, each containing at least one executable statement. This can consist of as little as a single remark statement, but they must each contain at least one executable statement to prevent the compiler from removing them as empty procedures.

OnStartupComplete Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthOnStartupCompleteMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthOnStartupCompleteMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthOnStartupCompleteMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthOnStartupCompleteMethodS"} {ewc HLP95EN.DLL,DYNALINK,"Interfaces":"vbmthOnStartupCompleteMethodI"}

Occurs when the startup of the Visual Basic IDE is complete.

Syntax

object. **IDTExtensibility_OnStartupComplete (***custom()* **As Variant)**

The **OnStartupComplete** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
custom()	An array of variant expressions to hold user-defined data.

Remarks

This method is part of the IDTExtensibility interface, which you should implement in the class that provides your connection object.

Important Don't directly enter the syntax given above. Instead, use the **Implements** statement to generate the appropriate method template for the interface. To do this, in the Declarations section of the Class module that provides your add-in's connection object, enter:

```
Implements IDTExtensibility
```

After adding this line, you can then select **IDTExtensibility** from the module's **Object** drop down box. Select each method from the **Procedure** drop down to get the procedure template shown above in Syntax. Notice that the necessary code is automatically added to the Class module.

An interface's methods are exposed through the **Implements** statement. When the above syntax is entered in the Declarations section of the Class module that handles an add-in's events, the interface's methods become available for your use through the module's **Procedure** and **Object** drop down boxes. To add the code to the module, select the method from the **Procedure** drop down box.

Note While the **OnStartupComplete** method is a method to the IDTExtensibility interface, to you as a Visual Basic programmer it acts and behaves like an event. In other words, when the startup of the Visual Basic IDE is complete, any code in the **OnStartupComplete** method automatically occurs, just as if it were an event procedure.

Important Since an interface is a contract between an object and Visual Basic, you must be sure to implement *all* of the methods in the interface. This means that all four **IDTExtensibility** interface methods are present in your Class module, each containing at least one executable statement. This can consist of as little as a single remark statement, but they must each contain at least one executable statement to prevent the compiler from removing them as empty procedures.

ReadProperty Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthReadPropertyFunctionC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthReadPropertyFunctionX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vaobjVBProject;vbmthReadPropertyFunctionA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthReadPropertyFunctionS"}

Returns a string from the specified user-defined section and key in the project's .Vbp or component file.

- **VBComponent** object:
- **VBProject** object:

Syntax

object.**ReadProperty** (*key* **As String**) **As String**
object.**ReadProperty** (*section* **As String**, *key* **As String**) **As String**

The **ReadProperty** function syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
section	A string expressioncontaining the name of the section where the key is found.
key	A string expressioncontaining the name of the key to return.

Remarks

If the *section* or *key* area in the file is empty or doesn't exist, you'll get run-time error 5: "`Illegal function call`."

Reload Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthReloadC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthReloadX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vbmthReloadA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthReloadS"}

Reloads the specified component from disk, discarding any unsaved changes.

Syntax

object.**Reload**

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

Remarks

Cursor position, code window and form visibility are not affected by the **Reload** method. **Reload** doesn't change the setting which indicates whether the project was edited since the last time it was saved.

Remove Method (Visual Basic Extensibility)

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthRemoveMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthRemoveMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjLinkedWindows;vaobjVBProjects;vbmthRemoveMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthRemoveMethodS"}

Removes an item from a collection.

Syntax

object.**Remove (***index***)**

The **Item** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	Required. A variant expression specifying the name or index in the collection of the object to be accessed.

Remarks

For the LinkedWindows collection, removes a window from the collection of currently linked windows. The removed window becomes a floating window that has its own LinkedWindowFrame. This is the point at which LinkedWindowFrame windows are created.

For the VBProjects collection, removes the specified project from the collection.

For the References collection, removes the specified reference from the collection.

SaveAs Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthSaveAsMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthSaveAsMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vaobjVBProject;vaobjVBProjects;vbmthSaveAsMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthSaveAsMethodS"}

Saves a component or project to a given location using a new filename.

Syntax

object.**SaveAs (***newfilename* **As String)**

The **SaveAs** method syntax has these parts:

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
newfilename	Required. A string expression specifying the new filename for the component to be saved.

Remarks

If a new path name is given, it is used. Otherwise, the old path name is used. If the new filename is invalid or refers to a read-only file, an error occurs.

When a form is saved, *newfilename* specifies the new name of the form file itself. The .Frx file, if applicable, is saved automatically with an .Frx extension.

Note Successfully invoking this method causes the associated events from the **FileControl** object to be invoked.

