

# Incorporating Plug-Ins and ActiveX Controls

---

**P**lug-ins offer you the ability to add many diverse types of media to your Web pages. Many of the desktop software packages you may use daily have plug-ins that enable the results of your work to be seen on the Web. Far too many types of plug-ins exist for this chapter to list all of them.

Netscape was first on the scene with plug-ins. ActiveX controls are Microsoft's answer to plug-ins. ActiveX controls, while they run under Netscape as well, were designed to add functionality to Internet Explorer 3 and higher. Just like plug-ins, before visitors to your site can use an ActiveX control on your Web page, they must download it.

What kind of functionality can ActiveX controls add to your page? Nearly any kind you can imagine and program. ActiveX controls are written in regular programming languages, using one of the Microsoft Software Development Kits (SDKs). Many ActiveX controls are available at no cost. Others can be licensed from the developers. Many resources are on the Web for finding both these kinds of controls. If you still can't find the one you want, you can write one yourself.

## Reviewing the OBJECT Element

The **OBJECT** element is one of the most versatile elements in HTML. You can use it to include inline graphics, audio, video, Java applets, and plug-ins in your page. Even though it has many attributes, you rarely need more than a few for any



### In This Chapter

Reviewing the OBJECT element

How plug-ins work

Plug-in pluses and minuses

Adding plug-ins

Testing your plug-in

Configuring your server

Introducing ActiveX

ActiveX pluses and minuses

Understanding the ActiveX security model

Finding ActiveX controls

Incorporating ActiveX controls

Defining options (parameters)

Getting around the Mac problem

Testing your control



instance of the element. Because this book aims to be comprehensive, you can find the complete OBJECT element definition in Chapter 19.

The main attributes you need in order to include inline plug-ins with the OBJECT element are data and type. For example, to include an inline VRML model, you would use the following HTML:

```
<OBJECT data="media/3D/music.wrl" type="world/wrl">
  Too bad you don't have a VRML plug-in. You can find one at <A
  href=" http://home.netscape.com/plugins/3d_and_animation.html"
  alt="Netscape plug-in download center">the Netscape plug-in
  download center</A>.
</OBJECT>
```

Visitors to your site would have to have a plug-in that recognized the MIME type “world/wrl.”

## How Plug-Ins Work

It's like magic. You download a plug-in and you install it. Then you come across a page with an application requiring that plug-in and voilà! Your browser runs that application just as if that functionality was built into the browser. How does the browser know which plug-in to use?

The answer is MIME types. Your computer has a list of MIME types it recognizes. Where does this list come from? Every time you install a software package, your computer adds this MIME type to the list. It uses this same list to assign a special icon to a file when you look at a file listing.

In Netscape, you can define a new MIME type manually, by going into Edit ⇨ Preferences ⇨ Applications. In Figure 42-1, you see a list of MIME types. A MIME type consists of the file extension (so the computer knows which files to associate with this MIME type on a PC or on UNIX), the category of file (audio, video, application, image, and so forth), and the application that should be used to open the file. For Macs, there aren't any file extensions.

In Internet Explorer, you actually set the MIME types and define new MIME types in the operating system, from any file listing (see Figure 42-2). If you go into an Explorer window (right-mouse click the Start button and choose Explore), and then select View ⇨ Options and click the File Types tab, you see a list of valid MIME types for your computer.

