

Getting Started with the Microsoft[®] ActiveX[™] Control Pad

Creating a “Hello World” Sample Web Application

Draft Document
Last Revised 6/14/96

Table of Contents

Building your first ActiveX Web Site with the ActiveX Control Pad.....	
Overview and Technical Support.....	
Introduction to ActiveX.....	
Using ActiveX Controls.....	
ActiveX Controls Design-Time Concept.....	
Methods, Properties and Events.....	
Getting Started with the ActiveX Control Pad.....	
Launch the ActiveX Control Pad.....	
Adding Your First ActiveX Control to a Web Page.....	
Adding a Second Control.....	
Introduction to Scripting: Making the Page Interactive.....	
VB Script vs. Visual Basic and Visual Basic for Applications (VBA).....	
Adding Scripting to Your Web Page.....	
Production.....	
Extending Your Web Pages with Third Party or Custom-built ActiveX Controls.....	
ActiveX Controls and Code Safety.....	
Positioning Controls on a Web Page.....	
Advanced ActiveX Pages.....	
The Microsoft HTML Layout Control.....	
Creating 2D HTML Designs using the Microsoft HTML Layout Control and ActiveX Control Pad.....	

Building your first ActiveX Web Site with the ActiveX Control Pad

Overview and Technical Support

This document will assist developers and multimedia designers in creating their first ActiveX™ Web site using the Microsoft® ActiveX Control Pad. This tutorial assumes a basic understanding of HTML documents. A basic knowledge of scripting languages, such as Visual Basic® Scripting Edition (VB Script) or JavaScript™, is helpful but not required.

If you require assistance using the ActiveX Control Pad, or have technical problems with the product, you can access the beta support forum on the Internet. Please visit <http://www.microsoft.com/intdev/author/cpad/> for information on how to access the support forum, and for the latest product information, downloads, samples and updates. Before beginning, you will need to download the Internet Explorer 3.0 beta release and the ActiveX Control Pad beta release from the Microsoft Web site (start at the URL listed above for complete download and setup instructions).

Introduction to ActiveX

ActiveX is a set of open technologies which bring the power of the personal computer to the ubiquitous connectivity of the Internet. ActiveX takes the Internet beyond static text and picture documents to provide users with a new generation of more active, exciting, and useful experiences. For Intranet developers (Intranets are private Web sites published on internal, corporate networks), ActiveX provides core functionality for building robust enterprise-wide applications that offer enhanced functionality and productivity beyond basic HTML document sharing.

Core ActiveX technologies include:

- **Leading-edge support for HTML.** At its heart, ActiveX is about providing the highest level of support for the latest HTML standards used on the Web, such as Cascading Style Sheets, tables, and rich frameset support as defined by the Worldwide Web Consortium (W3C). ActiveX also incorporates the other core Internet standards such as HTTP and NNTP network protocols.
- **ActiveX Controls.** ActiveX Controls are re-usable software components created by a variety of software vendors that can be used to quickly add specialized functionality to Web sites, desktop applications, and development tools. For example, a stock ticker control could be used to add a live stock ticker to a Web page, or an animation control could be used to add animation features to a Web page.
- **ActiveX Scripting.** Scripting allows multiple controls to be fully integrated into complete applications. Scripting refers to the use of a high-level programming language, such as Visual Basic Script (VB Script) or JavaScript to add programming logic to automate and integrate the behaviors of controls, documents, Web browsers, and Web servers. For example, scripting can be used to cause an audio clip to play in an audio control when the user pushes a push button control. Scripting might also be used to calculate the total price of an electronic commerce transaction, and display the result in a text window control on a Web page.
- **ActiveX Documents.** Web browsers typically display Web pages created using the Hypertext Markup Language (HTML). ActiveX Documents allow the Web browser to also browse content defined in other formats. For example, the Internet Explorer 3.0 can browse Visio drawings saved in the Visio format, documents saved by Microsoft Office applications such as Microsoft

Word, Microsoft Excel, and Microsoft PowerPoint, as well as documents saved in a variety of other application-specific formats. Applications supporting ActiveX Documents can have their documents be activated, browsed, and edited directly within the Internet Explorer in this manner, making it easier for users to find and use a wide variety of published information on the Web and corporate Intranets.

- **Seamless Multimedia.** ActiveX supports leading-edge multimedia standards, like Open Type fonts, a wide variety of graphic formats, and media formats for playback of audio, video, and animations over low-bandwidth connections.

To assist developers in developing Web pages using ActiveX technologies, Microsoft recently announced a new authoring utility, the Microsoft ActiveX Control Pad, which is being provided to all users, free-of-charge. The ActiveX Control Pad makes it very easy to add ActiveX Controls and ActiveX Scripting (either VB Script or JavaScript) to any HTML document.

Using ActiveX Controls

Before beginning to use the ActiveX Control Pad, a short introduction to the basic concepts and terminology of ActiveX Controls is a good idea. ActiveX Controls are a superset of the “OLE Control (OCX)” and Java applet component technologies. In this beta version of the ActiveX Control Pad, Java applets are not supported, but they will be in the final product.

There are over 2,000 commercially available ActiveX Controls. ActiveX Controls can be created using a variety of programming languages such as C, C++, the next version of Visual Basic, and Java. Once created, ActiveX Controls can be used by designers and developers as pre-fabricated components to quickly create custom applications. Using ActiveX Controls in such a manner does not require knowledge of how the component was created, and in many cases requires no programming whatsoever. The Internet Explorer 3.0 Web browser is the first Web browser to support ActiveX Controls in Web pages. ActiveX Controls can also be viewed in the Netscape Navigator using the ActiveX Plugin for Netscape (see <http://www.ncompass.com>).

