

OSBORNE 

The Complete Reference


CD-ROM Features
FrontPage 2000 45-Day Trial,
Web tools, add-ons, and pre-built scripts

"If FrontPage can do it, *FrontPage 2000: The Complete Reference* covers it. Every feature is described so that anyone can understand it. Recommended!"

— Randy Forgaard, Co-creator of Microsoft FrontPage

FrontPage[®] 2000

Detailed explanations on
creating robust, high-
quality Web applications

Full coverage of XML, Active
Server Pages, Java applets,
VBA, and much more

Covers the latest features—
themes, DHTML effects,
and cascading style sheets

Martin S. Matthews

Best-Selling Author of More than 50
Computer Books

Erik B. Poulsen

Highly Acclaimed Senior Web Designer
for Arcadia Web Service

Petitioner Microsoft - Ex. 1075, p. 1

FrontPage 2000: The Complete Reference

**Martin S. Matthews
Erik B. Poulsen**

Osborne/McGraw-Hill

Berkeley New York St. Louis San Francisco
Auckland Bogotá Hamburg London Madrid
Mexico City Milan Montreal New Delhi Panama City
Paris São Paulo Singapore Sydney
Tokyo Toronto

Osborne/McGraw-Hill
2600 Tenth Street
Berkeley, California 94710
U.S.A.

For information on translations or book distributors outside the U.S.A., or to arrange bulk purchase discounts for sales promotions, premiums, or fund-raisers, please contact Osborne/McGraw-Hill at the above address.

FrontPage 2000: The Complete Reference

Copyright © 1999 by The McGraw-Hill Companies. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a database or retrieval system, without the prior written permission of the publisher, with the exception that the program listings may be entered, stored, and executed in a computer system, but they may not be reproduced for publication.

34567890 AGM AGM 90198765432109

ISBN 0-07-211939-X

Publisher

Brandon A. Nordin

**Associate Publisher and
Editor in Chief**

Scott Rogers

Acquisitions Editor

Wendy Rinaldi

Project Editor

Ron Hull

Editorial Assistant

Monika Faltiss

Technical Editor

John Cronan

Copy Editor

Gary Morris

Proofreader

Rhonda Holmes

Indexer

James Minkin

Computer Designer

Ann Sellers

Mickey Galicia

Jani Beckwith

Illustrator

Robert Hansen

Beth Young

Series Design

Peter Hancik

Microsoft and FrontPage 2000 are registered trademarks of Microsoft Corporation.

Information has been obtained by Osborne/McGraw-Hill from sources believed to be reliable. However, because of the possibility of human or mechanical error by our sources, Osborne/McGraw-Hill, or others, Osborne/McGraw-Hill does not guarantee the accuracy, adequacy, or completeness of any information and is not responsible for any errors or omissions or the results obtained from use of such information.

The Complete Reference



Part I

Getting Started

The Complete Reference



Chapter 1

Designing Quality Web Applications

Communication, whether it be within a small group, throughout a large organization, or among many organizations, can almost always be improved—made faster, easier to receive, and easier to respond to. The *web application*, a multimedia form of communication including text, graphics, audio, video, and scripts transmitted by computers, is the latest improvement. While computers sit on the sending and receiving ends of web communication, it is what links the computers that gives web applications one of their most important features. The link means that senders and receivers can operate independently—senders can put the web content up according to their schedule, and receivers can get it anytime thereafter. The link used for the transmission of web content is one of two forms of networking, either the public *Internet*, which uses public and private networks, including phone lines; or a private *intranet*, which uses a *local area network* (LAN), generally within an organization.

The Internet is at the foundation of a global communications revolution that has changed the way people communicate, work, and conduct business. The Internet makes it easier and cheaper to exchange information, ideas, and products around the globe. Accessing a web site in Australia is as easy as accessing one across the street.

In this book you will learn how you can be part of this revolution. You will learn how to use FrontPage to create and maintain a presence on the Web for your business, your organization, or for yourself. This book will take you through all the steps necessary to create your own web application—from initial design to placing your content on a web server where it can be accessed by anyone on the Web.

The Internet

The Internet is a network infrastructure of computers, communications lines, and switches (really other computers) that use a set of computer hardware and software standards, or *protocols*, that allow computers to exchange data with other computers. The computers can be in the same room, or they can be located around the world from each other. They can use the same operating system software, such as Windows 98, or each can use a different computer operating system, such as Macintosh System 8.0 or UNIX. The standards that make up the Internet have become a modern *lingua franca*—a language enabling any computer connected to the Internet to exchange information with any other computer also connected to the Internet, regardless of the operating systems the computers use.

The birth of the Internet can be traced back to the late 1960s when the use of computers by Advanced Research Projects Agency (ARPA) and other government agencies had expanded so much that a way for the computer systems to share data was needed. ARPANET, the predecessor to what we now know as the Internet, was created to meet this need.

Another milestone in the history of the Internet came in the mid-1980s, when the National Science Foundation (NSF) added its five supercomputing centers (NSFNET) to the Internet. This gave educational centers, the military, and other NSF grantees

access to the power of these supercomputers and, more importantly, created the backbone of today's information superhighway. This backbone is made up of all the high-capacity (or wide-bandwidth) phone lines and data links needed to effectively transfer all the information now on the Internet. Until this wide-bandwidth infrastructure existed, the potential of ARPANET, NSFNET, and now the Internet couldn't be realized. By the end of the 1980s almost all the pieces were in place for a global telecommunications revolution.

The World Wide Web

By 1990 the Internet had grown to be a highway linking computers across the United States and around the world, but it was still a character-based system. That is, what appeared on computer screens connected to the Internet was simply text. There were no graphics or hyperlinks. A *graphical user interface* (GUI) to the Internet needed to be developed. Tim Berners-Lee, a scientist working at the European Laboratory for Particle Physics (CERN) in Geneva, Switzerland, proposed a set of protocols for the transfer of graphical information over the Internet in 1989. Berners-Lee's proposals were adopted by other groups, and the World Wide Web was born.

The Internet is a *wide area network* (WAN), as compared with a local area network (LAN) among computers in proximity. For computers to share information over a WAN, there must be a physical connection (the communications infrastructure created by ARPA and the NSF and now maintained by private industry) and a common software standard that the computers use to transfer data. The physical connection depends on whether you use a modem to dial up to the Internet, or whether your computer is part of a LAN with an Internet connection. The physical layer includes the modem or network interface card in your computer. You also need a phone or dedicated network line that connects you to the Internet backbone. In either case, your computer, connected to the Internet either with a dialup or network connection, is capable of sharing information with any other computer connected to the Internet anywhere in the world.

Note

The term "modem" as used here includes DSL and ISDN "adapters." Such devices provide the means to dial up an Internet connection over ordinary phone lines.

LANs and an Intranet

As important as the Internet has become to society, local area networks have become even more important to the exchange of information and communication within organizations. LANs started out as a way to share programs and data files among several people in an organization. This was then augmented by electronic mail (e-mail) for sending and receiving messages over the LAN. Recently, intranets have been added to LANs to provide a miniature version of the World Wide Web within an

organization—a place for people to post and read text and graphics documents whenever they choose.

A good example of how an intranet can be put to use is a project report. Instead of e-mailing a weekly update to a long list of potentially uninterested people (and filling up everybody's inbox in the process), you could post a web page on the intranet that would give not only the current status, but also other, more static information, such as the people working on the project, its goals, and its funding. In this way, those people who are truly interested can get the information.

With Active Server Pages (included with Microsoft's Internet Information Services) and Microsoft Office, even more intranet interactivity is possible. Users can send commands to the web application and then receive customized responses. Any Microsoft Office document can be included in a web page. Databases can be queried, and custom web pages can automatically be generated to display query results. Integrating Office and other files with your FrontPage web sites is covered in Chapter 12, Active Server Pages is covered in Chapter 16, and working with databases is covered in Chapter 18.

Note

Internet Information Services (IIS) is included with Microsoft's Windows NT Server. IIS and FrontPage provide all the software tools you need to set up a full-scale intranet or World Wide Web site. FrontPage along with the Microsoft Personal Web Server can host limited web sites by themselves.

Except for possibly the content, there is no difference between a web site on the World Wide Web and a web site on an intranet. They are created the same way and can have the same features and components. The discussion and instructions throughout this book are aimed equally at the World Wide Web and an intranet, and there are examples of each. So in learning to create a web application, you can apply that knowledge to either form of dissemination.

Tip

Think of the Internet and a LAN as equivalent means of information transmission, one public and the other private. And think of the World Wide Web and an intranet as equivalent means of posting and reading information being transmitted over the Internet or a LAN, respectively. The World Wide Web and an intranet are just advanced electronic bulletin boards, and a web page is an electronic document posted on that bulletin board.

Note

In a unique blending of two concepts, intranets can be extended to outside users, creating an extranet. For example, a corporation may provide other businesses, such as vendors or customers, access to a select subset of corporate data and a way to exchange the information.

Internet Protocols

The Internet and the Web are built upon several protocols:

- **Transmission Control Protocol/Internet Protocol (TCP/IP)** controls how information is packaged to be transferred between computers connected through the Internet.
- **Hypertext Transfer Protocol (HTTP)** is the language the computers use to exchange information on the Web.
- **Hypertext Markup Language (HTML)** is the programming language used to create the documents that are distributed on the Web and displayed on your monitor.

TCP/IP

To transfer information over the Internet or within a LAN, several requirements must be met. These include a way to assign each computer or site on the network a unique address (just like having a unique postal address) and “packaging” the information for transmission. These functions are handled by the Transmission Control Protocol and the Internet Protocol.

Internet Protocol

The foundation of the system is the Internet Protocol (IP). The IP converts data into packets and provides an address for each site on the Internet. *Packets* are like the pages of a book. An entire book contains too much information to be printed on one page, so it is divided into multiple pages. This makes the information in the book much more manageable. The Internet Protocol does the same thing with the information in a file that is to be transmitted over the Internet or a LAN. It divides the information into packets that can be handled more easily by the network.

