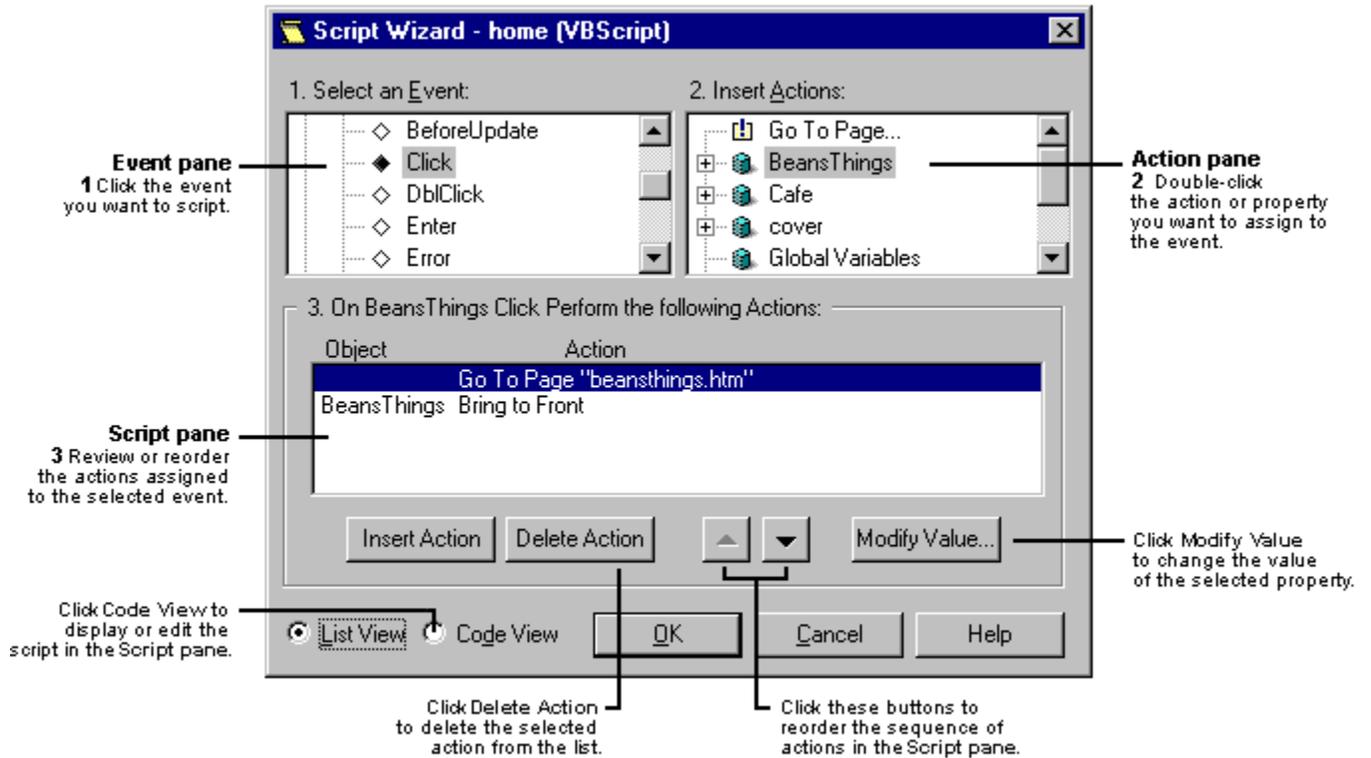


# Using the Script Wizard in List View

[See Also](#)



## The Event Pane

The Event pane provides a hierarchical view of all the objects and events that you can script. These include ActiveX controls and form fields that you've inserted into your page.

In the hierarchy, objects are listed in alphabetical order by ID name. Under each object are the events that you can script. The icons represent different types of events and objects.

When you click an event, the Script Wizard displays that event handler in the Script pane.