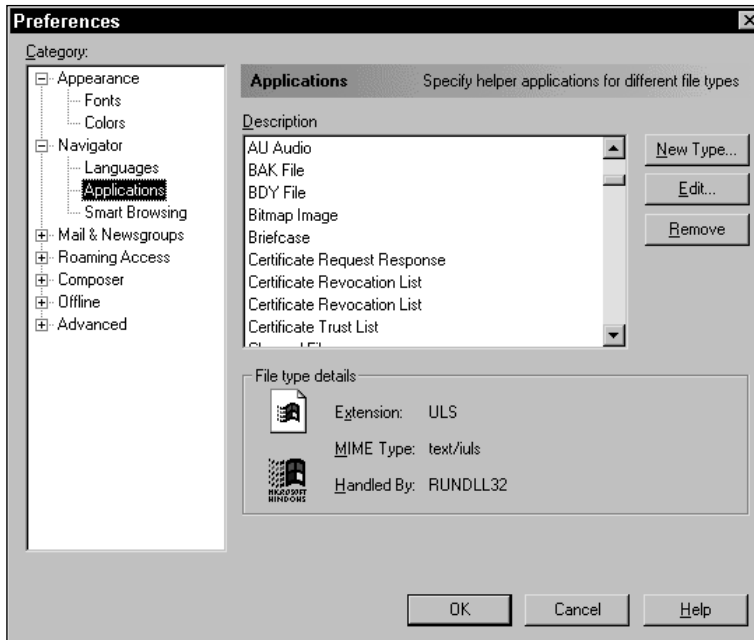


Figure 42-1: Setting MIME types manually in Netscape

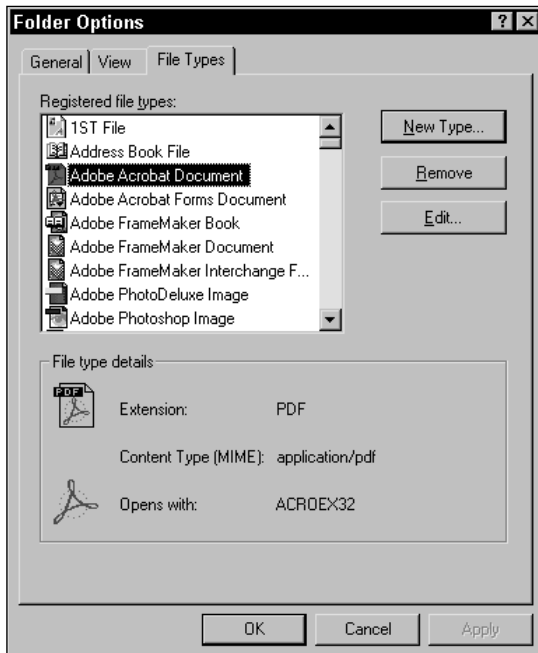


Figure 42-2: Setting MIME types manually for Internet Explorer

## Plug-In Pluses and Minuses

Plug-ins can be a rich addition to your site. They add interesting visual stimuli and are relatively quick to develop. As you read previously, adding an inline plug-in to your page is as simple as adding the `OBJECT` element. Adding an out-of-line plug-in to your page uses the familiar `A` element.

Plug-ins can do so much, so easily, why would a Web author want to avoid them at all? The major drawback to plug-ins is that visitors to your site might decide not to download the plug-in and would miss whatever exciting visual/aural stimuli you prepared for them. If a large portion of the message on your page is contained in a file requiring a plug-in, then you risk failing to communicate that message to much of your audience.

Plug-ins are best used in intranets, where you have some control over (or at least a knowledge of) the setup of each of the computers likely to visit your page. The other relatively safe environment in which to use a plug-in is on a site where the same information is offered in another format, say, text. Many news sites rely primarily on text to convey their messages, but offer video or audio to supplement the message of the text.

You want to avoid a page design where the only element on the page is your plug-in. This annoys people who don't have the plug-in, especially if you haven't provided enough information to convince them that they should go to the trouble of getting the plug-in and installing it. Make sure to provide a direct link to the site providing the plug-in, so people don't have to search for the plug-in.

## Adding Plug-Ins

Plug-ins can be divided into several categories: 3D and animation plug-ins, audio and video plug-ins, discipline-specific plug-ins, business plug-ins, image viewers, and presentation plug-ins. Often, the effects achieved with plug-ins can be achieved equally well with either JavaScript or Java. The drawback to using programming, specifically Java, is that the development time is much longer than simply using a plug-in. Frequently you must purchase additional software to implement a plug-in. Sometimes, your systems administrator must make changes to the server to accommodate new MIME types, because the Web server must also be familiar with the MIME type.