The other primary function of the IP is to provide addresses for the computers connected to the Internet. Each computer needs its own *IP address*, a group of four decimal numbers that provides a unique address for the computer. Examples of IP addresses are 198.68.191.10 and 204.250.144.70. These IP addresses are actually decimal representations of single 32-bit binary numbers. While a computer may be comfortable with 11000110 01000100 10111111 00001010 or 11001100 11111010 10010000 01000110 as an address, most people find decimal numbers easier to work with. This system of numbering allows for about 4.3 billion (2^{32}) possible combinations. If you are setting up a web server, you will need to get an IP address. You may also need a domain name. A *domain name* is a unique name that identifies a computer or network, much like the name of a city in a postal address, and is matched to a unique IP address. Examples of domain names are “microsoft.com” and “netscape.com.” The .COM extension identifies the domain as commercial. Other domain extensions include .ORG for nonprofit organizations and .GOV for government agencies. These are assigned by the Internet Network Information Center (InterNIC, <http://www.internic.net>). Your

Internet service provider (ISP) can help you get an IP address and domain name. You can also have your web site located as part of an existing domain. If you will be using an existing web server, the network administrator or webmaster will be able to tell you what the IP addresses are.

Transmission Control Protocol

While the Internet Protocol provides the basics for sharing information over the Internet, it leaves some things to be desired. The two most important are ensuring that all the packets reach their destination and that they arrive in the proper order. This is where the Transmission Control Protocol (TCP) steps in. To understand how it works, assume you want to send a book to someone and you have to mail it one page (or packet) at a time. Also assume that there are no page numbers in the book.

How will recipients know that they have received all the pages, and how will they know the proper order of the pages? TCP solves these problems by creating an “envelope” for each packet generated by the IP. Each envelope has a serialized number that identifies the packet inside it. As each packet is sent, the TCP assigns it a number that increases by 1 for each packet sent. When the packets are received, the numbers are checked for continuity and sequence. If any numbers are missing, the receiving computer requests that the missing packet be re-sent. If the packets are out of sequence, the receiving computer puts them back in order. TCP also makes sure the information arrives in the same condition it was sent (that the data was not corrupted in transit).

TCP/IP provides the basic tools for transferring information over the Internet. The next layer up the ladder is the Hypertext Transfer Protocol, the traffic director for the Web.

Note

To set up an intranet on a LAN, you must add the TCP/IP protocols to the existing networking protocols—possibly either IPX/SPX (Integrated Packet Exchange/Sequenced Packet Exchange) or NetBEUI (NetBIOS Extended User Interface)—on both the server and all clients. Chapter 23 tells you how to do this.

Hypertext Transfer Protocol

The Hypertext Transfer Protocol (HTTP) is the heart of the World Wide Web and is also used with an intranet. HTTP composes the messages and handles the information that is sent between computers on the Internet using TCP/IP. To understand how HTTP works, you first need to understand the nature of client/server relationships.

Client/Server Relationships

The basic function of the Internet or a LAN is to provide a means for transferring information between computers. To do this, one computer (the *server*) will contain information and another (the *client*) will request it. The server will process a client's request and transfer the information. The server may be required to process the request before it can be filled. For example, if the request is for information contained in a database, such as a request submitted to the AltaVista web search engine, the server would first have to extract the information from the database before it could be sent to the client.

The passing of information between a client and a server has four basic steps:

1. A connection is made between the client and the server. This is handled by TCP/IP.
2. The client sends a request to the server in the form of an HTTP message.
3. The server processes the request and responds to the client, again in the form of an HTTP message.
4. The connection between the client and the server is terminated.

TCP/IP creates the connection between the client and the server, and HTTP composes the request for information. The response from the server is composed as an HTTP response. This is how HTTP is used to transfer information over a TCP/IP connection.

To access information on the Web or an intranet, you need an application that can send requests to a server and that can process and display the server's response. This is the function of web browsers. The two most common web browsers are Microsoft's Internet Explorer and Netscape's Navigator/Communicator.

Note

With the release of versions 4.0, both Microsoft's browser, Internet Explorer, and Netscape's browser, Navigator, have grown into integrated suites of applications. Microsoft has kept the name Internet Explorer for their browser suite while Netscape renamed theirs Communicator. Netscape Navigator is available both as a stand-alone browser and as part of Communicator. The term Netscape Navigator is used to refer both to the stand-alone Navigator and the browser component of Communicator.

Hypertext Markup Language

The parts of the Web or an intranet covered so far, TCP/IP and HTTP, control how information is transferred over the network. Hypertext Markup Language (HTML) is the component that controls how the information is displayed. The information sent from a web server is an HTML document. Here's what a simple HTML document looks like:

```
<HTML>
<HEAD>
<TITLE>A Simple HTML Document</TITLE>
</HEAD>
<BODY>
<H1>A Simple HTML Document</H1>
<P><B>This text is bold</B> and <I>this text is italic</I>.</P>
</BODY>
</HTML>
```

Figure 1-1 shows how a web browser displays this HTML document. Web browsers interpret the HTML document and display the results on your monitor. HTML files are simple ASCII text files that contain formatting tags that control how information (text and graphics) is displayed and how other file types are executed (audio and video files, for example).

HTML tags are usually used in pairs. An HTML document must begin with the `<HTML>` opening tag and end with the `</HTML>` closing tag. The `<HEAD></HEAD>` tags enclose information about the web page, such as the title, which is defined by the `<TITLE></TITLE>` tags and displayed in the title bar of the web browser. The body of the web page is enclosed by the `<BODY></BODY>` tags. The `<H1></H1>` tags define the enclosed text as a level-1 heading. The `<B></B>` and `<I></I>` tags, respectively, define text as bold or italic, as you saw in the listing above. Each paragraph is usually enclosed in the `<P></P>` tags.

Note

HTML tags are not case sensitive. They can be upper- or lowercase, or mixed case, such as `<Body>`.

In the early days of the Web, these tags were typed in by use of a simple text editor to create web pages. This was time-consuming and not much fun. Today you can use

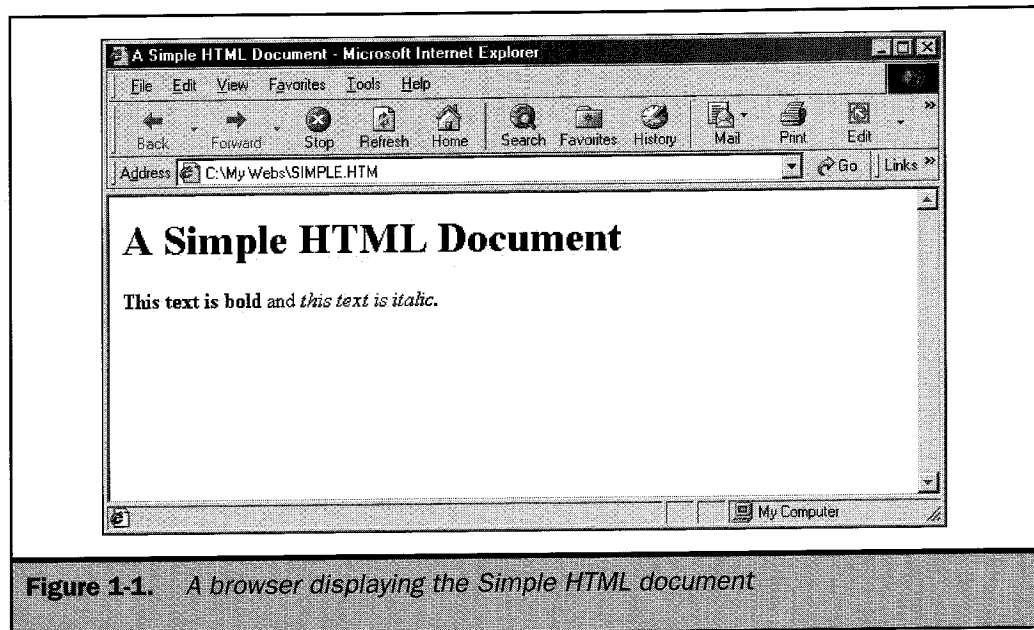


Figure 1-1. A browser displaying the Simple HTML document

FrontPage, a true WYSIWYG (what you see is what you get) HTML editor, to create your web pages. Gone are the days when you had to learn all the HTML tags and proper syntax. With FrontPage you design a page, and the proper HTML is created automatically. It's never been easier to create your own web site.

What Is a Web Page?

A web application consists of one or more web pages that are interconnected. Since the focus of this book is how to create web applications that are collections of web pages, here's a more detailed definition of what a web page is. A good starting place is that a *web page* is a text file containing Hypertext Markup Language (HTML) formatting tags, and links to graphics files and other web pages. The text file is stored on a *web server* and can be accessed by other computers connected to the server, via the Internet or a LAN. The file can be accessed by the use of *web browsers*—programs that download the file to your computer, interpret the HTML tags and links, and display the results on your monitor. Another definition is that a web page is an interactive form of communication that uses a computer network.

There are two properties of web pages that make them unique: they are interactive and they can use multimedia. The term *multimedia* is used to describe text, audio, animation, and video files that are combined to present information—for example, in an interactive encyclopedia or a game. When those same types of files are distributed over the Internet or a LAN, you can use the term *hypermedia* to describe them. With the World Wide Web it is now possible to have true multimedia over the Internet. However, unless your clients have a high-speed service, such as the Integrated Services Digital Network's (ISDN) 128-Kbps service or Digital Subscriber Line (DSL) from 56 Kbps to over 1 Mbps, downloading the large hypermedia files can take too long to routinely use them. On most LANs, which are considerably faster, this is much more feasible, but there are still limitations and a potential need to keep the LAN open for high-volume data traffic.