The ActiveX Control Pad includes a number of useful controls that you can use right away. Controls provided by the ActiveX Control Pad include:

Control Name	Function
Microsoft Forms 2.0 Label	basic text labels
Microsoft Forms 2.0 Textbox	multiline text entry and text display window
Microsoft Forms 2.0 Combo Box	allows users to choose from a drop-down list of options
Microsoft Forms 2.0 List Box	allows users to choose from a scrollable list of options
Microsoft Forms 2.0 CheckBox	allows user to check an option
Microsoft Forms 2.0 Option Button	allows users to choose between multiple options
Microsoft Forms 2.0 Toggle Button	a button that has a toggle state (for example, on/off)
Microsoft Forms 2.0 Command Button	a basic pushbutton control
Microsoft Forms 2.0 Tabstrip	provides multiple pages choosable via tabs
Microsoft Forms 2.0 ScrollBar	basic horizontal and vertical scroll bars
Microsoft Forms 2.0 Spin Button	a button that can be pushed “up” or “down”
The Microsoft Image Control	a control that can display progressively rendered images in metafile, JPG, GIF, BMP or wavelett formats
The Microsoft Hotspot Control	a transparent control that can be used to create clickable regions on a page

The Microsoft Web Browser Control (Shell Explorer Control, included with Internet Explorer 3.0)	a control that can display and browse ActiveX Documents including any HTML document as well as any other Active Document type (Word documents, Excel spreadsheets, etc.)
---	--

Additional controls are provided by various development tools and from a large and growing third party market of control developers. For example, MacroMedia® has a Shockwave for Director Control that allows designers to incorporate Shockwave Movies (animations) into a Web page; RealAudio has a RealAudio Control that allows designers to incorporate RealAudio streaming sound into a Web page; and Adobe™ has an Acrobat™ Control that allows Acrobat documents to be viewed over the Web. There are over 2,000 commercially available controls that provide functionality ranging from real-time stock tickers to database connectivity to multimedia authoring. Visit the Internet Explorer web site for more information on available controls (<http://www.microsoft.com/ie/controls>).

ActiveX Controls Design-Time Concept

ActiveX Controls support two modes of operation: **runtime** and **design time**. At runtime, users can see the control (unless it has been made invisible by the author) and use it in the context of a Web page or other application. During runtime, users cannot take the control out of the Web page to modify it and reuse it in another Web page or application. This protects the software license rights for software vendors providing ActiveX Controls.

Designers and developers can use controls in design time mode if they have the appropriate license for the control (for example, they have purchased the control, or downloaded a fully licensed control over the Web). Using the design-time behavior of the control, designers and developers can now incorporate an ActiveX Control into their own Web pages and applications. Because they support a design-time mode, ActiveX Controls provide rich benefits as pre-fabricated components to quickly build up feature-rich applications. At design time, controls can be visually modified to fit the needs of the designer or developer without requiring programming, and then be redistributed in a Web page over the Internet, or within the context of an application distributed on CD, diskette or online.

Methods, Properties and Events

Controls have **properties**, such as a size, a background color, a foreground color, font, etc. that can be easily set via a visual **property table**. The properties of any control can be modified at design time without requiring any programming. The available properties of any given control are determined by the software programmer that originally created the control, and depend on the functionality provided by the control. For example, a video control would likely have a video filename property (specifying the name of the video file to play), and a size property, among others. By changing the properties of a control at design time, the control is easily modified to fit the specific needs of designer or developer as they integrate it into their Web pages.

Controls also have **methods**---functions that a control supports. For example, a video control would likely have stop, play and rewind methods. Finally, controls have **events**---these are notifications that the control makes during runtime to other controls and applications that may want to take an action based on the event. For example, a pushbutton control has a “click” event that occurs when a user pushes it. A video control might have a “done playing” event that occurs when it finished playing a video clip.

Getting Started with the ActiveX Control Pad

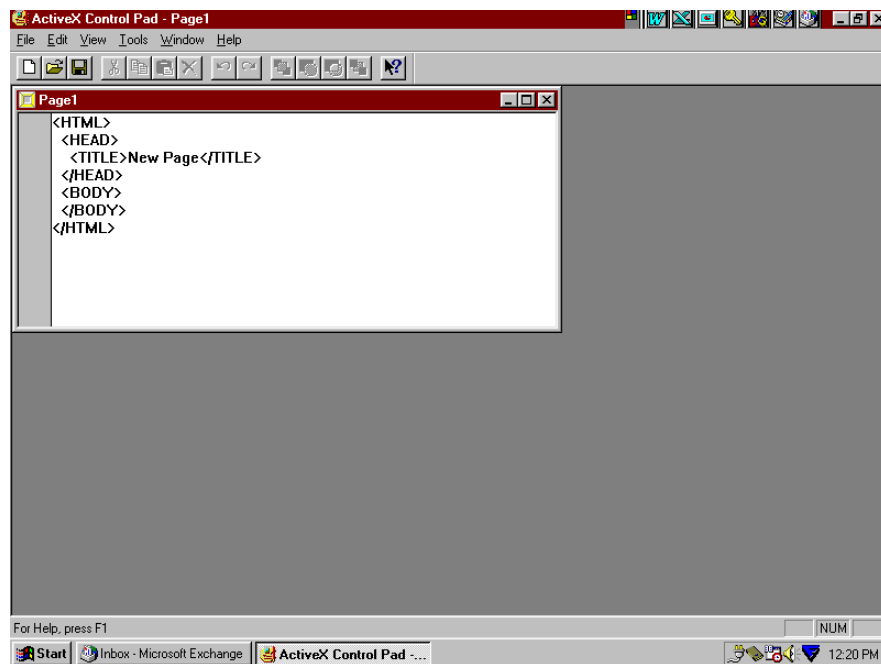
To begin using this tool, you must first download and install the Internet Explorer 3.0 Web browser (<http://www.microsoft.com/ie>) and the ActiveX Control Pad (<http://www.microsoft.com/intdev/author/cpad>). The beta release of the ActiveX Control Pad runs on

Windows 95 and the Beta2 release of Windows NT 4.0. Additional information on downloading and installing the software is available on the Microsoft Internet Explorer site, and on the Internet Developer Toolbox site (<http://microsoft.com/intdev>). When you finish installing Internet Explorer 3.0, reboot your system and then install the ActiveX Control Pad. Read the README file that is installed with the ActiveX Control Pad by using the Windows 95 start button. The ActiveX Control Pad Readme and application will be found in their own program group using the Start button. You should review the README for known issues with the beta release.