## 3D and animation plug-ins

If you want to take advantage of vector graphics or 3D models, you need a plug-in for now. Future versions of browsers undoubtedly will include support for vector graphics but, even then, not all visitors to your site will have the latest browser.

### Cosmo Worlds and World View

The most popular 3D plug-in is the one that has been around the longest: Cosmo Worlds by SGI. This runs on PCs or UNIX. If you want to display 3D models on a Mac, you can use World View by Intervista. The nonproprietary standard you can use to create 3D models is VRML. Creating sophisticated 3D models without some sort of tool is difficult and time consuming. Cosmo Create, also by SGI, is the most powerful tool for creating 3D models in VRML, the most commonly used 3D modeling language on the Web.

### Shockwave and Flash

Shockwave and Flash are plug-ins by Macromedia. Not surprisingly, these plug-ins support files created by Macromedia's own products: Director, Authorware, and Flash (see Figure 42-3). You can create impressive interactive animations and applications with Authorware and Director. Flash creates vector graphics.



Figure 42-3: Shockwave plays Director and Authorware files.

Why would you want to use Flash as a plug-in when you can create graphics that display inline without any plug-ins? Vector graphics are an efficient way to deliver graphics to the Web.

In Depth



Vector graphics files are small because, instead of a file containing the image, the file contains only the equations that create the images (remember geometry class?). Also, the graphics created by the equations are infinitely scalable without any loss of quality. Expect to see much more of vector graphics on the Web as Version 5 browsers support this format for inline graphics.

## Audio and video plug-ins

Both audio and video can be played by the Version 3 and 4 browsers without any special plug-ins. If you want to use fancier video or audio tools, however, including real-time streaming video or audio that takes advantage of a special server, you need a special plug-in. The other reason to use a plug-in is to show QuickTimeVR movies, which can be an effective way to show an object in 3D or to navigate around a space.

### RealPlayer

RealPlayer by RealNetworks, Inc. kills two birds with one stone. If you are using either the RealAudio server or the RealVideo server, RealPlayer is all your site visitors need. Either of these technologies, while relatively expensive to implement, result in excellent image/sound quality delivered in real-time without those pesky download delays. Another advantage to this technology is the browser computer reuses cache when downloading the files. This means the entire video or audio file doesn't reside on the visitor's computer at any given time, saving the visitor precious hard drive space.

### NetShow

NetShow is another streaming video application. NetShow is seen more commonly on sites designed to use the full suite of elements only available to Internet Explorer. NetShow does nearly as nice a job as RealPlayer. The NetShow server is available at no cost from Microsoft.

### QuickTimeVR

QuickTimeVR is a video application that competes with both video products and 3D modeling products. QuickTime is the original video technology used on Macs. QuickTimeVR can be effective at showing an object from any perspective. Say you

are creating an exhibit of products and you want people to move the objects so they can look at the objects from all sides. How would you do this? You might create a movie of the object as you move it (or move around it, depending on its size), but how fast should you move? What if visitors want to linger at a certain perspective? QuickTimeVR solves all this by creating a cylinder of images, woven together so viewers can turn the cylinder, looking at any angle, and then move the object so they can look at any other angle. The same can be done with panoramas: QuickTimeVR can create a seamless view of the Grand Canyon from the middle. In addition to giving viewers control over the speed of the “movie” and the navigation, QuickTimeVR creates relatively small files — compared to real movies — so this is a winner from every perspective.

## Discipline-specific plug-ins

For some fields, such as chemistry, incredibly powerful plug-ins enable you to express an idea or equation with such brevity, you just can't avoid using plug-ins. If your discipline is something other than chemistry, look and see if there isn't some plug-in to make your life easier before you embark on a Java program to solve your display problems.