Web pages are interactive to allow the reader or user to send information or commands back to the web site that hosts the application web. For example, Figure 1-2 shows the home page of the AltaVista web search engine. The AltaVista home page gives you access to an application that searches the AltaVista database of web sites. You can use this and other search engines to locate sites on the Web. From this web page you can select which part of the Internet to search, how the results of the search will be displayed, and the keywords that the search will be based on. When you click on the Search button, the information you've entered is sent to the AltaVista web server. The database is then searched, and the results are used to create a new web

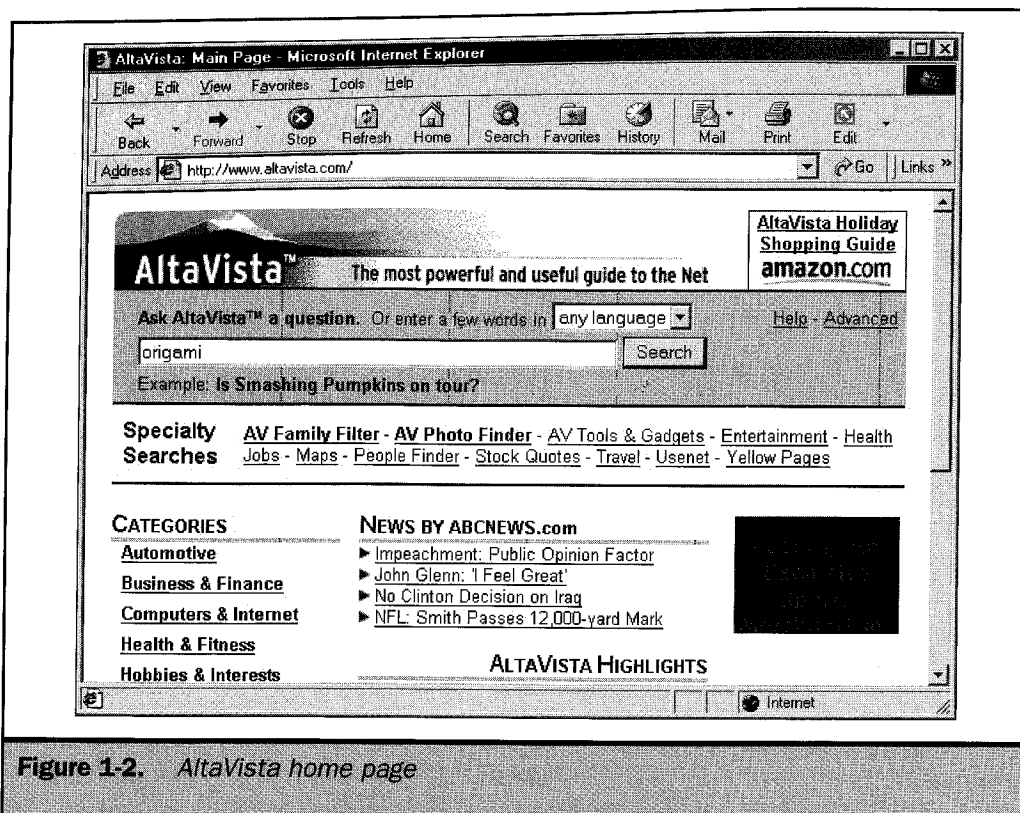


Figure 1-2. AltaVista home page

page, which is displayed by your web browser. Figure 1-3 shows the results of a search using the keyword “origami.”

Each web page has an address called the *uniform resource locator* (URL). The URL for the AltaVista home page is <http://www.altavista.com>. The URL is displayed in the Address combo box (a combination of a text box and a drop-down list) at the top of the screen (below the toolbar). A URL is the path on the Internet to a specific web page. It is used in the same way you use a path name to locate files on your computer. In this case, the URL tells you that the web page is located on a web server with the domain name *altavista.com* connected to the World Wide Web (“www”). A *domain* is one or more networked computers; the domain name provides a single address to access the network from the Internet. The network’s domain server routes the request to the correct place within the network. The actual filename of the *home page* (the top-level page of the web, which usually serves as a table of contents for the web) is usually either *Default.htm* or *Index.htm*; it is implied by being unstated. The default web page filename is set on the web server. This is the page that will be displayed if no web page

The Complete Reference



Chapter 19

Activating Your Webs

689

The World Wide Web has come a long way since the ability to create hyperlinks between documents was a revolutionary technology. Today's web browsers support animation, video, and audio files integrated in webs, but even these advances have been overshadowed. You've seen how JavaScript, VBScript, and Active Server Pages can produce true interactivity on the Web. Java and ActiveX further expand your horizons. Java applets and ActiveX controls are computer programs that can be downloaded to and then executed on your computer by your web browser, or can be run on the server in the same manner as server-side scripting. These technologies are truly revolutionary and are the future of world-class web sites.

In this chapter, you will first learn how current web browsers support HTML features such as *marquees* (text that scrolls across a web page) that do more than statically display information, and how these are different from Java applets and ActiveX controls. Then you will learn how to use these newest web technologies in your own webs. This is the frontier of the World Wide Web.

Active Browser Features

As you've learned in the previous chapters, a web page is a text file containing HTML instructions. Your web browser loads the HTML file and creates the web page displayed in the browser by interpreting the HTML instructions. In Chapter 20, you will see how to add multimedia to your web pages by using some of the newest HTML tags. However, these features only work with web browsers that support these HTML tags. This means that there often needs to be a cycle of browser upgrades before the features are widely supported. Even after browser support is available, not everyone downloads the latest version of his or her favorite browser promptly (particularly as browser downloads have become quite large). This all means that you use these features at your own risk—only a small portion of your audience may see your webs the way you intended.

Java, ActiveX, JavaScript, and VBScript are fundamentally different. If a browser supports these programming languages, as both Netscape Navigator and Microsoft Internet Explorer do to some extent, what you can do on a web page is limited only by your imagination. This frees the web designer from waiting for browser upgrades to begin using the latest web features. The resulting stability is essential as the Web grows. One of the biggest problems in creating webs is the dependence on the user's web browser. Almost every current browser can support the standard HTML features, such as displaying tables, to one degree or another, but support of multimedia varies greatly. Since Netscape and Microsoft offer conflicting HTML tags, the situation will not resolve itself soon.

With Java and ActiveX, the problem is simply sidestepped. You can either write your own applets and controls (which requires a good knowledge of computer programming), or download them from a number of sources on the Web.

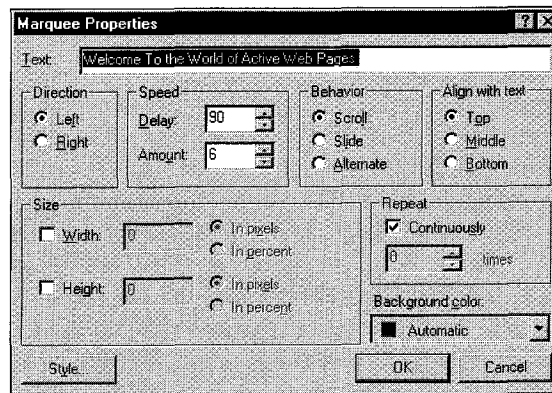
Note

Support for these technologies in current browsers is less than perfect. This situation is rapidly changing, and you can expect that most web users will soon be able to see your activated webs exactly as you intended.

Adding a Scrolling Marquee

A good example of how an HTML tag (interpreted by a browser) differs from a Java applet (a stand-alone executable computer program) can be seen by adding a marquee to a web page. This is an HTML tag currently only supported by Microsoft's Internet Explorer. If you use this feature, it will be lost on anyone using Netscape Navigator. By creating the same effect with a Java applet, you can make it possible for more of the people visiting your web site to see your pages as you intended. To understand the differences, first create a web page with a marquee in FrontPage using the Marquee HTML tag with these instructions:

1. Open FrontPage if it's not already open, and then create a new One Page Web and name it **Active**.
2. At the top of the new page, type **Welcome To the World of Active Web Pages**, and format it as Heading 2.
3. Select the text and open the Insert menu; then choose Component | Marquee. The Marquee Properties dialog box, shown next, will be displayed. The selected text is shown in the Text text box.



In the Marquee Properties dialog box, you specify the text to be displayed; the direction, speed, and type of movement of the text; its alignment; the number of times it repeats; the width and height of the marquee box; and the background color of the marquee.

4. Accept the defaults and click OK.

5. Click Save on the toolbar. Save your page with the Page Title of **Active Web Home Page** and the File Name **Default.htm**. Click Save. You will be prompted that this will overwrite the Default.htm page created with the web. Click Yes.
6. Open your Active Web Home Page in Microsoft Internet Explorer. The text you entered previously will scroll from the right side of the page to the left and repeat continuously.
7. Open Netscape Navigator and load the Active Web Home Page into it. The text simply sits at the top of the page, since the Marquee tag is not supported by Netscape's browser.
8. In FrontPage, right-click the marquee text and select Marquee Properties.
9. In the Marquee Properties dialog box, select the Alternate Behavior option and click OK. Save the page again.
10. In Internet Explorer, click Refresh on the toolbar. The marquee will now travel back and forth between the left and right margins of the page.

As you can see, the Marquee tag can add a little life to a web page, but only if it is supported. Since only Internet Explorer currently supports the tag, this is a feature of limited use. In the next section, you will find other ways to activate your webs that have better cross-platform support.

Java

The Java programming language was developed by Sun Microsystems, Inc., and initially released in fall 1995. Netscape was an early licenser of Java; Microsoft soon followed. With the release of Microsoft's Internet Explorer 4.0, the uneasy alignment between Sun and Microsoft came to an end. Sun has sued Microsoft for alleged violations of their licensing agreement, which Microsoft has denied. Sun claims that Microsoft changed certain core interface components in violation of the agreement. Microsoft did make the changes, but maintains that the licensing agreement allowed them to.

Note

At the end of 1998 Sun was granted a preliminary injunction against Microsoft that has compelled Microsoft to bring its version of Java in line with Sun's. Microsoft's "improvements" are still in its own flavor of Java, but they no longer are the defaults.

Java's proponents see it as a replacement for the Windows operating system. At the current level of development this is a bit of wishful thinking. Microsoft probably wishes Java would go away so everyone would use ActiveX. In all probability, neither side will get what it wants. Regardless of the growing pains the language is experiencing, it is and will continue to be an important language for many network and web applications.

Object-Oriented Programming

Java is an *object-oriented* programming language very similar to C++, one of today's standard programming languages. An object-oriented programming language defines an *object* as a process that accepts information, processes it, and then outputs the result of the processing. The format of the input is always clearly defined, as is the output. One object may receive its input from the output of another object. For example, an object may be a simple program that accepts a text string and then converts the text to all capitals. The input is the text string, regardless of case, and the output is the same text string converted to all uppercase. The object is the code that performs the conversion. Java applets are built by combining a number of objects, each performing a relatively simple task, into a computer program that can carry out complex operations. This concept is very powerful, as you will learn in the following sections.