Launch the ActiveX Control Pad

Launch the ActiveX Control Pad from the Windows 95 Start button. The ActiveX Control Pad will open showing a new, blank HTML page in an HTML source code Text Editor.

The ActiveX Control Pad Text Editor



Upon opening the ActiveX Control Pad, a blank HTML document will appear in the Text Editor.

While the ActiveX Control Pad does not itself provide WYSIWYG HTML authoring support, you can use the Control Pad in conjunction with such tools, such as the Internet Assistant for Microsoft Word (<http://microsoft.com/msword>), Microsoft Front Page (<http://microsoft.com/frontpage>), HotMetal Pro, etc. You can also type native HTML source code directly into the Control Pad Text Editor, or copy the source HTML code from an existing Web page.

When you open (via the File/Open menu choice) an HTML document in the ActiveX Control Pad, you will see the HTML source code for that document. For the Hello World example used in this tutorial, the complete HTML source code for the Web page is:

```
<HTML>

<HEAD>

<TITLE>Untitled</TITLE>

</HEAD>
```

```
<BODY>

<P>
<CENTER><B><FONT SIZE=2>My First Active Web Page Says "Hello
World"<BR>
<BR>
</FONT></B></CENTER>
</BODY>

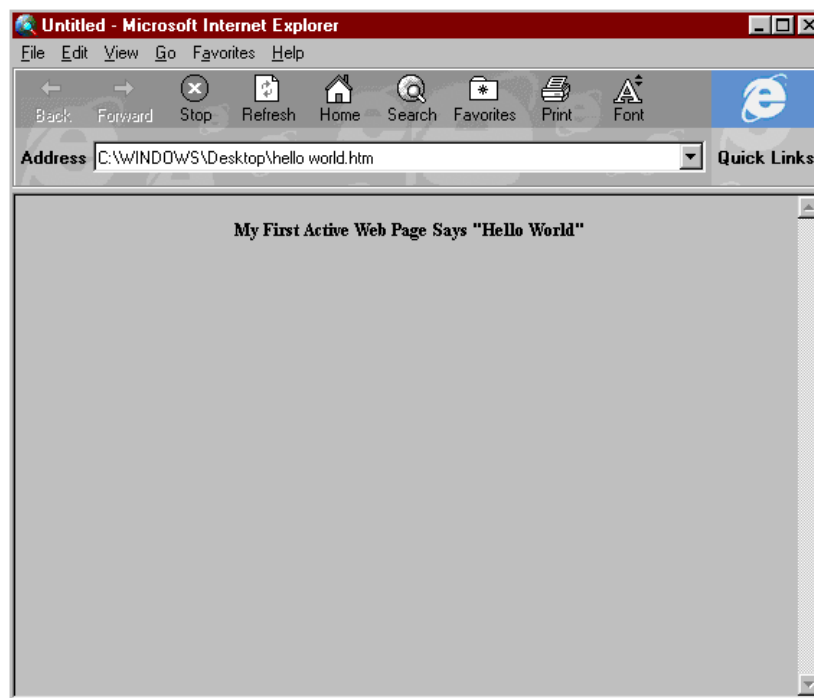
</HTML>
```

Edit your blank HTML document so that it looks exactly like this. Use the File/Save menu option to save the page, and name it "Hello.htm." You can save it to any directory, by default it will save into the "\program files\activeX Control Pad" directory.

Now open the Hello.htm Web page in the Internet Explorer by double clicking it from the appropriate folder on your hard drive, or by using the Internet Explorer's file/open menu option.

Its a very simple page, which is now ready to be activated with ActiveX Controls.

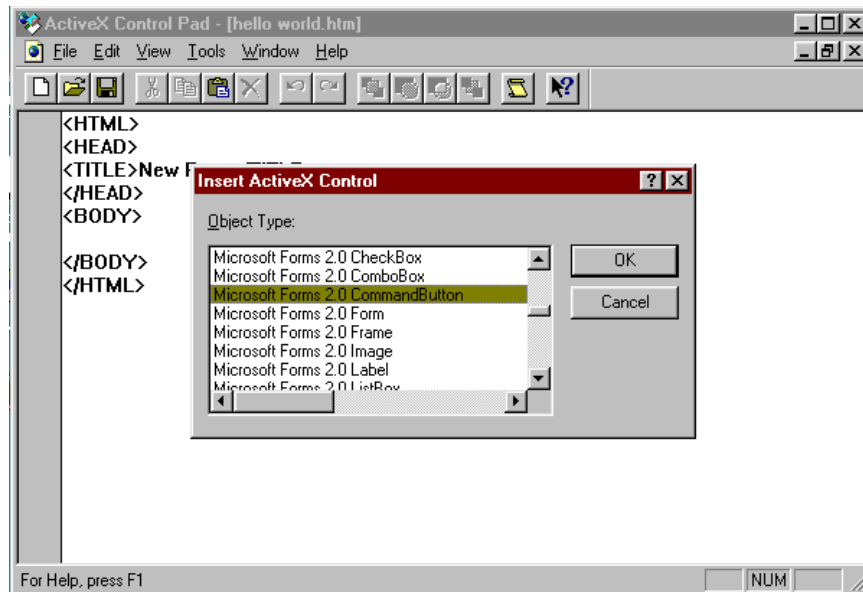
The Hello World Page in Internet Explorer 3.0



Adding Your First ActiveX Control to a Web Page

The next step is to add an ActiveX Control into the HTML document. To insert a control, go back to the ActiveX Control Pad and place your cursor in the HTML source code where you want your control to appear (in this case, just before the </BODY> tag). Then select the Edit\Insert ActiveX Control menu option. A dialog will appear showing all the ActiveX Controls on your system (many of which were installed with the Control Pad, listed above). Choose the "Microsoft Forms 2.0 Command Button" control.