Update Method

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbmthAddinsUpdateMethodC;vbproBooksOnlineJumpTopic"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbmthAddinsUpdateMethodX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbmthAddinsUpdateMethodA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbmthAddinsUpdateMethodS"}

Refreshes the contents of the **AddIns** collection from the add-ins listed in the Vbaddin.ini file in the same manner as if the user had opened the Add-In Manager dialog box.

Syntax

object.**Update**

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

Remarks

All add-ins listed in the Vbaddin.ini file must be registered ActiveX components in the Registry before they can be used in Visual Basic.

BuildFileName Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproBuildFileNamePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproBuildFileNamePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbproBuildFileNamePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproBuildFileNamePropertyS"}

Returns the executable or DLL name that will be used when the project is built.

Syntax

object.**BuildFileName**

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

CanPaste Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCanPastePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproCanPastePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproCanPastePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCanPastePropertyS"}

Returns a boolean value indicating whether or not the Clipboard contains appropriate information (that is, controls) for pasting to the form.

Syntax

object.**CanPaste**

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

CodeLocation Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCodeLocationPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproCodeLocationPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproCodeLocationPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCodeLocationPropertyS"}

Returns the line location in the code module where the member is defined.

Syntax

object.**CodeLocation** [= *propkind*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
propkind	An enumerated value of **vbext_PropertyKind**, as described in Settings.

Settings

The settings for **vbext_PropertyKind** are:

Constant	Value	Description
vbext_PropertyGet	1	(Default) Returns the code location for the Get element of the property.
vbext_PropertyLet	2	Returns the code location for the Let element of the property.
vbext_PropertySet	3	Returns the code location for the Set element of the property.

CompatibleOLEServer Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproCompatibleOleServerPropertyC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproCompatibleOleServerPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbproCompatibleOleServerPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproCompatibleOleServerPropertyS"}

Returns or sets the compatible ActiveX component of this project.

Syntax

object.**CompatibleOLEServer**

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

ControlObject Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproControlObjectPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproControlObjectPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproControlObjectPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproControlObjectPropertyS"}

Returns a reference to an instance of the design-time IDispatch pointer provided by the control. If there isn't one, this property returns **Nothing**.

Syntax

object.**ControlObject**

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

Remarks

For example, the **Toolbar** control provides an object through a **Property Page's ControlObject** property to set the number of buttons.

ControlType Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproControlTypePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproControlTypePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproControlTypePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproControlTypePropertyS"}

Returns the type of run-time window that a control creates.

Syntax

object.**ControlType As vbext_ControlType**

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

Settings

The settings for **vbext_ControlType** are:

Constant	Value	Description
vbext_ct_Light	1	(Default) No hWnd at run-time.
vbext_ct_Standard	2	hWnd at run-time.
vbext_ct_Container	3	hWnd at run-time, and can contain other controls.

DisplayModel Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproDisplayModelPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproDisplayModelPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBE;vbproDisplayModelPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproDisplayModelPropertyS"}

Returns or sets the display model used by the system.

Syntax

object.**DisplayModel As vbext_VBADisplayModel**

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

Settings

The settings for **vbext_VBADisplayModel** are:

Constant	Value	Description
vbext_dm_SDI	0	(Default) Display model is SDI (Single Document Interface).
vbext_dm_MDI	1	Display model is MDI (Multiple Document Interface).

FileControlEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFileControlEventsPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFileControlEventsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjEvents;vbproFileControlEventsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFileControlEventsPropertyS"}

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

Syntax

object.**FileControlEvents**

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

FileCount Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFileCountC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFileCountX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vbproFileCountA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFileCountS"}

Returns the number of files associated with a given component.

Syntax

object.**FileCount**

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

Remarks

The primary use for **FileCount** is to alert you to whether a component has an associated .Frx file.

For most components, this property setting is 1, but it can be greater. For example, if an .FRX file is associated with a form file, the **FileCount** property setting is 2.

FileName Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFileNamePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFilenamePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbproFilenamePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFilenamePropertyS"}

Returns the full path name of the group project file.

Syntax

object.**Filename**

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

Remarks

Group projects have no name other than the file name.

The path name returned will always be provided as an absolute path (for example, "c:\projects\myproject.vbp"), even if it is shown as a relative path in Visual Basic (such as "..\projects").

FileNames Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFileNamesC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFileNamesX"} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vbproFileNamesA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFileNamesS"}

Returns the current path name(s) in which the component will be stored.

Syntax

object.**FileNames(***index***)**

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
index	A long integer specifying the location of the filename in the indexed string to return.

Remarks

The path name returned will always be provided as an absolute path (for example, "c:\projects\myproject.vbp"), even if it is shown as a relative path in Visual Basic (such as "..\projects").