Note

Programming your own Java applets is a complex subject, well beyond the scope of this book. Two excellent resources for learning more about object-oriented programming and Java are The Java Handbook by Patrick Naughton (Osborne/McGraw-Hill, 1996) and Java 2: The Complete Reference by Patrick Naughton and Herbert Schildt (Osborne/McGraw-Hill, 1999). There are many preprogrammed Java applets available on the Web, so programming your own applet is not necessary for you to add Java applets to your web. The examples in this chapter will use readily available Java applets.

By its nature, Java offers several features besides object orientation that make it suitable for programming on the Web. It's also safe, robust, interactive, platform independent, and high performance.

Safe

Since Java applets are computer programs that are downloaded and executed on the user's computer, what's to stop the unscrupulous programmer from sending a destructive applet over the Web to wreak havoc on thousands of computers? With the rapid growth of the Internet, this has become a leading concern for many, including the creators of Java. Their solution was to strictly limit what a Java applet can do. Java applets cannot access or misuse operating system resources, which leaves little room for vandalism. It is possible, for example, to write an applet that will slow down your computer by monopolizing resources, but this does not cause permanent damage.

Robust

With millions of users around the world connected to the Internet, any program that is written to be used there must be able to function flawlessly on many different computers with unique configurations. By Java's nature, these problems are kept to a minimum. Provided the browser includes support for the Java language, applets can usually be depended on to function properly.

Interactive

Most web sites today display information passively; that is, the content is defined by the web author and displayed by a browser in much the same way as a page in a magazine is produced. In Chapter 18, you saw how databases could be integrated in webs so that information could be presented dynamically. Java and the other technologies covered in this chapter take the process a step further. As you will see in the examples in this section, Java applets, because they are computer programs running on the user's computer, can accept input from the user, process it, and display the output on a web page. All this can be done locally (on the user's computer) and so leaves the web server free to handle other tasks.

Platform Independent

On the Internet (and intranets), computers running Windows, Apple's Macintosh operating system, and UNIX (and its variants such as Linux) can coexist along with a few other minor operating systems. One of the beauties of HTML and the Web is that all these systems can access and display the same web pages. Java extends this platform independence by creating code that does not rely on a specific operating system. Each browser that supports Java applets contains a Java interpreter that handles the interaction between the applet itself and the computer operating system. In this way a single Java applet will function properly on a Windows PC, a Mac, or a UNIX computer. This is a capability with far-reaching implications. Up to this point, computer programs have been written to run only on a single operating system. With Java, it is now possible to write complex applications that can run on any operating system for which an appropriate browser is available.

Note

Though Sun's claim that Java is "write once, run anywhere" has often been rephrased to "write once, debug everywhere," it is still a quantum leap forward in platform independence.

High Performance

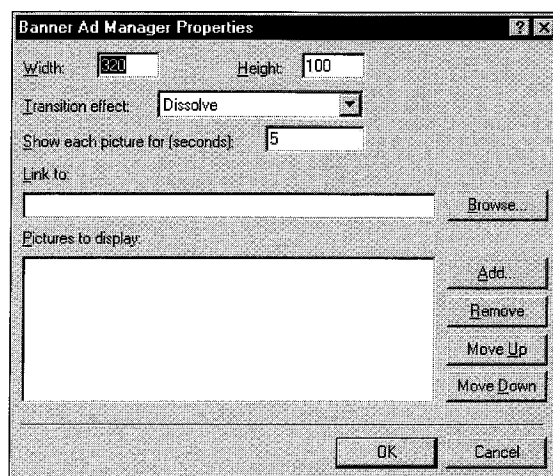
All the features of Java applets mentioned so far would be of limited use if the applets were not high performance. Almost every computer user has experienced the frustration of slow response times when running some applications. Java's creators made sure that the Java code would work efficiently even on older, slower computers. Also, both Netscape and Microsoft have worked to make the Java interpreters in their web browsers as fast as possible.

The features just described combine to make Java the first of a new generation of programming languages created to work efficiently and flawlessly across the Internet or a company intranet. Now that you have some understanding of just what Java is and can do, it's time to add an applet to your web.

Banner Ad Manager Component

You need look no farther than FrontPage's Insert menu to find your first Java applet. The Banner Ad Manager is a Java applet at heart. In Chapter 17, you learned how to use the ASP Ad Rotator Component to rotate ad banners on your site. The FrontPage Banner Ad Manager performs a similar task with different options. Add your first Java applet to your Active home page by first importing the graphics to display from the included CD.

1. In the Folder List, select your Images folder in your Active web.
2. Open the File menu and choose Import. In the Import dialog box, click Add File.
3. In the Add File To Import List dialog box, select your CD drive and then open the Chap19 folder. Select Banner1.gif, press and hold SHIFT, and then select Banner2.gif. Release SHIFT and click Open and then OK.
4. In Page view place your cursor on the first line below your Marquee text; then open the Insert menu and choose Component | Banner Ad Manager. The Banner Ad Manager dialog box shown here is displayed.



In the Banner Ad Manager dialog box, you set the size of the graphics to display, the transition effect to be used to change the graphic displayed, the length of time to display each graphic, a hyperlink for the graphics, and select the graphics to be used.

5. Type 468 in the Width text box and 60 in the Height text box.

6. Select Dissolve in the Transition Effect drop-down list, and accept the default of 5 seconds in the Show Each Picture For (Seconds) text box.
7. In the Link To text box, type **http://www.pbg.mcgraw-hill.com/computing/**. (Do not include the period.)
8. Click Add to add the graphics to display in the Pictures To Display list box.
9. In the Add Picture For Banner Ad dialog box, open your Images folder. Select **Banner1.gif** and click OK.
10. Click Add; then select **Banner2.gif** and click OK twice.
11. Save your work and then open your Active web home page in your browser. Figure 19-1 shows the page in Netscape Navigator in mid-transition.
12. In FrontPage, right-click the graphic and select Banner Ad Manager Properties.
13. Select Blinds Horizontal from the Transition Effect drop-down list and click OK.
14. Save your page and then reload it in your browser to see the effect.

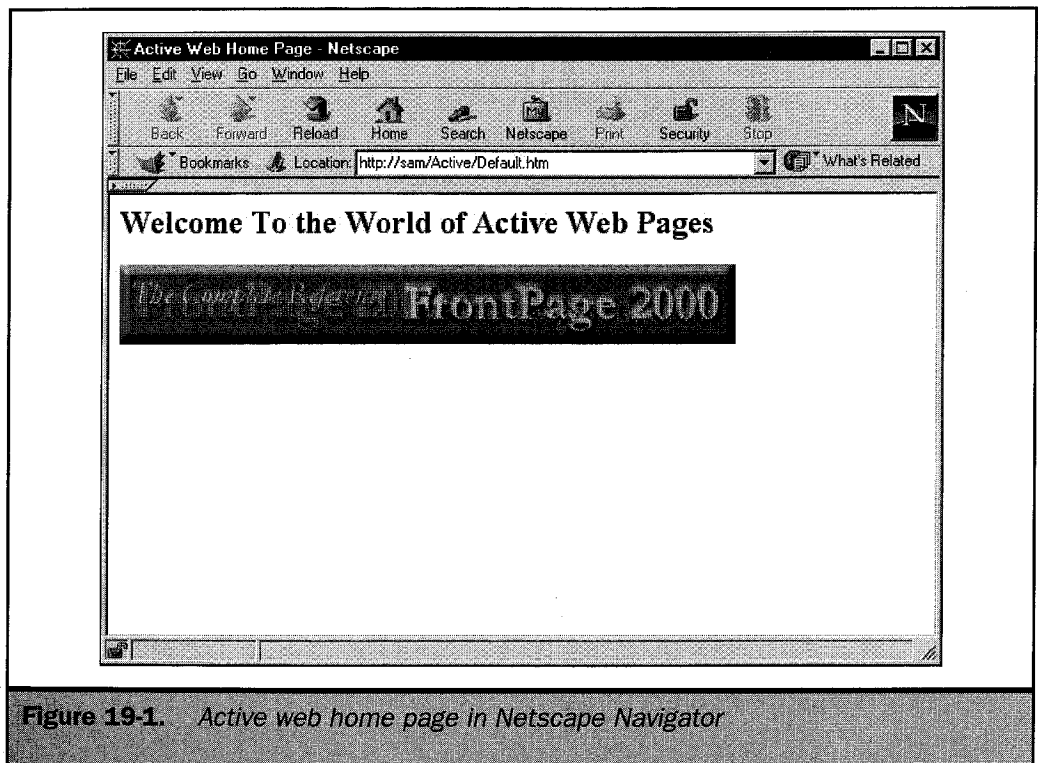


Figure 19-1. Active web home page in Netscape Navigator

The key difference between the ASP Ad Rotator Component and the FrontPage Banner Ad Manager Component is that the ASP component rotates banners each time the page is loaded (server-side) and the FrontPage component rotates them continuously because it is running in the browser (client-side). Another difference is that the ASP component is an ActiveX control, while the FrontPage component is a Java applet. Since only Internet Explorer supports ActiveX controls in the browser, it's necessary to use Java applets when you need cross-browser support. On an intranet, where you can control the browser in use, you could use ActiveX controls on the client. On the Web, this is impossible unless you don't care about the user experience for a large portion of your audience.

The transition effects you see are possible with both technologies, so in that respect neither offers a significant advantage over the other (though you will find programmers in both camps who will disagree with that statement). The Java applet also limits you to a single URL for the hyperlink, while the ASP component allows a separate URL for each graphic to link to. This is not an inherent limitation of the Java language, however. It is simply how the applet was written.

15. In FrontPage Page view, click the HTML tab. Here you see the HTML that displays the Banner Ad Manager applet, as shown in Figure 19-2.