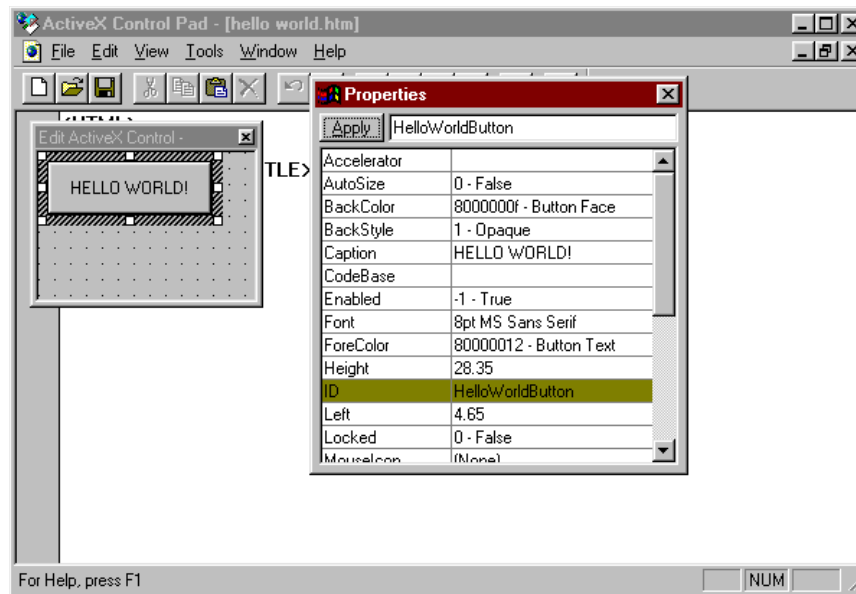
Inserting an ActiveX Control onto a Web page



After choosing Edit/Insert ActiveX Control, a dialog appears showing all the available controls on your system. This dialog will include Microsoft and third-party controls, such as the RealAudio Controls or MacroMedia Shockwave for Director Controls. There are over 2000 commercially available ActiveX Controls. To find out about available controls, visit <http://www.microsoft.com/ie/controls>

When you insert a control, the ActiveX Control Pad brings up a window, the Object Editor, with the control in it, as well as a Properties Window (the control's property table). You can now set properties on the control visually. You can resize the control inside the Object Editor, and you can also resize the Object Editor window if you need more space to work in. You can also type text directly into the control after clicking it once. Type "Hello World" into the button. Use the Properties Window to set other properties on the Command Button. For this example, set the ID property to "HelloWorldButton" (no spaces).

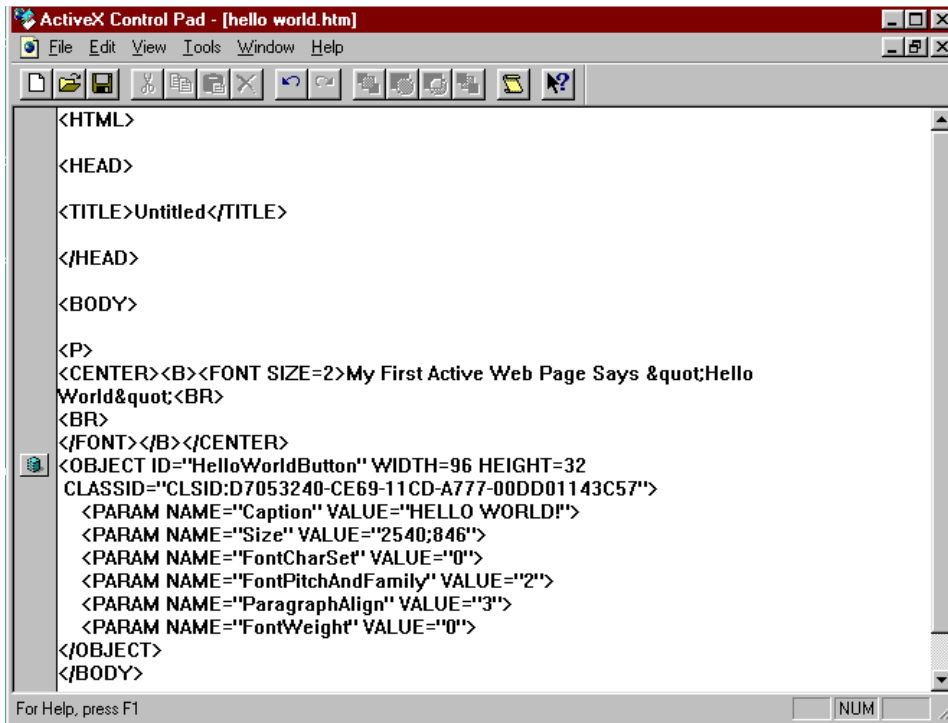
The Object Editor window



After choosing an ActiveX Control to insert into the HTML page, the control will be shown in design mode inside the Object Editor. Also, the control's properties can be visually set using the Properties Window.

When you are done setting properties (beta note: close the window with the control in it **before** closing the Properties Window to save your changes to the control), the ActiveX Control Pad automatically fills in all of the HTML syntax necessary to bring the control into the Web page and display it. This information is based on the W3C HTML <OBJECT> tag which uniquely identifies the control in the HTML page and sets its properties using <PARAM> tags based on your editing session in the Object Editor. To bring the Object Editor back at a later time to further modify property settings, choose the Edit/Edit ActiveX Control menu option or click on the blue cube icon next to the control's <OBJECT> tag. The cube icons that appear in the left pane of the Text Editor are a convenient way to track multiple controls on any given page. Because the ActiveX Control Pad automatically generates all of the required HTML syntax, you will find that you can add any number of ActiveX Controls to an HTML page in a matter of seconds.