Chime by Chemscape is one of these plug-ins. You might not need to represent molecules, but if you did, this would be the product to use. Go to [www.mdli.com](http://www.mdli.com) to download the tool.

## Business plug-ins

Probably the single most popular plug-in in use on the Web is the Acrobat Reader, which displays Portable Document Format (PDF) files. *PDF* is Adobe's own format for saving formatted files for printing. Who uses PDF files on the Web? The IRS for one. Nearly any tax form you could possibly want is available on the IRS Web site ([www.irs.gov](http://www.irs.gov)). The beauty of PDF is that the file can be printed the way it was intended. If you take a form and re-create it on the Web (in HTML), you can't be sure where page breaks will fall. If you take a form and create a PDF file with it, then everyone who opens it can print it to look exactly the same.

To create PDF files, you need to purchase one of the Adobe products that creates PDF files. Adobe Acrobat (\$249 as of this writing) is the basic package you need if you want to publish documents in PDF format. Figure 42-4 is an example of what a PDF file would look like on your screen. If you already use Adobe Illustrator to

create documents, you can export directly from Illustrator into the PDF format. For most documents, which you create in your favorite word processor or scan in using optical character recognition (OCR), you need Acrobat. Using the Acrobat Writer, you can “print” any document to PDF format. If you have a document in postscript format, Distiller converts it into PDF. You can even edit a file you have converted into PDF format using Adobe Exchange.

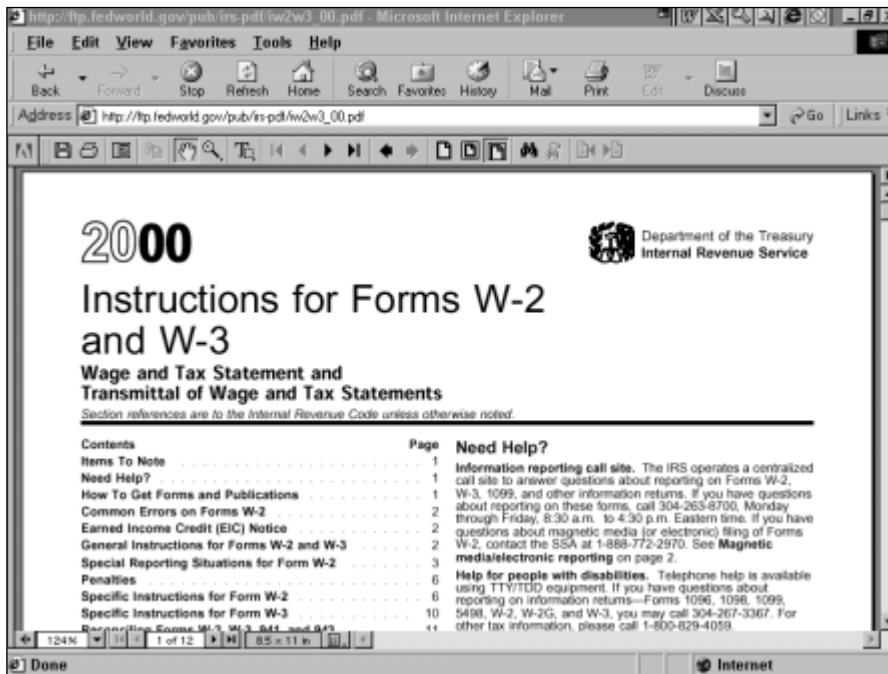


Figure 42-4: The IRS makes most forms available in PDF format on its Web site.

## Testing Your Plug-In

As with everything else you publish on your Web site, you should test your plug-ins extensively. You want to test them from multiple platforms, using Internet Explorer, Netscape, and AOL. In fact, you probably want to test your plug-in from a computer that doesn't already have the plug-in installed. This way, you can go through the process many of your site visitors will go through:

1. Arrive at your page and get notification you need a plug-in.



- 2. Follow the link on your page to get the plug-in.** This link should open a new window so the visitor doesn't lose your page.