## The Action Pane

The Action pane provides a hierarchical view of the actions and properties you can use in the event handler, as well as the global variables and procedures defined for the page. The icons represent different types of actions, properties, and objects.

- When you double-click an action, the Script Wizard adds that action to the list in the Script pane.
- When you double-click a property, you'll see a dialog box prompting you to choose a value for the property. The type of dialog box you see, and the values you can select, depend on the type of property you double-click.

Once you've specified a value for the property, the Script Wizard adds that property to the list in the Script pane.

**Tip** You can quickly script a jump to another page, change a control's front to back layering, or hide or show a control, by double-clicking the **Go To Page...** or **Hide Control/Show Control** actions.

## The Script Pane

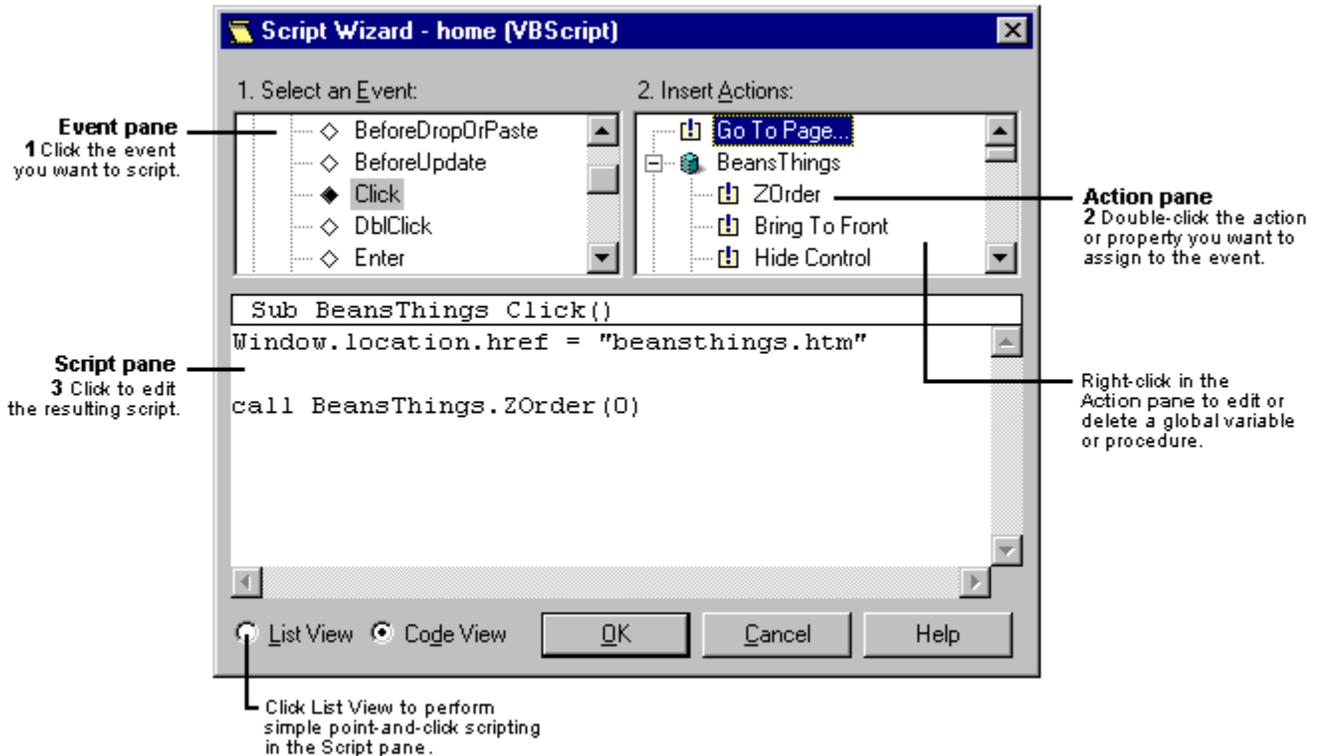
You can script multiple actions for any given event, and they'll be executed in the order they appear in the list in the Script pane. Use the **Up** and **Down Arrow** buttons to reorder the actions in the list, and the **Insert Action** and **Delete Action** buttons to add or remove actions from the list. If you specified a property, you can edit it by selecting that property and clicking the **Modify Value** button.