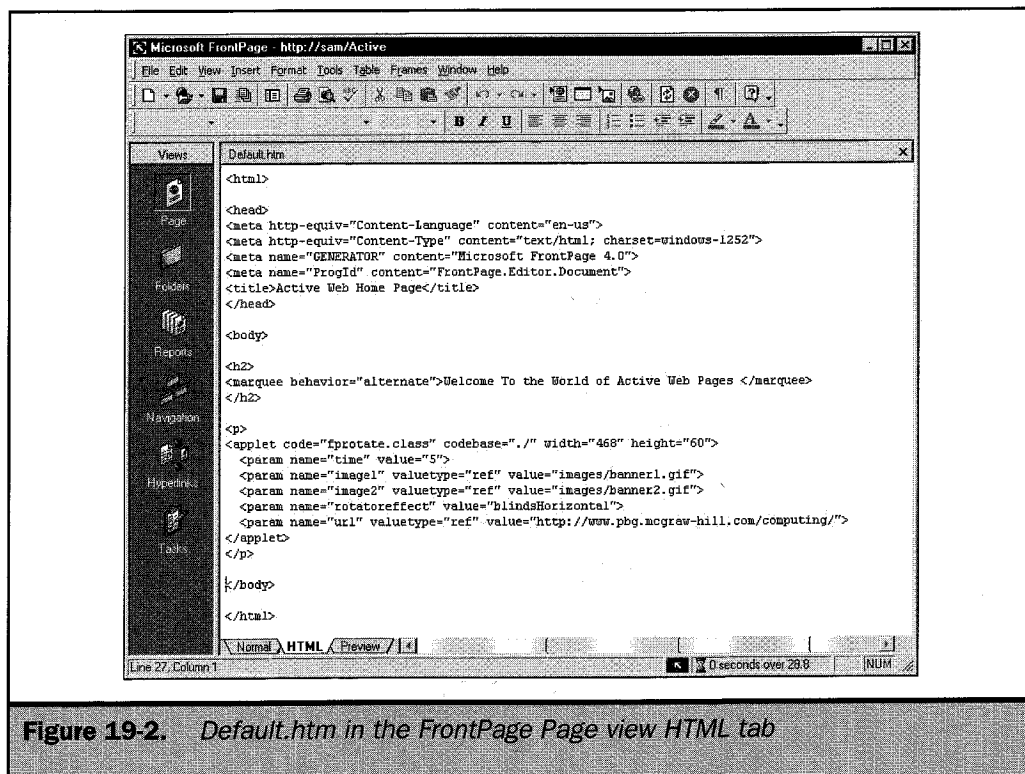


Figure 19-2. Default.htm in the FrontPage Page view HTML tab

The <APPLET> tag is used to identify the Java applet using the parameters listed in Table 19-1. There will usually be a series of <PARAM> tags between the opening and closing <APPLET> tags. These are name-value pairs that contain the values passed to the applet. The correct names and the types of values are determined at the time the

Property	Description
Align	Specifies the alignment of the applet. The values are the same as for the tag.
Alt	Is the alternative text that is displayed for that applet. This is used in the same manner as the Alt property for the tag.
Archive	Is the location of a compressed file containing the applet and any other files it needs, such as multimedia files. This property is Netscape specific.
Class	Identifies a style sheet to use with the applet.
Code	Is the name of the applet. The location is relative to the Codebase.
Codebase	Is the URL of the directory that contains the applet.
Datafld	Is the column name from a data source that the applet is bound to.
Datascr	Is the data source that the applet is bound to.
Height and Width	Is the height and width in pixels of the applets display area.
Hspace and Vspace	Set the horizontal and vertical space around the applet in pixels.
ID	Is a unique identifier used with a style sheet or to reference it in a script.
MayScript	Is a flag that indicates if the applet can be scripted using JavaScript functions.
Name	Is a unique name for the applet that can be used by other applets on the same page to interact with the applet.
Src	Is a URL that can point to resources used by the applet.
Style	Identifies any inline styles to be applied to the applet.
Title	Is a title for the applet. This is Internet Explorer specific and is displayed as a Tool Tip when the applet is pointed to.

Table 19-1. *Applet Tag Properties*

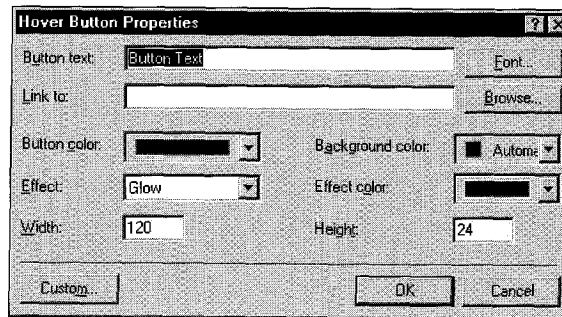
applet is written, so there are no standard names. The values that can be passed to the applet should be explained in the documentation for it. In this case, using the FrontPage Banner Ad Manager, the parameters are exposed in the Banner Ad Manager dialog box.

FrontPage contains another Java applet accessible by selecting the Insert menu and choosing Component | Hover Button, which you will use in the next section.

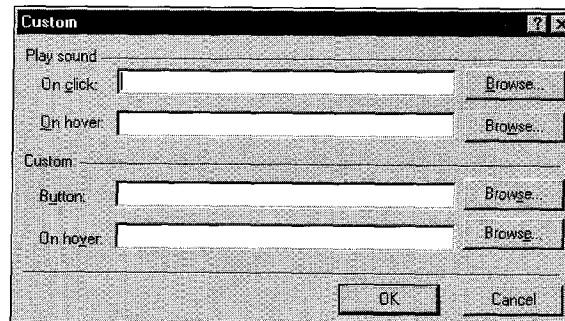
Creating a Hover Button

A *hover* button is one that changes in some way or that controls an action triggered when the user moves the mouse pointer over it. The easiest way to understand how it works is to create one. Do that now with these steps:

1. Click the Page view Normal tab and place the insertion point on the third line.
2. Open the Insert menu and choose Component | Hover Button. The Hover Button Properties dialog box shown next is displayed.



In this dialog box, you can set the text for the button, a hyperlink for it, its colors and size, and the effect. Clicking Custom displays the Custom dialog box, shown next, where you can set a sound to be played, or a custom image to be displayed when the mouse pointer moves over the button.



3. Accept the defaults in the Hover Button Properties dialog box by clicking OK.
4. Save your page, refresh it in your browser, and move your mouse pointer over the Button Text graphic to see the effect of the applet.

This example shows how easily a Java applet can be used. You can look at the HTML for the Hover button in the HTML tab to see the similarities between it and the Banner Ad Manager. The next step is understanding how Java applets are created. First, you need to download and install some Java development tools.

Finding Java Applets

Unless you're an experienced C++ programmer, your first Java applets will probably be ones that you download from the Web. There are a growing number of web sites that have applets available for downloading. The first step, then, is to get onto the Web and find some Java applets. A good place to start is Sun Microsystems' Java home page. The following steps will take you there.

Note

Web sites that offer Java applets change often as new applets are added. The descriptions of the web sites in this book reflect their status at the beginning of 1999, when this book was written.

Tip

The current versions of both Netscape Navigator and Internet Explorer support Java, but there are differences. The current turf war between Sun and Microsoft has made it even more important to have the latest versions of Netscape's Navigator and Microsoft's Internet Explorer browsers and to test your applets with both the newest and older browser versions.

1. Make sure your Internet connection is functioning, and open Netscape Navigator.
2. In the Location text box, type **java.sun.com/applets/** and press ENTER. In a few moments the web page shown in Figure 19-3 should be displayed in your browser.

Tip

Web sites change often, and URLs sometimes become outdated. If the URLs used in this chapter no longer work when you try them, start from the home page and follow the hyperlinks to the resources described.

3. Scroll down the page until you see the heading "Freebie Applets." In the first paragraph under the heading, click the "free applets available for use" hyperlink.

Note

The file you download will be compressed and you will need to extract the individual files using a Zip utility such as WinZip from Nico Mak Computing, Inc.

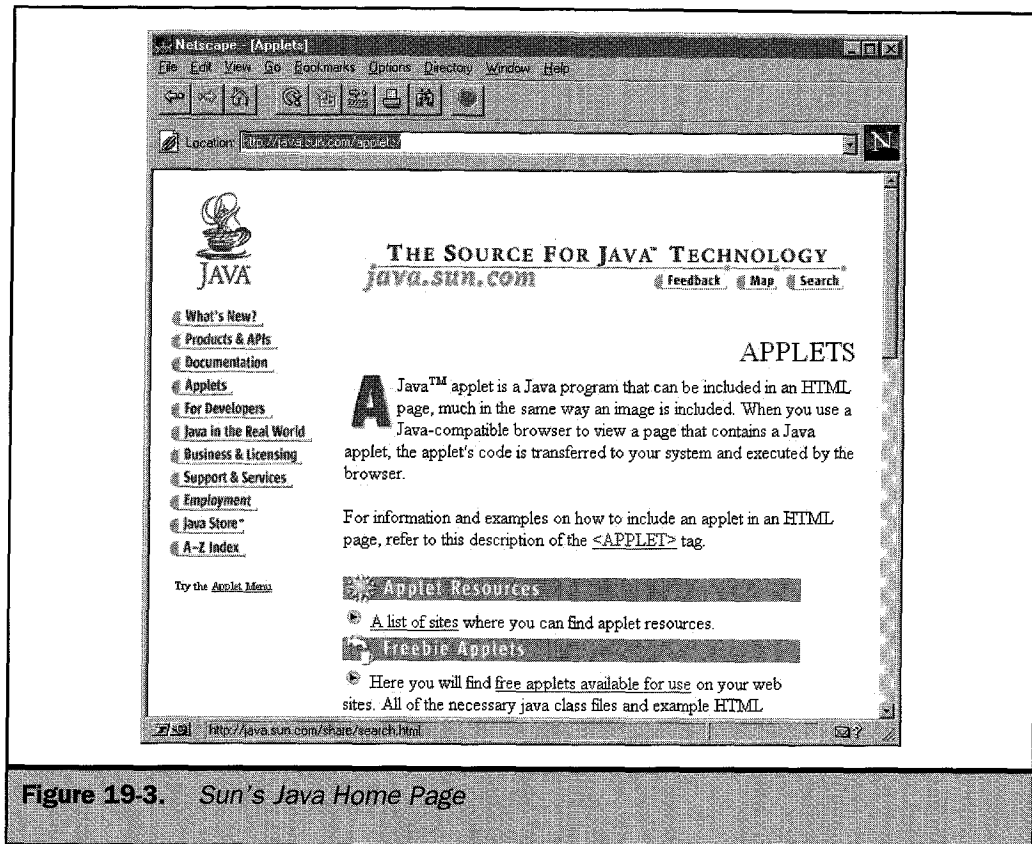


Figure 19-3. Sun's Java Home Page

4. When the Freebie Applets You Can Use web page is displayed, scroll down to the Clock hyperlink and then click it. This takes you to the page (<http://java.sun.com/openstudio/applets/clock.html>) where you can download the applet's files, as shown in Figure 19-4. This page also contains sample HTML code for using the applet and lists the applet's parameters.
5. Click Download Now. If your browser prompts you to open the file or save it, choose to save it to disk. Select a work folder on your hard drive to save the file. You will have to unzip the download in this folder and then place the necessary files in your Active web. The compressed file actually contains several sample applets that you can use, but this exercise will focus exclusively on the Clock applet.