The number of entries in the indexed string is determined by the **FileCount** property setting. The indexed string for classes and modules contains only one filename, while the indexed string for forms contains both the .Frm and .Frx filename for the form. The values of the filenames are updated when the **SaveAs** method is invoked on the *object.*

FullName Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproFullNamePropertyAddInC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproFullNamePropertyAddInX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBE;vbproFullNamePropertyAddInA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproFullNamePropertyAddInS"}

Returns the full path name of the Visual Basic IDE. (That is, the path where vb5.exe was run.)

Syntax

object.**FullName**

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

Remarks

The path name returned is always provided as an absolute path (for example, "c:\projects\myproject.vbp"), even if it is shown as a relative path in Visual Basic (such as "..\projects").

IconState Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIconstateC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIconstateX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBComponent;vaobjVBProject;vaobjVBProjects;vbproIconstateA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIconstateS"}

Returns or sets the source code control icon (or "glyph") for the project in the project window, indicating its status.

Syntax

object.**IconState** [= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	A long integer or constant that determines the file status, as described in Settings.

Settings

The settings for *value* are:

Constant	Value	Description
vbextSCCStatusNotControlled	0	File is not under source code control.
vbextSCCStatusControlled	1	File is under source code control.
vbextSCCStatusCheckedOut	2	File is checked out to current user.
vbextSCCStatusOutOther	4	File is checked out to another user.
vbextSCCStatusOutOfDate	32	The file is not the most recent.
vbextSCCStatusShared	512	File is shared between projects.

Remarks

The **IconState** property can be logically **OR**'ed together to form combined states.

IndexedValue Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproIndexedValuePropertyAddInC;vbproIndexedValuePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproIndexedValuePropertyAddInX;vbproIndexedValuePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjProperty;vbproIndexedValuePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproIndexedValuePropertyS"}

Returns or sets a value in an indexed list or an array.

Syntax

object.**IndexedValue** [*index1*, *index2*,][*index3*,] [*index4*]][= *value*]

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
indexn	A numeric expression specifying index position. **IndexedValue** accepts up to four indices. The number of indices accepted by **IndexedValue** is the value for the same property returned by the **NumIndices** property.
value	An expression that evaluates to a type acceptable to the control.

Remarks

If the property is a list (as indicated by **NumIndices**), then **IndexedValue** returns the element of the list specified by the index parameters.

IndexedValue is used only if the value of the **NumIndices** property is greater than zero. Values in indexed lists, as in the **List** property of a **ListBox** control, are set or returned with a single index.

InSelection Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproInSelectionPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproInSelectionPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproInSelectionPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproInSelectionPropertyS"}

Returns or assigns a control's selection state.

Syntax

object.**InSelection**

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

Remarks

If the **InSelection** property of a control contained within another control is set to **True**, then any controls not within that control (or any controls within other controls) will be unselected.

LastUsedPath Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproLastUsedPathPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproLastUsedPathPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBE;vbproLastUsedPathPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproLastUsedPathPropertyS"}

Returns or sets the last path used for the file dialog boxes used in Visual Basic, such as the **Open Project** dialog box.

Syntax

object.**LastUsedPath ([***pathname* **As String])**

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

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
pathname	(Optional) A string containing the last path name used for a file dialog box.

NumIndices Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproNumIndicesPropertyAddInC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproNumIndicesPropertyAddInX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjProperty;vbproNumIndicesPropertyAddInA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproNumIndicesPropertyAddInS"}

Returns the number of indices on the property returned by the **Property** object, which is the number of indices required to access the value.

Syntax

object.**NumIndices**

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

Remarks

The value of **NumIndices** can have a value from 0 to 4. For normal properties, as in the **ForeColor** property, **NumIndices** returns 0. Conventionally indexed properties, such as the **List** property of a **ListBox** control, return 1. Property arrays might return a 2.

ProgID Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproProgIDPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproProgIDPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproProgIDPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproProgIDPropertyS"}

Returns the ProgID (programmatic ID) for the control represented by the **VBControl** object.

Syntax

object.**ProgID**

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

ReadOnlyMode Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproReadOnlyModeC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproReadOnlyModeX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBE;vbproReadOnlyModeA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproReadOnlyModeS"}

Returns or sets a value that determines how the Visual Basic development environment interacts with read-only files.

Syntax

object.**ReadOnlyMode [**= *value***]**

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

Part	Description
object	An object expression that evaluates to an object in the Applies To list.
value	A long integer or constant that determines the project status, as described in Settings.