In the Event pane, if you click an event handler that's associated with a custom action — for example, a script that contains an "if" statement — you'll see a message in the Script pane advising you to click **Code view** to edit the action.

When you click **OK** or close the Script Wizard, the event handlers you create, and the global variables and procedures you define, are stored as VBScript or JavaScript in your page. To discard any changes you don't want to take effect, click **Cancel**.

# Using the Script Wizard in Code View

[See Also](#)



## The Event Pane

The Event pane provides a hierarchical view of all the objects and events that you can script. These include ActiveX controls and form fields that you've inserted into your page.

In the hierarchy, objects are listed in alphabetical order by ID name; under each object are the events that you can script. The icons represent different types of objects and events. When you click an event, the script for that event appears in the Script pane.

## The Action Pane

The Action pane provides a hierarchical view of all the actions, properties, variables, and procedures that can be invoked; each is represented by a different icon. When you double-click an action, the script for that action appears in the Script pane.

**Tip** You can quickly script a jump to another page, change a control's front to back layering, or hide or show a control, by double-clicking the **Go To Page...** or **Hide Control/Show Control** actions.

## The Script Pane

The Script pane displays the actual script in the default scripting language you specified for the page, either VBScript or JavaScript. Click an insertion point to edit the script. The Script Wizard automatically adds **end sub** or **}** end-of-procedure marks to any script you create.

**Tip** In the Script pane, you can Cut, Copy, or Paste script, or change the display Font, using the shortcut menu.

When you click **OK** or close the Script Wizard, the scripts you create, and the global variables and procedures you define, are stored as VBScript or JavaScript in your page. To discard any script or

changes you don't want to take effect, click **Cancel**.

## New Global Variable Dialog Box

[See Also](#)

This dialog box appears when you're working in the Action pane of the Script Wizard, and you click **New Global Variable** on the [shortcut menu](#). Use this dialog box to add a new [global variable](#) to [HTML](#) for the active Web page.

Type the name of the global variable you want to add. You can include subscripts or an initial value in the proper syntax for the script language you're using (either VBScript or JavaScript).

When you click OK, that variable is inserted at the beginning of HTML, in the form **dim** *variable-name* for VBScript or **var** *variable-name* for JavaScript.

## Edit Global Variable Dialog Box

### See Also

This dialog box appears when you're working in the Action pane of the Script Wizard, and you click **Edit Global Variable** on the shortcut menu. Use this dialog box to edit a global variable to HTML for the active Web page.

Type the new name for the global variable. You can include subscripts or an initial value in the proper syntax for the script language you're using (either VBScript or JavaScript).

When you click OK, that variable is inserted at the beginning of HTML, in the form **dim** *variable-name* for VBScript or **var** *variable-name* for JavaScript.

## Text String Value Dialog Box

[See Also](#)

This example of a Text String Value dialog box asks you to insert a text string for the GoTo Page action. You can either type in a known value, such as the URL for the page associated with this particular event, or click the Custom or Color buttons to assign a value.

When the value is inserted into the script, quotation marks are automatically added to the string. For example:

<u>What you enter:</u>	<u>Result:</u>	<u>Result for INPUT, SELECT, TEXTAREA, or FORM event</u>
Sample Text	“Sample Text”	‘Sample Text’

You receive an error message if you enter an invalid text string. This can happen because of a single quotation mark (') or a double quotation mark (") in the value you typed. For example:

<u>What you enter:</u>	<u>Result:</u>	<u>Result for INPUT, SELECT, TEXTAREA, or FORM event</u>
It's the Internet	“It's the Internet” (Valid)	‘It's the Internet’ (Invalid)
It is the “Internet”	“It is the “Internet”” (Invalid)	‘It is the “Internet”’ (Valid)

If you are scripting an INPUT, TEXTAREA, SELECT or FORM element, then the problem is the presence of a single quotation mark in your value. In all other cases, the problem is the presence of a double quotation mark in your value. The solution to this problem depends on the scripting language you are working with.