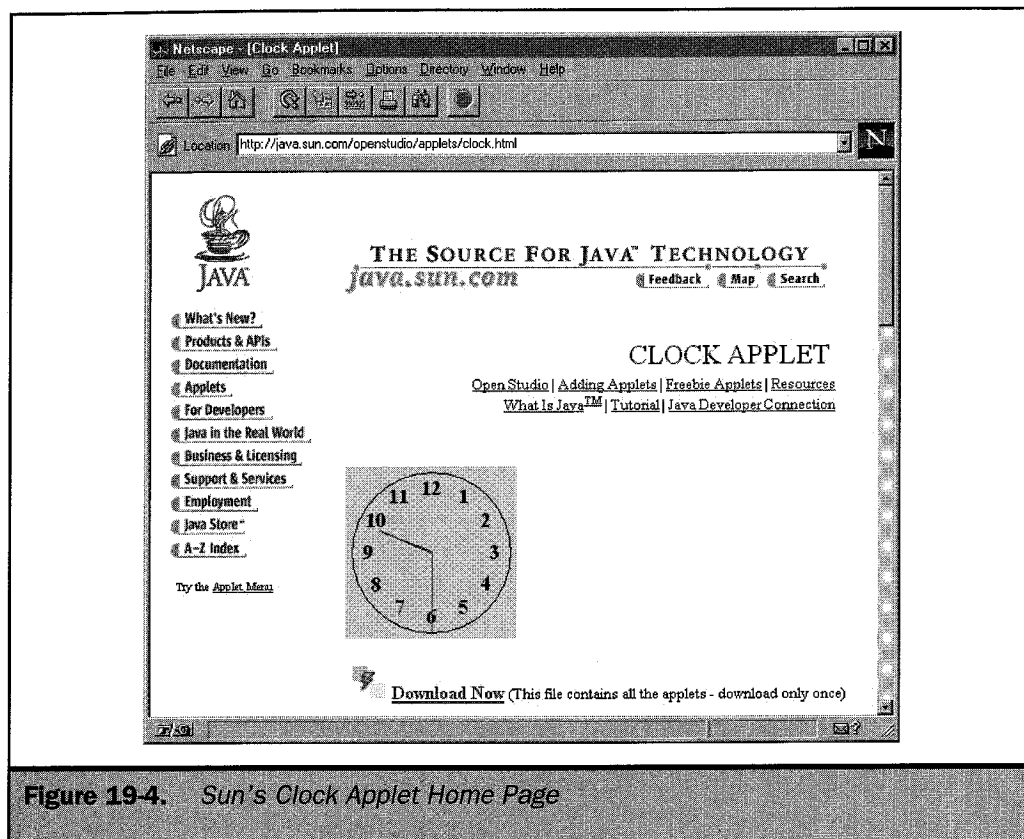


Figure 19-4. Sun's Clock Applet Home Page

6. When you have finished downloading the compressed applet file, you will need to unzip it. Do this in the work folder where you saved the download. When the compressed file is unzipped, it will create a folder named Demo that will contain subfolders for the Clock applet as well as the other sample applets.
7. In FrontPage, import the Clock applet to your Active web by opening the File menu and choosing Import.
8. In the Import dialog box, click Add Folder. In the Browse For Folder dialog box, locate the folder that contains the Demo folder created when you unzipped the file.
9. Open the Demo folder; then select the Clock folder and click OK twice. This will import more files than you absolutely need for this exercise, but is easier at this point than selecting the individual files. When the folder is imported, your Active web should have a folder named Clock that contains two subfolders, Classes and Src, as shown in Figure 19-5.

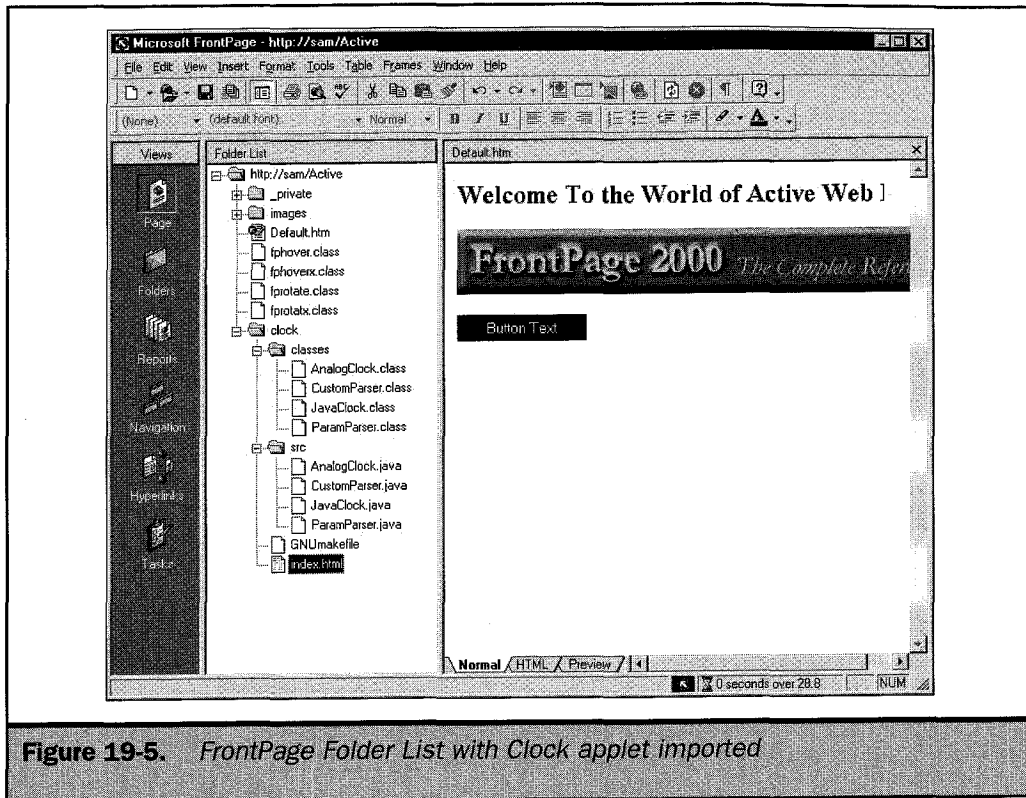


Figure 19-5. *FrontPage Folder List with Clock applet imported*

The Classes folder contains the compiled Java applet files that have a .CLASS file extension. These are the files you will use when you add the applet to your page. The Src folder contains the source code for the applets, and these have a .JAVA file extension. You will learn more about the source files and using them in the section "Compiling Java Applets." Two other files were also imported in the Clock folder. GNUmakefile is a batch file used to create the .CLASS files. You will not use this file. The other file is a web page (Index.html) that displays the Clock applet. Figure 19-6 shows this page in Netscape Navigator.

10. In your Folder List, select the file **GNUmakefile** and press DEL. In the Confirm Delete dialog box, click Yes.
11. In Page view place your cursor on the line below your Hover button.
12. Open the Insert menu and choose Advanced | Java Applet. The Java Applet Properties dialog box, shown in Figure 19-7, will be displayed.