Tip

Use an A element to link with the value of the `target` attribute set to "blank" to open a new window for the plug-in page.

- 3. Download and install the plug-in.** Does it require restarting the browser? You want to know this so you can tell the visitor to bookmark your page before quitting the browser. You might also want to tell the visitor on your page how long a download takes and what is involved in installation.
- 4. Play the plug-in on your page.** Does it require reloading the page? You should mention this somewhere.

## Configuring Your Server

For some plug-ins, the server needs some special information. For the Chime plug-in, for example, the server must be notified some special MIME types exist. When you use a plug-in, be sure you have read all the documentation on the plug-in site, so you know if any changes must be made to your server. The odd thing about these server changes is, if you had played the same plug-in from a CD on your desktop, you wouldn't need any new MIME types. No good rule exists about when you will need to have your systems administrator make MIME type changes to your server. You have to read the documentation and do some testing yourself to see if your plug-ins will play from your server.

## Introducing ActiveX

ActiveX is Microsoft's solution for letting Web developers add their own functionality to Internet Explorer. *ActiveX* is a marketing name for a set of technologies based on the Component Object Model (COM). ActiveX controls run under Netscape as well, but they require a Netscape ActiveX control to do so. Programmers, writing in traditional second-generation languages, such as C, C++, Visual Basic, and Java, can write ActiveX controls. By taking advantage of existing Object Linking and Embedding (OLE) architecture, ActiveX controls are smaller than regular programs or Java Applets, and they are optimized for download and execution. In addition, they register themselves on the client computer.

Unlike pages with plug-ins, pages with ActiveX control initiate the download of the ActiveX control. The beauty of this approach is that visitors to the page don't need to know anything about anything. They can simply accept or refuse the download.

Licensing for ActiveX controls can be complicated. Some ActiveX controls are freely distributable. Some require licensing from the author. In these cases, you must get a license file to place on your server so the ActiveX control will run properly. Of course, if you write your own, you needn't worry about licensing issues.

## ActiveX Pluses and Minuses

ActiveX has its advantages over plug-ins. The biggest advantage is to the site visitor. Visitors to the site needn't know anything to accept the ActiveX control. All they need to do is click the word *Accept* to have the ActiveX control download.

ActiveX controls exist that don't support specific commercial technologies. The vast majority of Netscape plug-ins were written — like the Acrobat Reader and RealPlayer — to facilitate distribution of proprietary media standards over the Internet. ActiveX controls tend to focus more on performing a specific task (running a clock, displaying a calendar, showing a stock ticker) than on delivering technologies.

One downside of ActiveX controls is, while ActiveX controls aren't necessarily difficult to implement, they are more complicated to implement than a plug-in. Another disadvantage of ActiveX controls, which is the same as the disadvantage of plug-ins, is that people can choose not to accept the download of the plug-in. When this happens, your page might have a big hole in it or it might not function properly.

### Mac Support

The biggest minus to ActiveX controls is they don't run on Macs. If you are building an application for the Internet or, more specifically, for education, this is an insuperable obstacle to using ActiveX controls. You can do something to get around this major hole in ActiveX design. See the section "Getting Around the Mac Problem" later in this chapter.

## Understanding the ActiveX Security Model

How safe is it to use ActiveX controls? That depends. The ActiveX security model relies on the goodwill of the ActiveX control programmer. ActiveX controls are inherently dangerous. The ActiveX security model enables the programmer to have full and free control of your computer. This makes ActiveX controls powerful. They can read, write, and edit files. This also makes ActiveX controls potentially *very* dangerous. Because ActiveX controls have unrestrained access to your computer, no limit exists to the damage they can do. There is no logging of the actions ActiveX controls take, so there is no way later to trace which controls, if any, caused system problems you are having.