### VBScript Solution

If you are using VBScript, the solution is to double the number of quotation marks. If you are typing a double quotation mark in your text string which is causing problems, turn it into two (") characters: ("""). If you are typing a single quotation mark which is causing problems, turn it into two (') characters: ('').

<u>Invalid result:</u>	<u>What you should type:</u>
‘It's the Internet	It''s the Internet
“It is the “Internet””	It is the ""Internet""

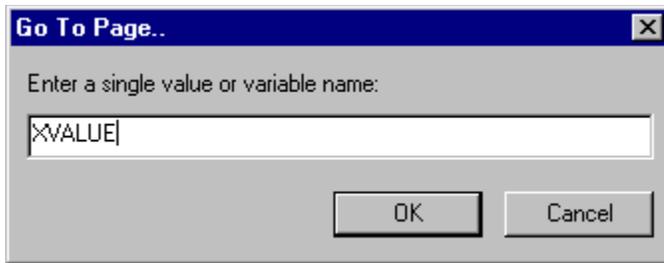
### JavaScript Solution

If you are using JavaScript, you can place a backslash character in front of the problem character. For example, if a double quotation mark is causing problems, change it to (\"). Similarly, if a single quotation mark is causing problems, change it to (\').

<u>Invalid result:</u>	<u>What you should type:</u>
‘It's the Internet’	It\'s the Internet
“It is the “Internet””	It is the \"Internet\"

## Custom Value Dialog Box

[See Also](#)



This dialog box appears when clicking Custom from the previous Go To Page Text String Value dialog box. If you have a pre-defined variable, property value or value that is not a simple constant, you can enter it here.

The Custom Value dialog box is the default Value dialog box and appears when the selected property is not categorized as a text string, number, Boolean value, or enumeration. You can also bring this dialog box up by clicking the Custom option in one of the Value dialog boxes as previously displayed, or click the Modify Value option in the Script pane.

## Color Dialog Box

### See Also

Clicking the Color button displays the standard Windows color picker window. Selecting a color creates a value in the Script pane depending on which property you are modifying.

If you are modifying a Window object property, the value is displayed according to HTML rules in RRGGBB format as hex constants with the # symbol as the prefix.

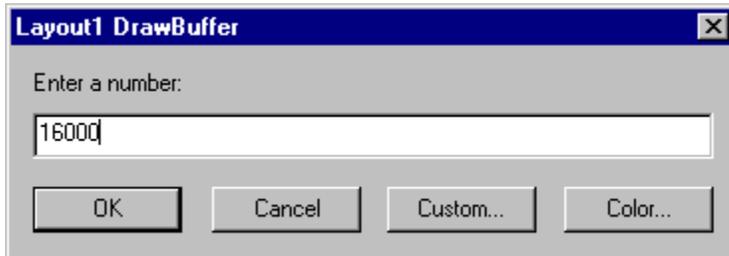
If you are modifying a control property and reaching the Color dialog box directly or by choosing Color from the Number Value dialog box, the value is interpreted as an OLE color in BBGGRR format as hex constants, with the &H symbol as the prefix for VBScript, and the 0x prefix for JavaScript.

Note that the Color button is available for many properties and not just for changing the foreground or background colors of an object. For example, if you change the value of the **Caption** property from text such as “Push” to the color red, the actual text “#FF0000” is placed in your control rather than the color red.

HTML-named colors such as “red” or “white” are not supported values.