The Hello World HTML Page with the CommandButton Control inserted

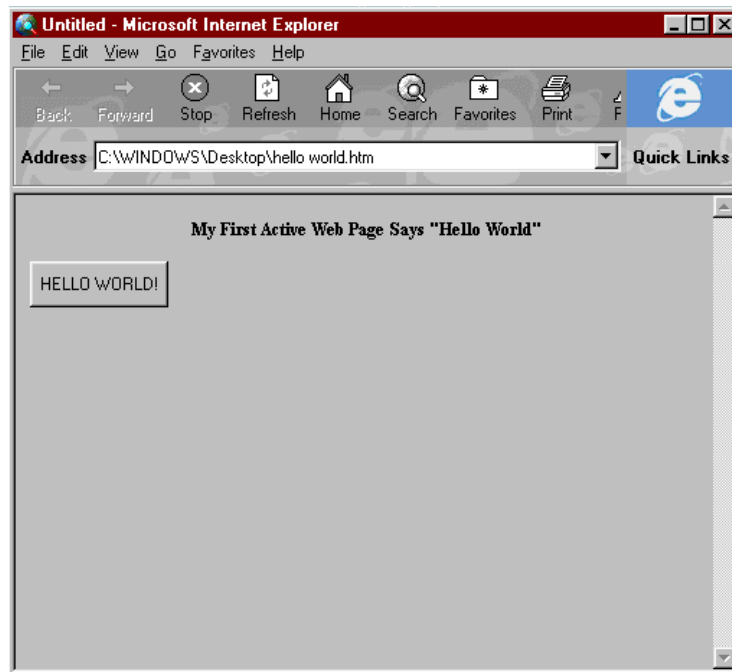


This picture illustrates the HTML syntax automatically generated after closing the Object Editor. The HTML syntax enclosed by the <OBJECT> tags includes all the information necessary to display the ActiveX Control in the Web Page based on properties set in the Object Editor. The cube icon on the left can be used to easily track controls on a page, and quickly bring them up for further editing.

Next, save your HTML file from the Text Editor, and now open it again in Internet Explorer 3.0 (if you double click on the file, Internet Explorer will open and display the page, if the old version of the page is already running inside Internet Explorer, simply hit the “refresh” button on the Internet Explorer toolbar).

You should now see the command button control embedded on the Web page, and while it “works” (you are able to click the button), nothing will happen yet.

The Hello World Web page running inside Internet Explorer 3.0



Adding a Second Control

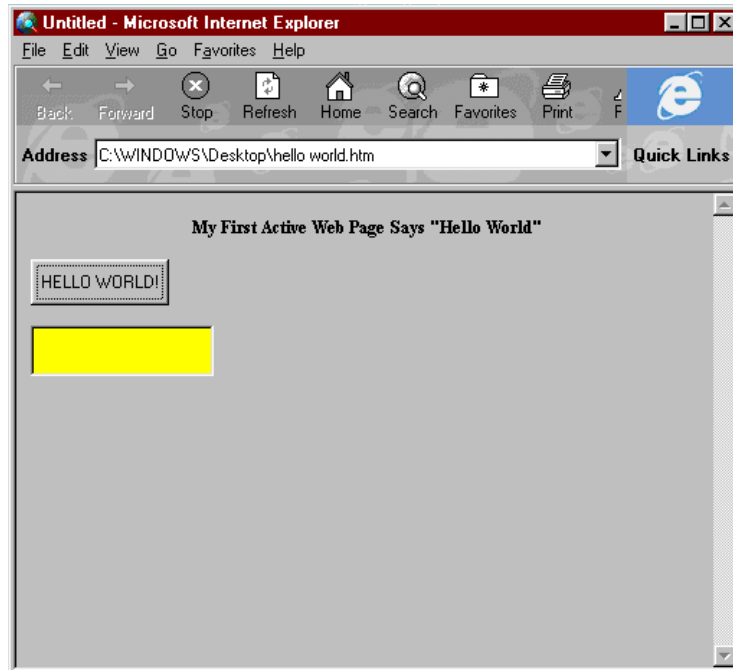
While the ability to add a single control can be very useful in constructing Web pages, the real power of ActiveX is the ability to include multiple controls and then integrate their behaviors using scripting. In this way, rich applications can be built within Web pages. To illustrate this, we will add a second control to the “Hello World” Web page.

Go back to the Control Pad, and if necessary reopen your “Hello.htm” document. Position the cursor on the line following the `</Object>` tag, and insert two return characters by typing (`
`) on two consecutive lines. Now insert a second control by selecting the Edit\Insert ActiveX Control menu option. Again, the dialog will appear showing all the ActiveX Controls on your system. Choose the “Microsoft Forms 2.0 TextBox.”

The Object Editor will appear with the textbox in design mode. Do not type anything into the textbox, but for practice, use the Property Window to change the background color to yellow. To do this click once on the “BackColor” row in the property table. Now, at the top of the property table, click the ellipse “...” button, and a color picker will appear. Choose bright yellow. Now click “OK” to dismiss the color picker. Now click “Apply” to apply the new background color.

Now close the Object Editor Window. The information for bringing that control into the Web page will be automatically written into the HTML document. Save the HTML document. Now reopen the “Hello.htm” document in Internet Explorer, to see what it will look like to a user. You should see the pushbutton control and the text window with a yellow background.

