43

CHAPTER

Adding Java Applets

✦ ✦ ✦ ✦

In This Chapter

✦ ✦ ✦ ✦

Java! What technology has produced more hype than Java? Is it merited? Will it really change the face of computing? This chapter won't cover much about that, but it can help you understand what Java applets can do for your Web page today. It explains how Java differs from plug-ins and ActiveX controls, the other ways of adding extra functionality to your Web page. You learn about the Java security model, which is much stricter than the ActiveX or plug-in security model. You also learn how to insert a Java applet into your page.

Introducing Java

Java is a way to add additional functionality to your browser. Because it downloads within the browser page and doesn't operate outside of the browser's operating space, it doesn't require permission from the page visitor before it can run. Visitors can set their browsers not to accept any Java applets, but this isn't all that common. As long as visitors to your page have a Java-enabled browser and the patience to wait for the applet to start, they will get your Java applet.

Java is a nonproprietary software standard developed by Sun Microsystems. Java is an open standard, intended to take the best of C++ and simplify it — stripping out the most convoluted parts, such as memory management, which Java handles automatically. Java is fully supported in both Microsoft Internet Explorer and Netscape Navigator.

Java is an *object-oriented programming language*. Note two things about this catch phrase: First, *object-oriented* means a program is designed around the data. This only means something if you understand the old way of programming — procedure-oriented programming — where the program was designed around what it did, not around the data. Second, Java is a *programming language*. Java is not for the faint-hearted. If you are not a programmer, this probably isn't how you want to begin. JavaScript, which is completely unrelated to Java, is a better starting point for aspiring programmers.

Two kinds of Java actually exist: server-side Java and client-side Java. This chapter only discusses client-side Java, which is inserted into your page as an applet. Server-side Java enables your page to talk to a database or to perform advanced server functions. Server-side Java doesn't have all the constraints on what it can and can't do that client-side Java has. Server-side Java uses the full power of the Java programming language.

Java Pluses and Minuses

Why would anyone want to use Java? Java is a safe way to add limited functionality to your browser. Java uses a much tighter security model, so people are less worried about damage to their computers. Consequently, more people have their browsers set to accept Java applets than have their browsers set to accept (or are willing to accept) ActiveX controls.

So, what can Java do? It can read the local file system, meaning it can show you the contents of a local file, and it can perform animations. Java can create new windows within the existing browser, and it can perform certain actions based on the position of the mouse.

The biggest drawback to using Java applets is you have to write a program. The beauty of the Web and the reason it has grown so fast is that creating Web pages doesn't require any special programming skills or training. Anyone can write HTML, as this book shows. Java applets return to the programming model of delivering functionality. When you program, you have to test thoroughly to make sure there aren't any unintended consequences from your actions. With HTML, most of the mistakes you can make are immediately visible and easily corrected.

Programming Java is time consuming and requires special skills. It is often faster to find a plug-in or ActiveX control to do the same thing or to find a way to provide comparable functionality without programming.

The Java virtual machine

Why else might you not want to use Java? Speed, or lack thereof. Java is a clever programming language, but this cleverness results in decreased performance in a network environment. Java is designed to run on any platform. When you compile Java (as you normally do with any programming language), instead of being compiled into machine language, which is specific to the platform on which you are compiling, Java is compiled into byte-code, which is generic and platform-independent.

That byte-code is what is sent over the network to your browser. Your browser has a virtual machine built into it that compiles the byte-code into machine language before it runs. The time saved by the programmer in writing the code only once and having it run on every platform is spent by everyone who ever tries to run the program on his or her own virtual machine.

The Just-In-Time compiler

Virtual machines, through the use of Just-In-Time compilers, are getting faster. Perhaps as desktop machines get faster you won't notice the time it takes to run the byte-code through the virtual machine to get the machine code, but today it is slow. Some of your site visitors will get impatient and stop the download, not understanding why it takes so long.

Understanding the Java Security Model

The Java security model is much safer for the casual Web visitor. While the ActiveX security model and the plug-in security model rely on the intelligence of the user and the goodwill and competence of the programmer, the Java security model relies on the sandbox.

Everything that takes place within a Java applet takes place within a sandbox that constrains the Java applet. The Java applet can't act on anything outside of this sandbox in any permanent way. What can a Java applet do?