## Number Value Dialog Box

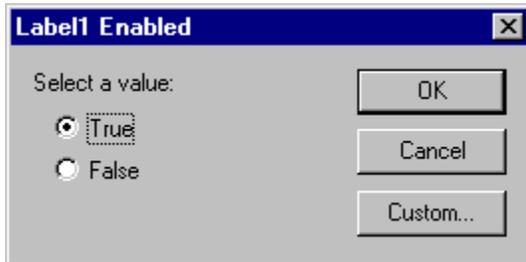
[See Also](#)



This example of a Number Value dialog box asks you to insert a number for the Layout1 **DrawBuffer** property. Only valid decimal, hexadecimal, or octal numbers in the syntax of the current scripting language may be entered in this dialog box. If an invalid number is entered, an error message appears. The Custom button, from which you can assign a single constant value or a variable assignment, is also available from this dialog box.

## Boolean Value Dialog Box

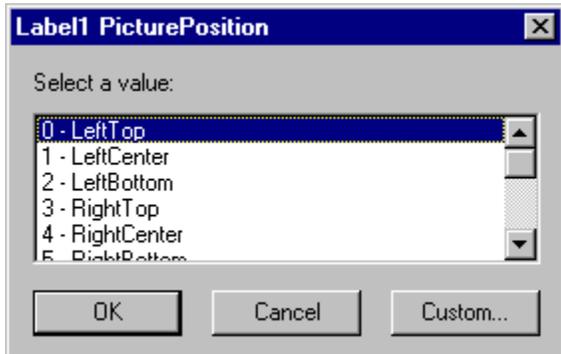
[See Also](#)



This example of a Boolean Value dialog box provides option buttons from which to choose the Boolean values **True** or **False** for the Label1 **Enabled** property. The Custom button, from which you can assign a single constant value or a variable assignment, is also available from this dialog box.

## Enumeration Value Dialog Box

[See Also](#)



This example of an Enumeration Value dialog box provides a list box of integer values from which to choose the Label1 **PicturePosition** property value.

The Custom button, from which you can assign a single constant value or a variable assignment, is also available from this dialog box.

# Glossary

## **A-C**

control

## **D-F**

event

event handler

## **G-M**

global variable

HTML

## **N-R**

property

## **S-Z**

script

shortcut menu

tag

## **tag**

Embedded between angle brackets in HTML text to add character or paragraph formatting to the text. Web browsers display text and graphic elements based on the tags an author uses. The tag itself is not displayed by the browser.

For example, the text

Make `<B>this text</B>` look bold

is displayed this way by a browser:

Make **this text** look bold

**control**

An ActiveX Control, Java applet, plug-in, PowerPoint animation, or form field that you have inserted in your page.

Controls have their own set of recognized properties and events. You use controls to receive user input, display output, and trigger event procedures. You can manipulate most controls using methods.

**event**

An occurrence, often initiated by the user, to which a program can respond. A key press, button push, and mouse movement are typical events.

**global variable**

A variable whose value can be accessed and modified by any event handler on a Web site.

**property**

An attribute of an object that you set to define one of the object's characteristics (such as size or color) or an aspect of its behavior (such as whether it is hidden).

**script**

In FrontPage terminology, code written in VBScript or JavaScript. Inserted in a page, a script consists of a set of instructions that connect events with actions.

**HTML**

Hypertext Markup Language. A system of marking up, or tagging, a document so it can be published on the World Wide Web. Documents prepared in HTML include reference graphics and formatting tags. You use a Web browser (such as Microsoft Internet Explorer) to view these documents.

**shortcut menu**

A list of commands that is displayed when you click the right mouse button. Shortcut menus provide quick access to frequently used commands that are also available from the main menu bar. The commands listed depend on the object you click.

**event handler**

Code that is executed when a particular event occurs. The following example shows the event handler for the **Click** event associated with **CommandButton1**. The event handler sets the **Caption** property of **BannerLabel**.

```
Sub CommandButton1_Click ()  
BannerLabel.Caption = "New Caption"  
End Sub
```

## Bring to Front Action

[See Also](#)

The **Bring to Front** action positions a control within an HTML Layout on top of overlapping controls.