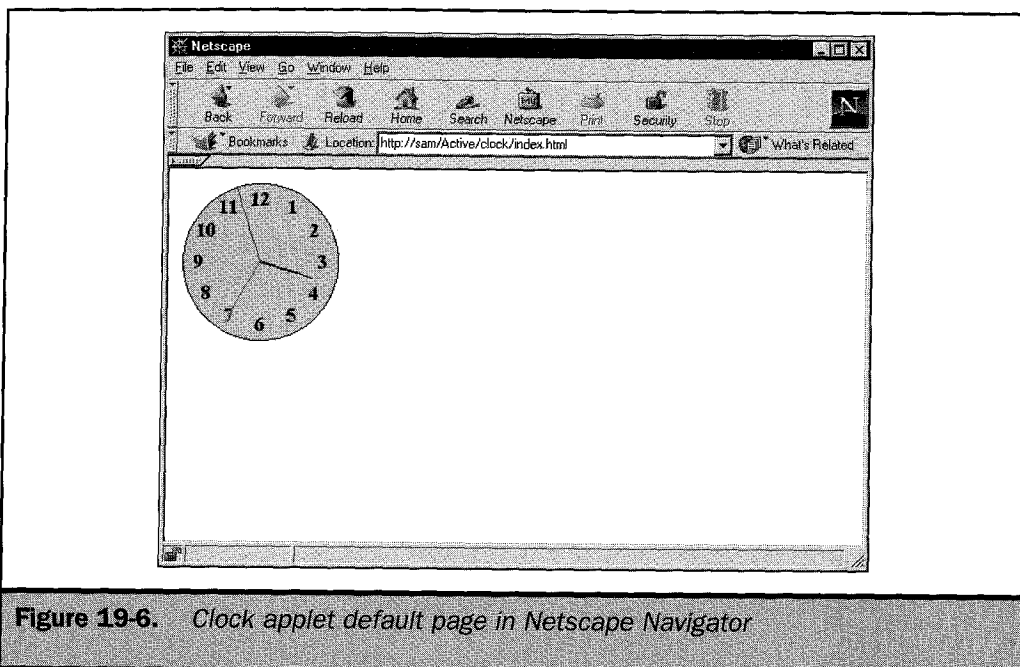


Figure 19-6. Clock applet default page in Netscape Navigator

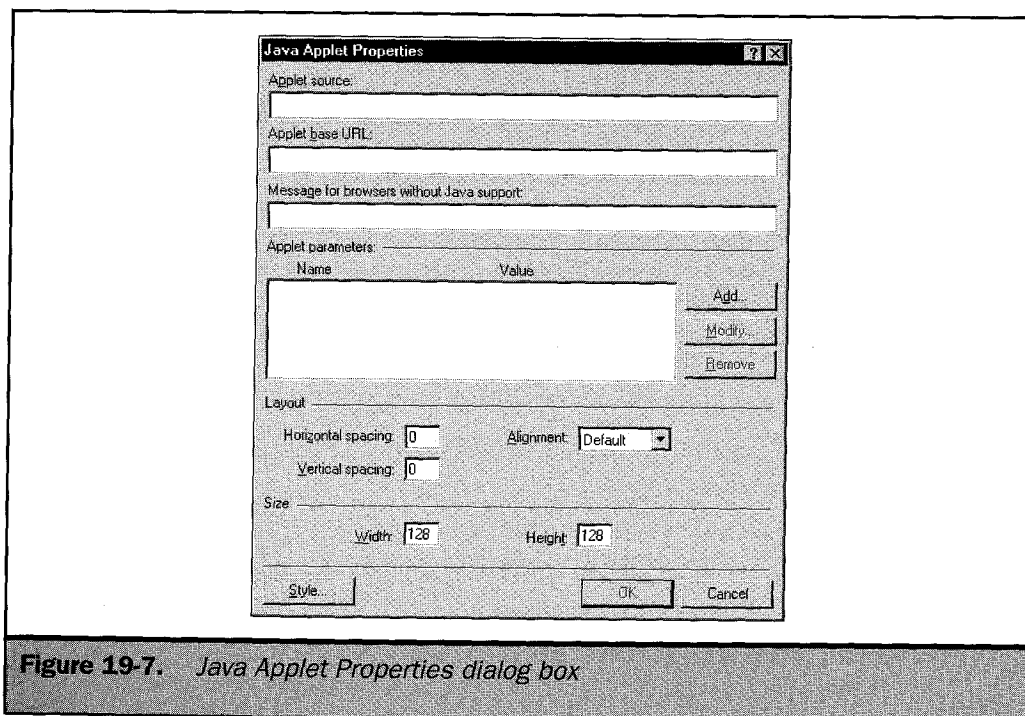


Figure 19-7. Java Applet Properties dialog box

13. Type **JavaClock.class** in the Applet Source text box; then type **Clock/Classes** in the Applet Base URL text box. Click OK. This is the minimum information FrontPage needs to properly display the Clock applet. Figure 19-8 shows the page in FrontPage Page view. FrontPage displays a graphic placeholder for the applet.

14. Save your page, and then open it in your browser. Figure 19-9 shows the page in Netscape Navigator.

Most applets accept additional parameters that control their appearance and/or functionality. The parameters the Clock applet accepts are listed in Table 19-2. In the remainder of this exercise, you will use the Java Applet Properties dialog box to set some of these parameters. The applet will use default values for any parameters that are not specifically set.

15. In FrontPage Page view, right-click the Clock applet and choose Java Applet Properties.

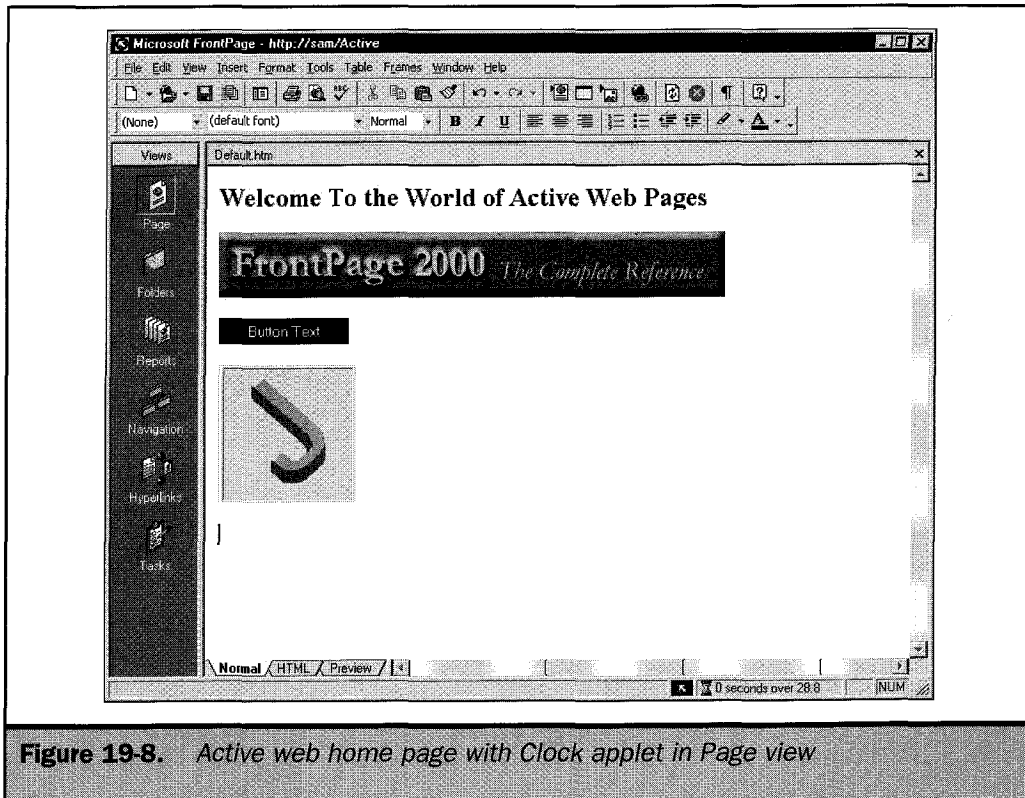


Figure 19-8. Active web home page with Clock applet in Page view

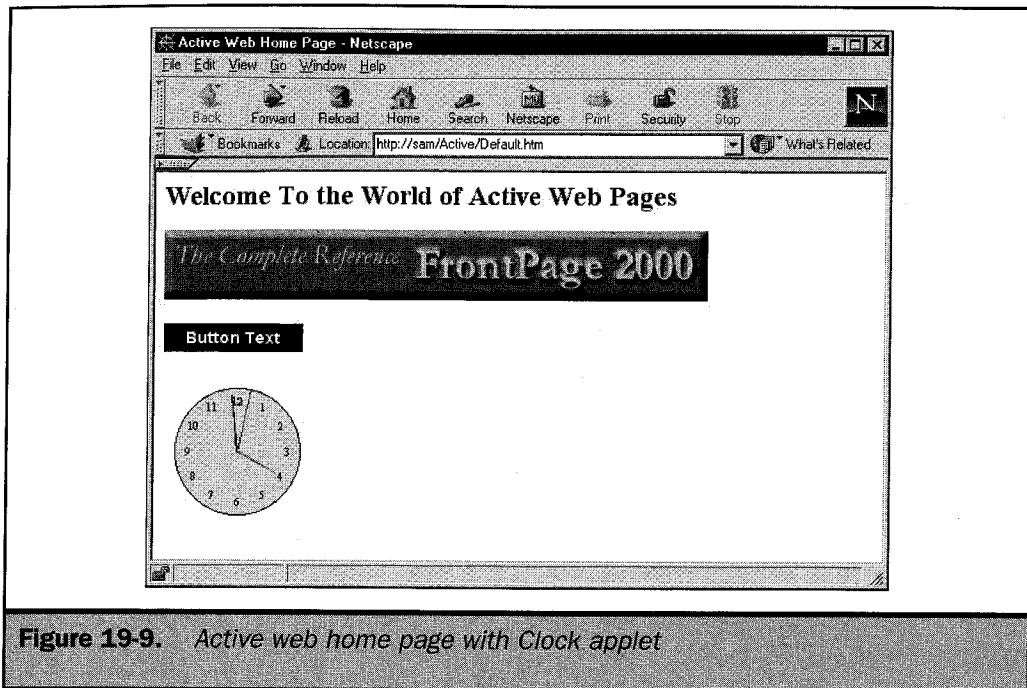


Figure 19-9. Active web home page with Clock applet

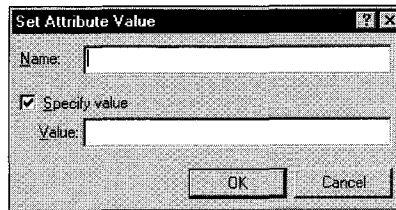
Parameter	Description
bgcolor	Is the RGB background color, in hexadecimal, for the applet.
border	Is the width in pixels of the space around the clock's face.
ccolor	Is the RGB color, in hexadecimal, of the clock's face.
cfont	Is a text string containing the font, style, and point size of the numbers on the clock's face. Each element is separated by the character.
delay	Is the refresh rate in milliseconds for the applet.
hhcolor	Is the hour hand RGB color, in hexadecimal.
link	Is an optional URL if the clock is to be a hyperlink.

Table 19-2. Clock Applet Parameters

Parameter	Description
mhcolor	Is the minute hand RGB color, in hexadecimal.
ncolor	Is the RGB color of the numbers, in hexadecimal.
nradius	Is the radius, in pixels, where the numbers will be drawn.
shcolor	Is the second hand RGB color, in hexadecimal.

Table 19-2. *Clock Applet Parameters (continued)*

16. In the Java Applet Properties dialog box, click Add. This displays the Set Attribute Value dialog box shown next. You will enter the name of the parameter in the Name text box and a value for the parameter in the Value text box. The Specify Value check box should be selected.



17. Type **bgcolor** in the Name text box; then type **0000FF** in the Value text box. This will give the clock a blue background. Click OK.
18. Click Add; then type **cfont** in the Name text box and **Arial | Bold | 18** (there are no spaces in this text) in the Value text box. The vertical bar (the “pipe” character) is entered by pressing SHIFT+\. Click OK.
19. Type **150** in both the Width and Height text boxes and click OK.
20. Save your work and then reload your Active web home page in your browser. Figure 19-10 shows the page in Netscape Navigator.

For any applets you have that are compiled (you have a file with a .CLASS extension), this is basically all you need to do to add a Java applet to your FrontPage webs. The next step is understanding how Java applets are created. First, you need to download and install some Java development tools.