The Hello World Web Page with two ActiveX Controls



Introduction to Scripting: Making the Page Interactive

Now you have an HTML document with two embedded controls. The next step is to use the Script Wizard to very easily integrate the two controls and provide interactivity to your user. The Script Wizard can work with either Visual Basic Script (VB Script) or JavaScript. You can set your language choice using the \Tools\Options\Script menu, but for now leave it at the default, VB Script. With either language, developers can capture events from ActiveX Controls, invoke methods, modify properties of ActiveX Controls, and more. Because the scripting code is very simple to write and is always maintained in source code form in the Web page itself, it is very easy to modify. Using the Script Wizard is the best way to coordinate activity across ActiveX Controls written in any development language like Java, C/C++, Basic etc. The power of ActiveX is that any ActiveX Control, written in *any* programming language and supplied by *any* software vendor, can be wired together with any other ActiveX Control. This is called *component software*, because it provides the ability to use ActiveX Controls as pre-fabricated building blocks for quickly constructing feature-rich applications.

VB Script vs. Visual Basic and Visual Basic for Applications (VBA)

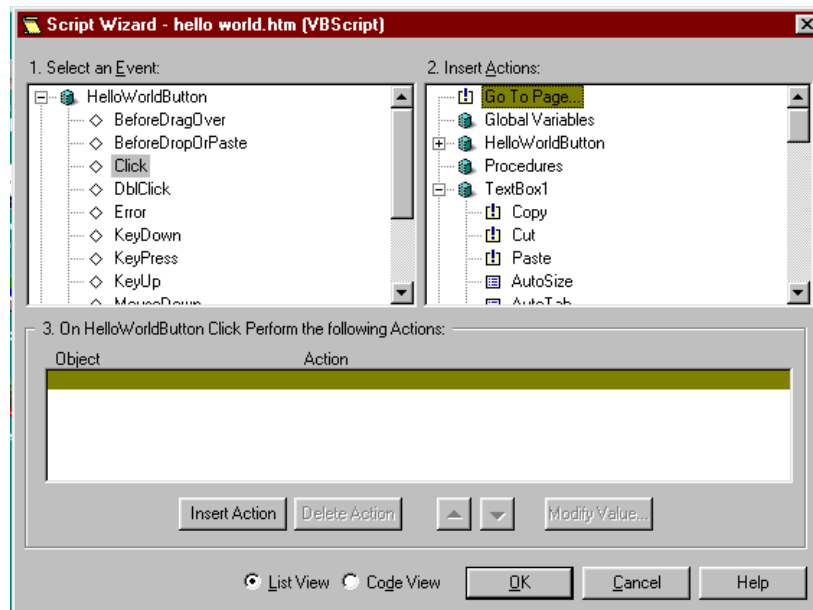
Compared to the Visual Basic or Visual Basic for Applications development environments, VB Script is a subset of the Visual Basic syntax, it is “safe” in terms of restricting developers to commands that cannot directly access the system. It is also portable across platforms, and can be redistributed royalty free (the interpreter is posted on the Web). Microsoft is supplying VB Script on Windows and the Macintosh, other software vendors are making it available most other platforms such as UNIX. VB Script is very lightweight, and optimized for the Web. In addition, although VB Script does not yet provide an advanced debugging environment like the other versions of Visual Basic, the Script Wizard makes adding code as easy as point and click. For more information on VB Script, see the VB Script Web site at <http://microsoft.com/vbscript>.

Adding Scripting to Your Web Page

Go back to the Control Pad with your open “Hello World” Web Page. To add interactivity between the two ActiveX Controls in your HTML document, click on the Script Wizard toolbar icon or choose the Tools/Script Wizard menu option.

In the Script Wizard, you will see the upper left pane shows all of the controls you currently have on your page. The left pane is the “event” pane—click the plus sign next to the HelloWorldButton control to see all of the **events** this control supports. The right pane is the “action” pane. It also shows all of the controls currently on your Web page. But instead showing events, it shows the controls’ **methods** and **properties** that can be manipulated using scripting. As you build scripts in the Script Wizard, the bottom pane will show the actions you are assigning. Note that the bottom pane can display the actions in either List View (as a statement) or Code View (showing the generated VB Script or JavaScript code).

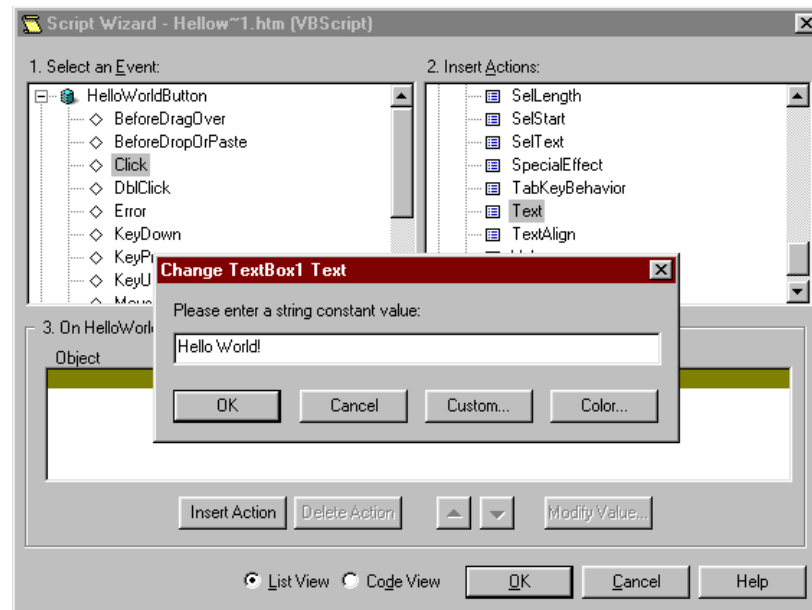
The Script Wizard



The Script Wizard shows events for controls in the left pane, methods and properties in the right pane, and generated actions in the lower pane. Switching to code view will display actual lines of VB Script or JavaScript code generated by the Script Wizard. Developers and Designers can thus script their Web pages just by pointing and clicking to wire different controls together. Developers can modify the generated script code, and code custom lines of scripting in the bottom pane if desired.