### List View format

When selecting the **Bring to Front** action, the name of the control you have selected appears under the Object category in the Script pane and the **Bring to Front** action appears under the Action category.

### Code View format

When selecting the **Bring to Front** action in Code View, an action appears in the Script pane with the following command:

<b>VBScript</b>	<b>JavaScript</b>
<code>call <i>object</i>.ZOrder(0)</code>	<code><i>object</i>.ZOrder(0)</code>

## Go To Page Action

The **Go To Page** action causes the browser to navigate to a specified Web location. **Go To Page** appears as the first action in the Action pane, at the highest level, and is available when editing any file.

### List View format

When selecting the **Go To Page** action, the Text String Value dialog box appears. You can type in a URL or customized variable name and select **OK**. No object is shown under the Object category in the Script pane, and the **Go To Page** action appears under the Action category, along with the URL assignment in quotation marks.

### Code View format

When selecting the **Go To Page** action in Code View, an action appears in the Script pane in the following format with empty quotation marks:

```
Window.location.href = ""
```

You can type in a URL within the quotation marks.

## Hide Control Action

[See Also](#)

The **Hide Control** action makes a control invisible.

### List View format

When selecting the **Hide Control** action, the name of the control you have selected appears under the Object category in the Script pane and the **Hide Control** action appears under the Action category.

### Code View format

When selecting the **Hide Control** action in Code View, an action appears in the Script pane with the following Boolean assignment:

<b>VBScript</b>	<b>JavaScript</b>
<code>object.Visible = False</code>	<code>object.Visible = false</code>

## Send to Back Action

[See Also](#)

The **Send to Back** action positions a control within an HTML Layout behind overlapping controls.

### List View format

When selecting the **Send to Back** action, the name of the control you have selected appears under the Object category in the Script pane and the **Send to Back** action appears under the Action category.

### Code View format

When using VBScript and selecting the **Send to Back** action in Code View, an action appears in the Script pane with the following command:

<b>VBScript</b>	<b>JavaScript</b>
<code>call object.ZOrder (1)</code>	<code>object.ZOrder (1)</code>

## Show Control Action

[See Also](#)

The **Show Control** action makes a control visible.

### List View format

When selecting the **Show Control** action, the name of the control you have selected appears under the Object category in the Script pane and the **Show Control** action appears under the Action category.

### Code View format

When using VBScript and selecting the **Show Control** action in Code View, an action appears in the Script pane with the following Boolean assignment:

<b>VBScript</b>	<b>JavaScript</b>
<code>object.Visible = True</code>	<code>object.Visible = true</code>

## Choose List View or Code View

### See Also

The Script Wizard has two views: List view and Code view. You can switch between views by clicking the appropriate option at the bottom of the Script Wizard window.

- Use **List view** to assign simple actions to events, or to set property values, using a “point-and-click” approach.
- Use **Code view** to write scripts directly in VBScript or JavaScript — when you want to use the power and full support of the scripting language.

## Start the Script Wizard

See Also

- 1 Select the ActiveX control or form field you want to script, click the right mouse button, then select the **Script Properties**.
- 2 In the **Script** dialog box, click **Script Wizard**.

## Event and Action Pane Icons

[See Also](#)

In the Script Wizard, each item in the **Event** and **Action panes** is preceded by an icon. The following table describes each of these icons.

<b>Icon</b>	<b>Description</b>
	An event that hasn't been scripted.
	An event that has been scripted.
	An action.
	A <u>property</u> .
	An object.
	The window object.