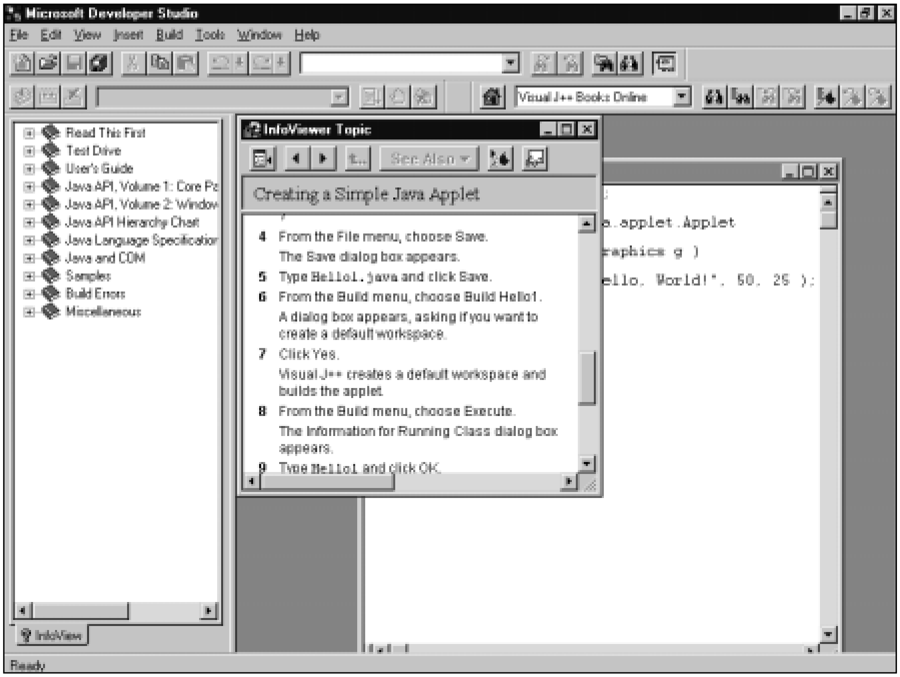
- Access the CPU.
- Work within the limited space the browser uses.
- Read input from the mouse or keyboard when the mouse has clicked within the applet window.
- Work within the browser's windowing system.
- Create new windows within the browser.

Where can even this tight security fail the user? If the sandbox security is not implemented properly by the browser developers, then the Java applet could presumably take advantage of any security hole they created.

Overall, the Java security model is much tighter than the ActiveX security model. ActiveX can write to your file system, potentially corrupting essential system files. Java can't touch your file system. Java can view your file system, but can't make any changes to it.

Java Development Tools

Many Java development tools are on the market. Two with wide popularity for good reasons are Microsoft's Visual J++ (see Figure 43-1) and Visual Cafe for Java by Symantec (see Figure 43-2).



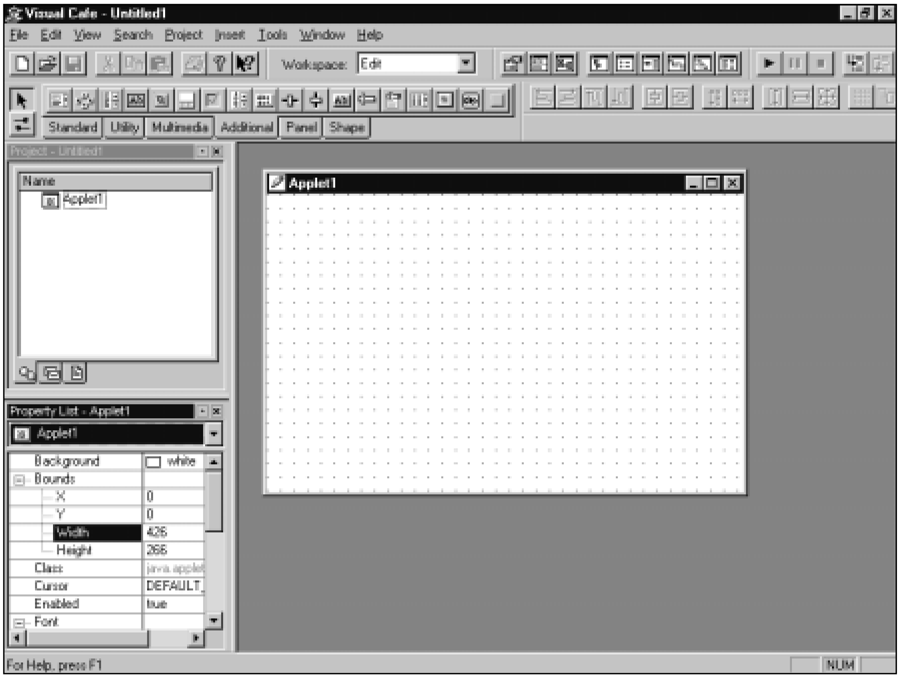
Creating a Simple Java Applet

4 From the File menu, choose Save.
The Save dialog box appears.
5 Type Hello1.java and click Save.
6 From the Build menu, choose Build Hello1.
A dialog box appears, asking if you want to create a default workspace.
7 Click Yes.
Visual J++ creates a default workspace and builds the applet.
8 From the Build menu, choose Execute.
The Information for Running Class dialog box appears.
9 Type Hello1 and click OK.

```
a.applet.Applet
raphics g )
ello, World!", 50, 25 );
```

Figure 43-1: Visual J++ by Microsoft

Microsoft's Visual J++ isn't completely compatible with Netscape. Sun Microsystems writes the Java standard. Microsoft's standard isn't completely compatible with Sun's standard. To make matters worse, Netscape is behind on the Java standard, meaning Netscape 4 requires a patch to run the 1.1 Version of Java.