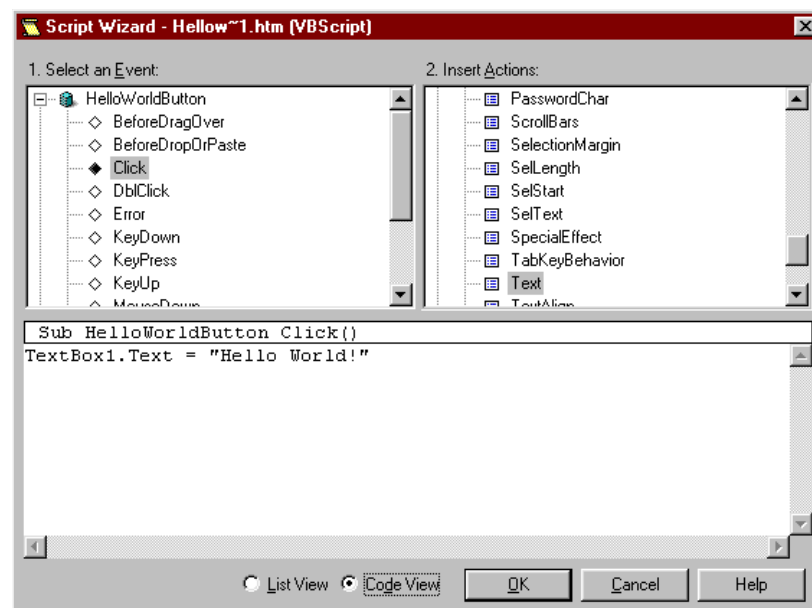
For this Web page, in the Event (left) pane, select the Command Button control, which drops down a list of all its events, and select the “Click” event by clicking once on it. Then in the Action (right) pane, select the TextBox control which drops down a list of its methods and properties. The controls methods are shown with a yellow/! icon, and its properties with a blue/document icon. Double-click the “Text” property in the right pane. A dialog will appear, where you should type “Hello World!”.

Changing the text property via scripting



Choose OK to close the dialog. Notice the action has been inserted into the lower pane. Change to “Code View” to see the code in VB Script form - in this view you can also add additional scripting code, if you wish. For example, you might add if/then loops, variables, or any number of lines of custom script code. For now just close the Script Wizard by choosing “OK.”

The completed script shown in Code View



The lower pane shows the actual lines of VB Script code generated by the Script Wizard. You can add multiple actions per event. Programmers can enter any number of custom lines of VB Script code in the lower pane when viewing in Code View. The Script Wizard can also be configured to generate and work with JavaScript code.

Save the HTML file in the Control Pad, and reopen in Internet Explorer 3.0 (refresh the browser to see the new page). Now, click the button and see what happens! You should be able to click on the Command Button and see “Hello World” appear in the TextBox control. You have now added two controls and scripted them using the Script Wizard.

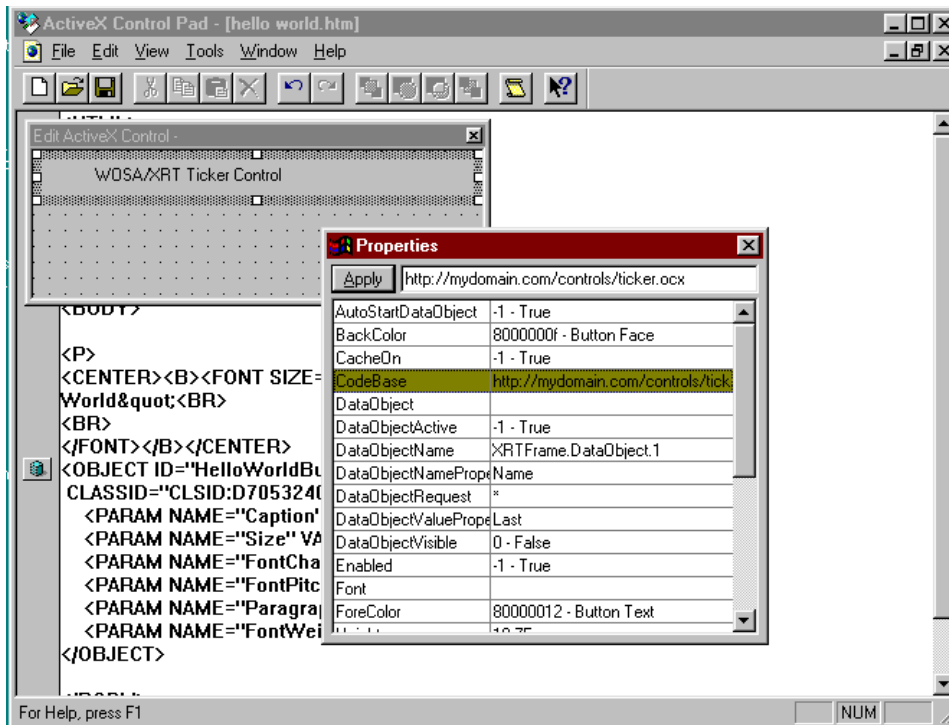
Production

The idea of a “Hello World” application is to show designers and developers how to build a very simple program in the Control Pad. Using your knowledge of HTML, ActiveX Controls, and the Control Pad, you will be able to create interesting, useful, interactive Web sites with ease. To publish the page you just created, you would simply copy the Hello.HTM HTML document to a Web server, and now it can be browsed by users over the Internet!

Extending Your Web Pages with Third Party or Custom-built ActiveX Controls

The ActiveX Controls you just inserted into your Web page will be provided as integrated components of the Internet Explorer 3.0 Web browser. However, there will be many cases where you will want to use a third party or custom-built control that is not part of Internet Explorer. This means you will need to provide information to the Web browser on where to find that control on the Web, so that it can be downloaded to the user’s machine when they browse that page. To do this, you can simply set the “CodeBase” property for the control using the property table that appears with the Object Editor in the Control Pad. You can enter a URL or series of URLs separated by “;”, pointing to a location to look for the control. You will also need to place the control (.OCX file) at that location on a Web server.