# Create Event Handlers

## See Also

- 1 In the **Event pane** of the Script Wizard, click the Plus sign (+) next to the object you want to script. The hierarchy expands to display all of the events you can script for that specific object. ◆ indicates scripted events;
- ◆ indicates events that haven't been scripted.
- 2 Click the event you want to script. For example, click **MouseOver** to specify what happens when the user moves the mouse pointer over the object.
- 3 In the **Action pane**, double-click the action, property, procedure, or variable you want to add to the event handler. For example, double-click **BackColor** to specify a background color that will change when the user moves the mouse pointer over the object.
  - If you're working in List view and you double-click a property, you'll see a dialog box asking you to choose a value for the property. The type of dialog box you'll see, and the values you can select, depend on the type of property you've double-clicked.
  - If you're working in Code view and you double-click a property or variable, edit the script in the **Script pane** and type a value for the property or variable.

## Delete Event Handlers

See Also

In the **Event pane** of the Script Wizard, click the Plus sign (+) next to the object that contains the event handler you want to delete.

The hierarchy expands to display all of the events for that specific object. ♦ indicates events that have been scripted.

- 1 Click the event.
- 2 Using the right mouse button, click to display the shortcut menu, then click **Delete Event Handler**.

## Define Global Variables

[See Also](#)

- 1 With the pointer in the **Action pane** of the Script Wizard, using the right mouse button, click to display the [shortcut menu](#), then click **New Global Variable**.
- 2 In the **New Global Variable** dialog box, type the name of the [global variable](#) you want to add to the page.

You can include subscripts in VBScript or an initial value in JavaScript, in the proper syntax for that scripting language.

When you click **OK**, the Script Wizard inserts that variable at the beginning of your page in the form **dim** *variable-name* for VBScript or **var** *variable-name* for JavaScript.

## Edit or Delete Global Variables

### See Also

- 1 In the **Action pane** of the Script Wizard, select the global variable you want to edit or delete.  
**Note** In JavaScript, only variables defined with the **var** statement are considered global variables.
- 2 Using the right mouse button, click to display the shortcut menu, then click **Edit** or **Delete**.  
**Note** You can't delete a global variable if it's part of a multivariable **dim** or **var** statement.

## Define Procedures

See Also

- 1 With the pointer in the **Action pane** of the Script Wizard, using the right mouse button, click to display the shortcut menu, then click **New Procedure**.
- 2 In the **Script pane**, edit the procedure:
  - If you're working in **List view**, in the **Action pane** double-click the action or property to add to the procedure.
  - If you're working in **Code view**, in the **Script pane** edit the script for the new procedure. The Script Wizard automatically adds an **end sub** to the end of the procedure.

When you click **OK**, the Script Wizard inserts the procedure at the beginning of your page.

**Note** The Script Wizard doesn't check for duplicate procedures when reading an existing page that already contains procedures. Therefore, when you're adding a new procedure to an existing page, you should verify that a procedure with the same name doesn't already exist.

## Specify the Scripting language

[See Also](#)

The Script Wizard lets you create scripts in either VBScript or JavaScript. However, you can use only one scripting language per page, and you can't change the scripting language for a page that already contains script.

### ► To specify the scripting language

- 1 Choose **Insert: Script**.
- 2 In the **Script** dialog box, select VBScript or JavaScript.
- 3 Click **OK**.

## Specify the Script Pane Font

See Also

- 1 With the pointer in the **Script pane**, in Code view, using the right mouse button, click to display the shortcut menu, then click **Font**.
- 2 Select the font you want to use in the **Script pane**.

## Script a Jump to Another Page

See Also

Use this procedure to script an event that jumps to another page.

► **To script a jump to another page**

- 1 In the **Event pane** of the Script Wizard, click the Plus sign (+) next to the object you want to script. The hierarchy expands to display all of the events for that specific object.
- 2 Click the event from which you want to create a jump. For example, click **Click** to jump to another page when the user clicks the object.
- 3 In the **Action pane**, double-click the **Go To Page...** action.
  - If you're working in List view, in the **Go To Page** dialog box, type the URL, without beginning or ending quotation marks, for the jump destination page, then click **OK**.
  - If you're working in Code view, in the **Script pane**, type the URL of the jump destination page for the **window.location.href = ""** property.