Background	□ white
Bounds	
X	0
Y	0
Width	426
Height	266
Class	java.applet
Cursor	DEFAULT_
Enabled	true
Font	

Figure 43-2: Visual Cafe for Java by Symantec

If you do anything that is more than a basic applet, you must choose which browser it runs on, because the two major browsers are so incompatible. Also, for anything more than the basic sandbox application, you need to get a digital signature (which gives you more control over the system, like an ActiveX control) from Verisign. If you want digital signatures for both Microsoft and Netscape, you have to buy both separately. Then you have to get the digital signature tool from either Netscape or Microsoft (each is application-specific).

Java SDK

You can get the software development kit (SDK) at `www.javasoft.com`. It has all the libraries, instructions, and Java tools that work on the command line. The JavaSoft site is the place to go for information about writing Java and adding Java applets to your pages.

Incorporating Java Applets

Not surprisingly, Java applets can be included in your Web page with the OBJECT element (defined in its entirety in Chapter 19). Before HTML 4, Java applets were routinely inserted with the APPLET element, but that is deprecated in HTML 4. Because the OBJECT element doesn't work with every browser, this chapter gives you examples of how to include an applet with both the APPLET element and the OBJECT element. The following code demonstrates how to use the APPLET element.

```
<APPLET code=graph.class width=760 height=470>
<PARAM name= "title" value = "Hanes-Woolf Kinetics">
<PARAM name="xLabel" value = "[S]">
<PARAM name="yLabel" value = "[S]/v">
<PARAM name="xUnits" value = "mM">
<PARAM name="yUnits" value = "sec/micro-mole">
<PARAM name="imageName" value = "hw.gif">
<PARAM name="scaleX" value = ".2">
<PARAM name="scaleY" value = ".01">
<PARAM name="quadrants" value = "2">
</APPLET>
```

You only need a few attributes of the OBJECT element for your Java applet definitions, as shown in the following code. Implementing an applet with the OBJECT element isn't supported yet in either Netscape 4.7 or Internet Explorer 5.

```
<OBJECT
classid="http://nmc.itc.virginia.edu/nlii/grisham/kinetics/hw/
graph.class" width=760 height=470 type="application/java-
archive">
<PARAM name = "title" value = "Hanes-Woolf Kinetics">
<PARAM name = "xLabel" value = "[S]">
<PARAM name = "yLabel" value = "[S]/v">
<PARAM name = "xUnits" value = "mM">
<PARAM name = "yUnits" value = "sec/micro-mole">
<PARAM name = "imageName" value = "hw.gif">
<PARAM name = "scaleX" value = ".2">
<PARAM name = "scaleY" value = ".01">
<PARAM name = "quadrants" value = "2">
</OBJECT>
```

Defining Options (Parameters)

As with ActiveX controls, Java applets take parameters in the PARAM element. Important to note is the PARAM names must exactly match what the Java applet is looking for. The values assigned to these names must also be of the correct type.

Cross-Reference

See Chapter 42 for a more thorough explanation of how the PARAM element works.

Testing Your Applet

You must test your Java applet on both your local workstation (before you include it on your page) and on your Web page. The best place to do your first round of testing is in the Java development tool you use.

When you are confident your Java applet does what you want it to do, you can upload it to your server. Your Java applet must reside in a directory with execute permissions. Most of your pages only require read permission, but because the Java applet is executable code, your directory also must permit execution. If your applet won't run at all, check with your systems administrator to see whether you have execute permissions on that directory.

As with everything else on your pages, you should test your Java applet from a variety of platforms, from more than one browser. If any unusual behavior occurs in any of these environments, you should warn visitors and refer them to a configuration that works. Not every implementation of Java on every platform works consistently.

Introducing Server-Side Java

What can you do with server-side Java? Server-side Java enables your Web pages to interact with your Web server. Pretty much anything you can do with CGI (Common Gateway Interface) scripts, Perl, or server-side JavaScript, you can do with server-side Java. Using the JDBC (Java database connector), you can have your Web page communicate with a database, returning dynamic pages based on the information entered. You can also create and read files on the server.

Cross-Reference

If you already know Java, this may be a way to perform server functions, such as those discussed in Chapter 21. Many of the back-end tools you can purchase for your server, including some that perform database access, are written in Java.

From Here

Cross-Reference

Go to Chapter 45 to learn about accessing external databases.

Jump to Chapter 48, which provides an introduction to JavaScript.

Review how to create forms in Chapter 21.

Summary

Java applets can add functionality to your pages. They generally adhere to a tight security model that doesn't permit them to do any damage to your machine. Signed applets can be given (by you) permissions to act on your machine like ActiveX controls. Unfortunately, Java isn't yet truly cross-platform; applets you write to run under Internet Explorer won't necessarily work under Netscape. Chapter 45 discusses how you can access external databases with your form data. Server-side Java is one technology that enables you to do this.

✦ ✦ ✦