Settings

The settings for *value* are:

Value	Description
1	Strict. **Remove File** and **Add File** commands are not available if the project file is read-only on the disk. Windows and forms can be moved, but changes will not be saved. For files which are also read-only on disk, the Code Window is read-only; controls cannot be added or removed, control positions are locked, and custom **Properties** dialog boxes are disabled.
0	Lenient. Modifications can be made to code, forms, and the project if files are read-only, but changes can't be saved back to the files.

Scope Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproScopePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproScopePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproScopePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproScopePropertyS"}

Returns whether a member is public, private, or friend.

Syntax

object.**Scope**

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

SelectedVBControlsEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproSelectedVBControlsEventsPropertyC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproSelectedVBControlsEventsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjEvents;vbproSelectedVBControlsEventsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproSelectedVBControlsEventsPropertyS"}

Returns all events supported by the controls currently selected on a form.

Syntax

object.**SelectedVBControlsEvents (***vbproject* **As Variant)**

The **SelectedVBControlsEvents** 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 form and controls.

Remarks

Returns an event object of type **SelectedVBControlsEvents**. This event is sourced from a VBForm.

StartMode Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproStartModePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproStartModePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProject;vbproStartModePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproStartModePropertyS"}

Returns or sets the start mode of the project.

Syntax

object.**StartMode**

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

Remarks

Returns **RunMode** if code is currently executing, **BreakMode** if there is code on the stack but not running, and **DesignMode** otherwise.

StartProject Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproStartProjectPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproStartProjectPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBProjects;vbproStartProjectPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproStartProjectPropertyS"}

Returns or sets the project that will start when the user selects **Start** from the **Run** menu, or presses the F5 key.

Syntax

object.**StartProject**

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

StartUpObject Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproStartUpObjectPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproStartUpObjectPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproStartUpObjectPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproStartUpObjectPropertyS"}

Returns or sets the startup component for the project.

Syntax

object.**StartUpObject**

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

Return Values

The value that is returned is a variant that contains either an enumerated value of type **vbext_StartupObject**, or a **VBComponent** object that represents the startup object.

The **StartUpObject** property settings for **vbext_StartUpObject** are:

Constant	Value	Description
vbext_so_SubMain	0	Startup object is the sub Main.
vbext_so_None	1	There is no startup object.

Remarks

Only visual at run-time project items can be the startup object.

Static Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproStaticPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproStaticPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vbproStaticPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproStaticPropertyS"}

Returns whether the referenced variable or method is declared as **Static**.

Syntax

object.**Static**

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

TemplatePath Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTemplatePathPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproTemplatePathPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjVBE;vbproTemplatePathPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTemplatePathPropertyS"}

Returns the full pathname where Visual Basic stores template files.

Syntax

object.**TemplatePath**

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

Type Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproTypePropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproTypePropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjReference;vaobjVBComponent;vaobjVBProject;vaobjWindow;vbproTypePropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproTypePropertyS"}

Returns the type of the currently selected member or project.

Syntax

object.**Type**

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

Return Values

The **Type** property settings for the **Member** object (**vbext_MemberType**) are:

Constant	Value	Description
vbext_mt_Method	1	Member is a method.
vbext_mt_Property	2	Member is a property.
vbext_mt_Variable	3	Member is a variable.
vbext_mt_Event	4	Member is an event.
vbext_mt_Enum	5	Member is an enumerated value.
vbext_mt_Const	6	Member is a constant.
vbext_mt_EventSink	7	Member is an event sink.

The **Type** property settings for the **Project** object (**vbext_ProjectType**) are:

Constant	Value	Description
vbext_pt_StandardExe	1	Project type is Standard Exe.
vbext_pt_ActiveXExe	2	Project type is ActiveX Exe.
vbext_pt_ActiveXDll	3	Project type is ActiveX Dll.
vbext_pt_ActiveXControl	4	Project type is ActiveX control.

VBComponentsEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproVBComponentsEventsPropertyC"}
{ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVBComponentsEventsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjEvents;vbproVBComponentsEventsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVBComponentsEventsPropertyS"}

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.

VBControlsEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproVBControlsEventsC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVBControlsEventsX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjEvents;vbproVBControlsEventsA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVBControlsEventsS"}

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.

VBProjectsEvents Property

{ewc HLP95EN.DLL,DYNALINK,"See Also":"vbproBooksOnlineJumpTopic;vbproVBProjectsEventsPropertyC"} {ewc HLP95EN.DLL,DYNALINK,"Example":"vbproVBProjectsEventsPropertyX":1} {ewc HLP95EN.DLL,DYNALINK,"Applies To":"vaobjEvents;vbproVBProjectsEventsPropertyA"} {ewc HLP95EN.DLL,DYNALINK,"Specifics":"vbproVBProjectsEventsPropertyS"}

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.