Using the CodeBase Property to tell Web browsers how to fetch your control over the Web



The CodeBase property allows you to enter a URL, or series of URLs pointing to search locations where a control can be found and downloaded over the Web. Thus, if a user navigates to a page using a control he/she does not yet have on their system, the browser will be able to find it and dynamically load it over the Web. The ActiveX Control file (typically designated with a “.OCX” extension) must be copied to the proper Web server location.

ActiveX Controls and Code Safety

Microsoft has also provided a mechanism for ActiveX Controls to be digitally signed. In this way, creators of controls can ensure that the controls they distribute carry a “digital certificate” that will identify that they are the verified supplier of that component, and that the control has not been modified or tampered with. Digital certificates protect consumers from potentially harmful components supplied by unknown sources, much like retail software stores provide consumer confidence by carrying software from brand-name sources that can be contacted if there is a problem. If a control is digitally signed, users will see the digital certificate displayed by the Internet Explorer when the control is downloaded over the Web, and they can choose to run the control or not run the control. They can also have any future signed components from that vendor auto-approved so that future messages do not appear when controls from that same vendor are downloaded over the Web. Once downloaded, a control does not need to be re-downloaded the next time they visit a page that uses the same control (even if the control has different properties set on different pages). The Internet Explorer will cache the control on the users system for fast performance.

If a control is not digitally signed, the user will be notified the control is from an unknown source, and the user can choose to either run or not run the control. For more information on dynamically downloading controls over the Web, or for getting controls digitally signed, refer to the Microsoft ActiveX SDK at <http://www.microsoft.com/intdev>. There is also an excellent article on downloading components and code safety entitled “Safe Web Surfing With the Internet Component Download Service” in the July, 1996 issue of the *Microsoft Systems Journal (MSJ)*.

Positioning Controls on a Web Page

The W3C <Object> tag used to insert ActiveX Controls into HTML pages is flexible in that controls can be inserted anywhere within the document. Thus controls might be centered or aligned in the HTML document, placed inside of HTML table cells, or positioned within HTML frames.

Advanced ActiveX Pages

You can move beyond simple HTML, scripting, and ActiveX Controls to include additional technologies such as database access (using dbWeb or IDC files with the Microsoft Internet Information Server), access to Web server processes, ActiveX Documents (such as embedded Microsoft Excel worksheets), and more advanced layout features. To add advanced layout features to a Web page, the ActiveX Control Pad includes the Microsoft HTML Layout Control, a special ActiveX Control that implements new W3C extensions to HTML for frame based layout.

The Microsoft HTML Layout Control

Until recently, Web designers have been limited in their ability to control the placement of features in Web pages, limiting their ability to create sophisticated user interfaces for their Web sites. For example, the current HTML standard does not provide Web designers precise, two dimensional, coordinate control over individual objects placed on a page; nor does it provide the ability to overlap objects and frames to facilitate the creation of more advanced pages with interactive designs.

The HTML Layout Control is a preliminary implementation of the draft specification published by the W3C (<http://www.w3.org/pub/WWW/TR/WD-layout.html>) for 2D-style layout extensions to HTML\CSS. Microsoft has been working closely with the W3C on standards for 2D, frame-based layout capabilities, which provides:

- precise placement, in X, Y, and Z coordinates, of ActiveX Controls on a web page.
- placement that can be changed dynamically, allowing for interactive web pages in which individual objects move or are moved by the user
- overlapping and transparency of ActiveX Controls as defined by the OCX'96 specification
 - scripting of controls with VBScript or JavaScript.

In other words, the HTML Layout Control is itself an ActiveX Control into which you may place additional ActiveX Controls with precise placement and layout support. The HTML Layout Control in turn produces a file (a “.ALX” file) that stores the layout information using the draft W3C extensions to HTML that extend HTML style sheets to incorporate 2D (frame-based) layout. The HTML Layout Control reads this “.ALX” file when the page is viewed. The ALX file is a simple text file that can also be edited in any text editor. The HTML Layout Control can either be an entire Web page or a portion of a Web page, since it behaves like any other inserted object. Samples of the HTML Layout Control are available from either the HTML Layout Control (<http://www.microsoft.com/ie/controls/layout>) or the ActiveX Control Pad Web site.

Today, the HTML Layout Control works only in Microsoft Internet Explorer 3.0, but future versions will be supported by Netscape Navigator as well using the ActiveX Plugin for Netscape.

To insert an HTML Control into a Web page, choose “Edit/Insert HTML Layout” from the Control Pad menu. An object tag will be generated, along with an icon in the left pane. Click the icon to bring up a full frame-based page editor, which can now be used to create an advanced, 2D design by laying out multiple controls in a precise manner. Controls can be layered as well. Read the upcoming second part to this tutorial for more information on creating 2D designs. All of the layout information will be saved using the new draft W3C tags extending HTML to allow these advanced layout features.

Creating 2D HTML Designs using the Microsoft HTML Layout Control and ActiveX Control Pad

This section is under construction, refer to the ActiveX Control Pad Whitepaper at <http://www.microsoft.com/intdev/author/cpad> for basic information until this tutorial is complete.

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This document is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS OR IMPLIED, IN THIS DOCUMENT.

© 1996 Microsoft Corporation. All rights reserved.

Microsoft and Visual Basic are registered trademarks and ActiveX and FrontPage are trademarks of Microsoft Corporation.

Adobe and Acrobat are trademarks of Adobe Systems, Inc. Macromedia is a registered trademark of Macromedia, Inc. JavaScript is a trademark of Sun Microsystems, Inc.