The Microsoft answer to the cavalier approach ActiveX controls take to security is that all controls will be digitally signed by the distributor, and you, as the site visitor, have the power to accept or refuse controls, based on whether you trust the distributor. Is this enough? No. What if the programmer, whom you trust, accidentally leaves a security hole on your system and another site you go to knows about this and takes advantage of it? Once you accept an ActiveX control, it gets installed on your system and stays there. Any site you subsequently visit can use this control, even if you never granted it explicit permission to do so.

What does all this mean to you as a Web developer? It means you should be careful about using other ActiveX controls you come across — even if you think they're perfectly safe — because you may unwittingly contribute to problems on your visitors' computers. It also helps you understand why so many people who come to your site choose not to accept your controls. Visitors need to look after the integrity of their own computers.

When can this security model be a good thing? If you are developing for an intranet, you can write ActiveX controls that perform powerful actions on client computers, without having to install that software manually on every computer. Be sure to test your controls carefully, especially if they affect the file system at all.

## Finding ActiveX Controls

Chances are, you'll start your foray into using ActiveX controls by using ones others have written. Lots of places exist on the Web where you can find sources of controls. Some of these controls have licensing costs associated with them, but you can find out about this when you identify the controls you want to use.

Dozens of sites consider themselves resources for ActiveX controls. One place to start is the **Browser Watch** site (<http://browserwatch.internet.com/activex.html>). Although you may find 20 sites that list controls, you will find they all list basically the same controls. Browser Watch does a nice job of directing you to the control without numbing your senses with visual clutter and advertising in the process.

In Depth



Publicizing your own controls. If you find no one has written the control you need and you are brave enough to take this on yourself, you might want to have your own control listed on these sites. Each site has its own form for submitting your ActiveX control information.

## Incorporating ActiveX Controls

Once you identify the ActiveX control you want, you insert it with the **OBJECT** element. Even though over a dozen attributes exist for the **OBJECT** element, most of the time you only need four of them. You also need an indeterminate number of **PARAM** elements.

```
<OBJECT id="Calendar1" width=372 height=279
  classid="CLSID:8E27C92B-1264-101C-8A2F-040224009C02">
  <PARAM name="_Version" value="458752">
  <PARAM name="_ExtentX" value="9843">
  <PARAM name="_ExtentY" value="7382">
  <PARAM name="_StockProps" value="1">
  <PARAM name="BackColor" value="12632256">
  <PARAM name="Year" value="2000">
  <PARAM name="Month" value="3">
  <PARAM name="Day" value="9">
</OBJECT>
```

In the previous example, which inserts the calendar ActiveX control into your Web page, only four attributes of the **OBJECT** element are needed.

```
<OBJECT id="ShockwaveFlash1" width="192" height="192"
  classid="CLSID:D27CDB6E-AE6D-11CF-96B8-444553540000">
  <PARAM name="Movie" value="">
  <PARAM name="Src" value="">
  <PARAM name="WMode" value="Window">
  <PARAM name="Play" value="-1">
  <PARAM name="Loop" value="-1">
  <PARAM name="Quality" value="AutoLow">
  <PARAM name="SAlign" value="">
  <PARAM name="Menu" value="-1">
  <PARAM name="Base" value="">
  <PARAM name="Scale" value="ShowAll">
  <PARAM name="DeviceFont" value="0">
  <PARAM name="EmbedMovie" value="0">
  <PARAM name="BGColor" value="">
</OBJECT>
```

In the previous example, which inserts the Shockwave ActiveX control, again only four attributes are needed.



Tip

The `classid` attribute. Where do you get that long, nasty `classid` attribute? You get this information, which is the unique identifier for that ActiveX control, from the owner of the control. Depending on the licensing of the control, you can sometimes copy the source of a control you find on a page you like. If licensing restrictions exist, then this approach won't work and the control won't run on your page.