## Hide or Show a Control

[See Also](#)

Use this procedure to [script](#) an [event](#) that hides or shows a [control](#).

► **To hide or show a control**

- 1 In the **Event pane** of the Script Wizard, click the Plus sign (+) next to the object that contains the event you want to script. The hierarchy expands to display all of the events for that specific object.
- 2 Click the event that will initiate the hide or show action. For example, click **Click** to hide or show a control when the user clicks the object in the **Event pane**.
- 3 In the **Action pane**, click the Plus sign (+) next to the control that you want to hide or show. The hierarchy expands to display all of the actions and [properties](#) for that specific control.
- 4 Double-click the **Hide Control** or **Show Control** action as appropriate.

# Script Wizard Technical Notes

[See Also](#)

## Differences between List view and Code view

In **List view**, you can create [event handlers](#) that use these simple actions:

- Any method invocation on any object or procedure that takes no [arguments](#).
- Any method invocation on any object or procedure that takes the same number and names of arguments as the event handler.
- The **Go To Page**, **Hide** or **Show Control**, or **Bring to Front/Send to Back** actions.
- Any assignment to an object [property](#) or [global variable](#).

In **Code view**, on the other hand, you have full support for the default scripting language, including control flow.

## Scriptable HTML elements

Scripting of [HTML](#) `<A HREF="...">` or `<FRAMESET>` [tags](#) isn't supported.

## <INPUT> elements with common NAME=

Option buttons and other `<INPUT>` elements that share the same `NAME=` are not differentiated from one another in the Script Wizard **Event pane**.

For example, in the case `<INPUT TYPE=RADIO>`, each item is displayed in the following way:

```
NAME 'VALUE'
```

So if you have three option buttons, all named **Color**, but each having a distinct value, they might appear as:

- Color 'Blue'
- Color 'Red'
- Color 'Green'

## Reading HTML

When you start the Script Wizard, it reads your page and searches for any occurrences of script. The first language found is established as the default scripting language for that page.

If the Script Wizard finds more than one variable on a **dim** or **var** line, that variable can't be deleted or modified.

The Script Wizard does not support the `SRC=` attribute of the HTML `<SCRIPT>` tag. If your HTML page includes such a tag, you'll see a warning message when you start the Script Wizard.

The Script Wizard doesn't check for duplicate procedures or object names when reading a page that already contains procedures or objects. Therefore, when you're adding a new procedure or object to a page, you should verify that a procedure or object with the same name doesn't already exist.

## Writing HTML

When you close the Script Wizard, existing scripts are saved back into your page in the same location.

All new external event handlers are placed before, and as near as possible to, the object within the `<BODY>` tags of the HTML file.

Any global variables or user-defined procedures will appear in `<SCRIPT>` tags within the `<HEAD>`

section of the page. The Script Wizard automatically adds an **end sub** to the end of any procedure.

If the Script Wizard inserts a new `<SCRIPT>` tag before any others in the HTML file, the new `<SCRIPT>` tag will specify the default scripting language.

### **Quotation marks in scripts**

External **List view** scripts use double quotes ("), while internal **List view** scripts use single quotes (').

## **URL**

Uniform Resource Locator. Identifies the full path of a document, graphic, or other file on the Internet or on an intranet.

A URL expresses the protocol (such as FTP or HTTP) to be accessed and the file's location. A URL may also specify an Internet e-mail address or a newsgroup. Some examples of URLs are:

`http://www.someones.homepage/default.html`

`ftp://ftp.server.somewhere/ftp.file`

`gopher://server.name`

`file://Server/Share/File.doc`

**argument**

A constant, variable, or expression that supplies information to an action, procedure, or method.

For example, in the Beforeupdate event syntax,

**Private Sub *object*\_BeforeUpdate( *Cancel* As Boolean)**

*Cancel* is an argument