## Defining Options (Parameters)

How many parameters do you need for an ActiveX control? This depends on the control. For some controls, no parameters exist. For others, there can be ten or more. You can find out about the parameters you have to set wherever you find out about the control.

How do you set parameters? With the `PARAM` element.

**Parameter** <PARAM>

**Start Tag:** Required

**Content:** Empty

**End Tag:** Forbidden

**Attributes:** `id`: document-wide unique ID; optional

`name`: name of the parameter; defined by the ActiveX control

`value`: value associated with the parameter specified by the `name` attribute

`valuetype`: how to interpret the value: `data` or `ref` or `object`; `data` is the default, `ref` indicates the value is a URL, `object` indicates the ID of another object defined in this page

`type`: MIME type of parameter

## Getting Around the Mac Problem

The easiest thing you can do is to nest `OBJECT` elements. Place the `OBJECT` you most want your visitors to use in the outermost `OBJECT` element. Then, after your `PARAM` elements, place the next `OBJECT` element you would want them to use. Finally, within your last choice `OBJECT` element, you can include alternate text explaining what they are missing.

```
<OBJECT id="ShockwaveFlash1" width=192 height=192
  classid="CLSID:D27CDB6E-AE6D-11CF-96B8-444553540000">
  <PARAM name="Movie" value="mymovie.dcr">
```

```
<PARAM name="Src" value="">
<PARAM name="WMode" value="Window">
<PARAM name="Play" value="-1">
<PARAM name="Loop" value="-1">
<PARAM name="Quality" value="AutoLow">
<PARAM name="SAlign" value="">
<PARAM name="Menu" value="-1">
<PARAM name="Base" value="">
<PARAM name="Scale" value="ShowAll">
<PARAM name="DeviceFont" value="0">
<PARAM name="EmbedMovie" value="0">
<PARAM name="BGColor" value="">
<OBJECT name="veryshocked.dcr"
type="application/futuresplash"
This is a very nice Shockwave animation. You should consider
downloading the plug-in for this at <A
href="http://www.macromedia.com" alt="Macromedia">the
Macromedia site</A>.
</OBJECT>
</OBJECT>
```

## Testing Your Control

In every chapter, when this book discusses testing your work, you hear you need to test from more than one browser. This is extra important when using ActiveX controls because they weren't really designed to run in Netscape. ActiveX controls also aren't natively supported in AOL or in Internet Explorer browsers prior to Version 2. To test your control, start by testing your control from a computer that doesn't already have the control installed. This is the only way to check and make sure the control downloads and installs itself properly. You also want to test your control on multiple PC operating systems to make sure it works properly. Because ActiveX doesn't run on Macs reliably, you needn't worry about testing it on a Mac, but you will want to include alternate text between your `OBJECT` begin and end tags explaining what the viewer is missing.

## From Here



Jump to Chapter 48 and learn about another way to make your pages interactive — with JavaScript.

Proceed to Chapter 43 and learn about Java applets.

## Summary

In this chapter you learned about some of the powerful effects you can add with plug-ins. This chapter discussed how they work, using MIME types, and how you include them in your Web page, using the `OBJECT` element. You also learned about the pros and cons of using plug-ins and why you should include the same information in an alternate form for people who don't bother to go out and get the plug-ins. You learned about some of the hottest plug-ins on the Web today, how to test your plug-ins, and a bit about configuring your server for plug-ins that require the server to know about its MIME type.

In this chapter you also learned about the pros and cons of ActiveX controls. You learned how they differ from plug-ins and how the security model (or lack thereof) works. You also learned ActiveX controls aren't your best choice if you anticipate any of the visitors to your site might be using Macs. This chapter gave you resources for finding ActiveX controls on the Web and for publicizing your own, if you choose to write any yourself. Finally, you learned how the `OBJECT` element and `PARAM` element are used to define an ActiveX control within your Web page. Testing an ActiveX control is similar to testing a plug-in.