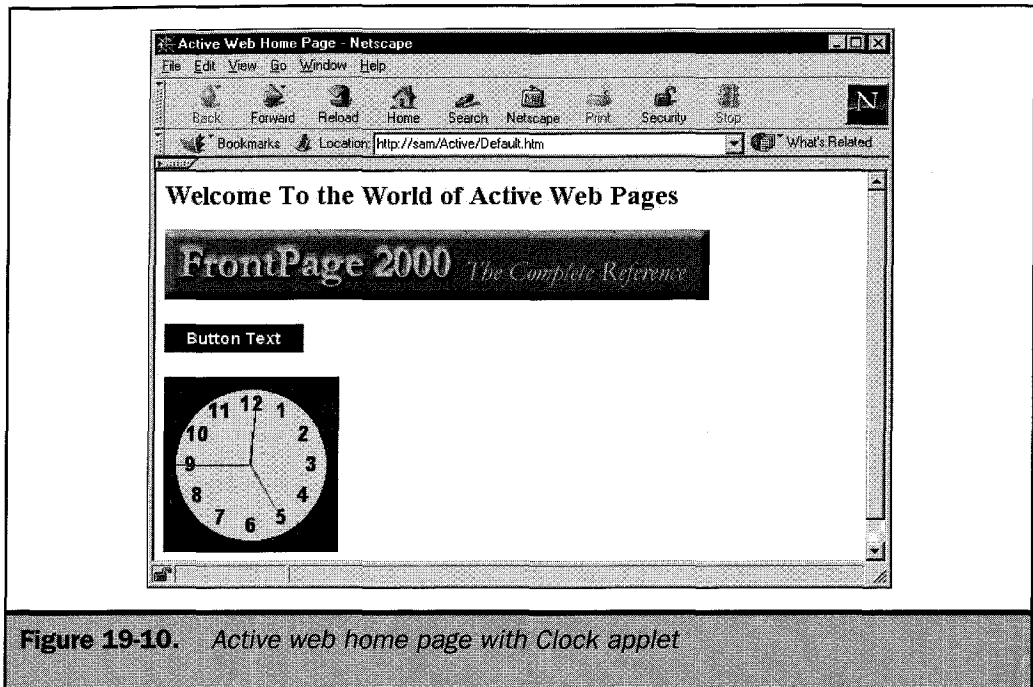


Figure 19-10. Active web home page with Clock applet

The Sun Java Development Kit

Java applets are often distributed as source code that must be compiled before it can be used in your webs. This provides an additional measure of security. Rather than downloading an executable applet that may or may not perform as advertised, you download a text file that is compiled into the executable. This helps ensure that the applet will do what it's advertised to do and nothing more. There are several sources for Java toolkits. The Java Development Kit (JDK), available from Sun Microsystems' web site (http://java.sun.com/products/OV_jdkProduct.html), is the one you will use in the following sections. It contains the source code for a number of sample applets as well as a Java compiler for converting the Java source code into Java applets that can be included in your webs. Another useful toolkit is the Microsoft Software Development Kit (SDK) for Java (<http://www.microsoft.com/java/sdk>). If you plan on really working with Java, you should download both toolkits. They contain extensive documentation that will make working with Java much easier.

Note

There are also a growing number of tools, such as Microsoft's Visual J++, that simplify the process of creating Java applets. If you decide to write your own applets, you should investigate these products. Probably the best place to start looking is Sun's Java Solutions Guide at <http://java.sun.com/solutions/>.

All the basic tools you need for working with Java code are included with Sun's Java Development Kit (JDK). The JDK includes example Java applets, both as source code and compiled applets, one of which you will use in the exercises in the following sections. The documentation files are in HTML format, and the demos also include HTML pages. Begin your exploration of Java by downloading the Java Development Kit.

Note

The JDK is a hefty download (just over 20MB for the JDK and another 16MB for the documentation), but the tools provided are necessary if you really want to work with Java. A full installation of the JDK will require about 65MB of hard drive space without the documentation.

1. In your web browser, go to Sun's Java Development Kit home page at <http://java.sun.com/products/jdk/1.2/>.
2. Scroll down the page until you see the hyperlink for downloading the JDK. At the time this was written, the link text was **JDK 1.2 Windows95/98/NT Production Release**. Click this hyperlink. This will load the first page of the download, shown in Figure 19-11, where you can choose to download the JDK as a single large file or several smaller files.

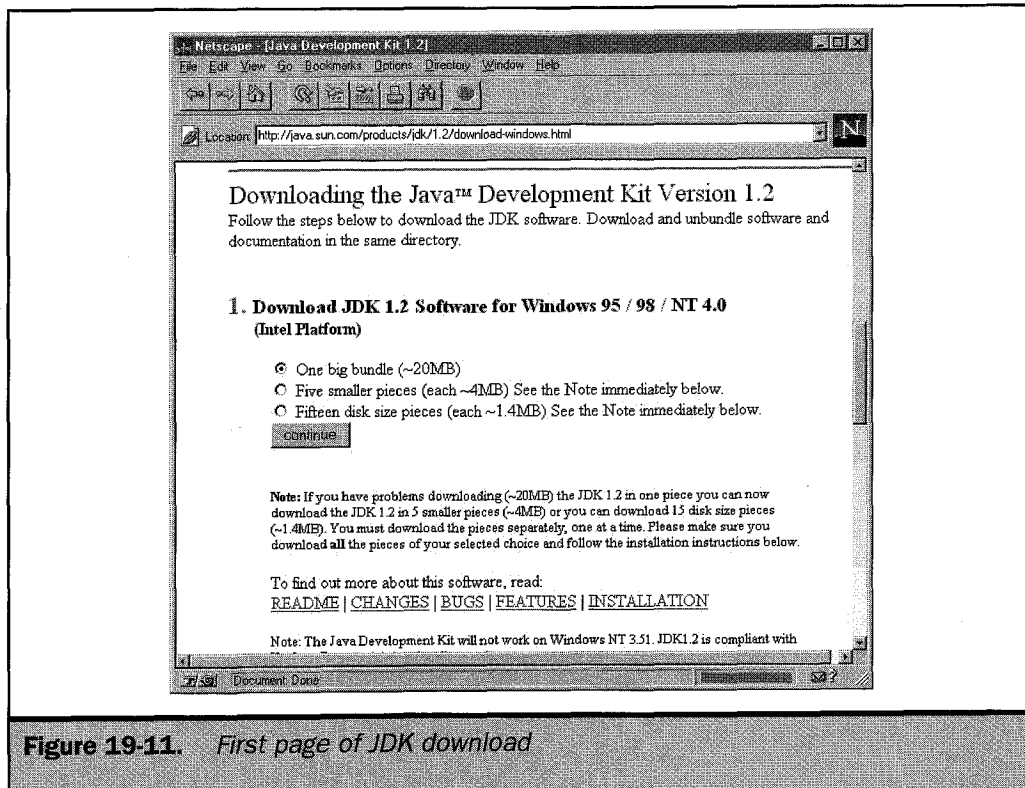


Figure 19-11. First page of JDK download

Note

You can choose to download the JDK as a single download or in several smaller ones. Choose whichever is easier for you. If you download the JDK as several smaller files, you will need to concatenate them into a single file after download. Instructions for concatenating the files are included on the Download page.

3. Select how you want to download the files and then click Continue. This will load the second page for the download, which contains the Sun JDK license agreement, shown in Figure 19-12.
4. After reading the license agreement, click Accept if you accept the terms of the agreement. This will take you to the final page where you can choose to download the JDK using either the FTP or HTTP protocol. Using FTP will probably be somewhat faster, but you may have difficulties if you are behind a firewall. In that case, use the HTTP protocol. This page also contains the installation instructions for the JDK so you may want to print it for reference.

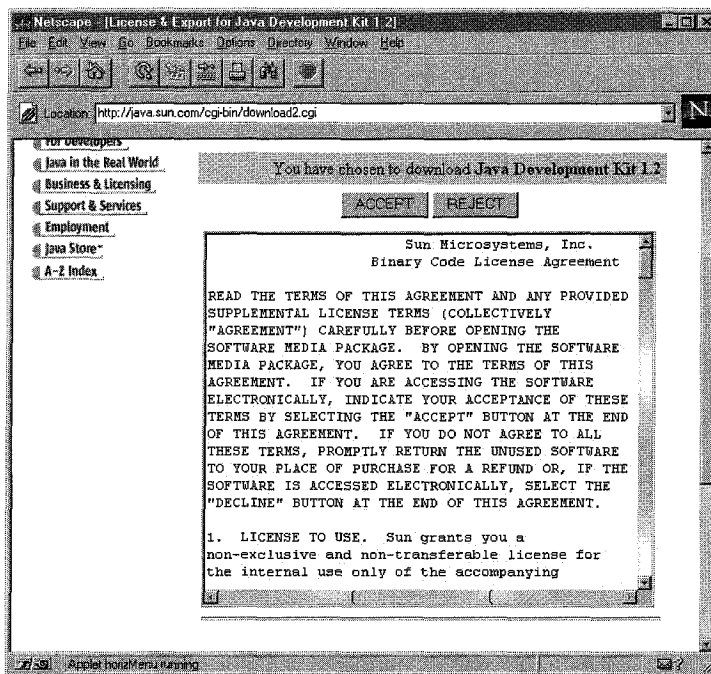


Figure 19-12. Second page of JDK download

5. You will be prompted for a location to save the JDK to. This should be your Temp or a work folder. Select the location to save the file(s) and click OK.

Note

At the time this was written (the beginning of 1999), the most recent release of Sun's JDK was version 1.2. By the time you read this, there may be a more recent version. If so, the filenames will have changed.

6. Next download the documentation. The link to the documentation download page is located by scrolling down the JDK Download page in your browser until you see a series of hyperlinks under the heading "The installation procedure has the following steps:" Click number 8, *Where Do I Go From Here*.
7. You should then see a text hyperlink in the first paragraph, "JDK docs download page." Click this hyperlink. This opens the JDK 1.2 Documentation page.
8. Scroll down the page until you see a drop-down list box that displays Select A Download Package. Select *ZIP File (For Windows Or Non-Solaris) – JDK 1.2 Docs* and click Continue.

Note

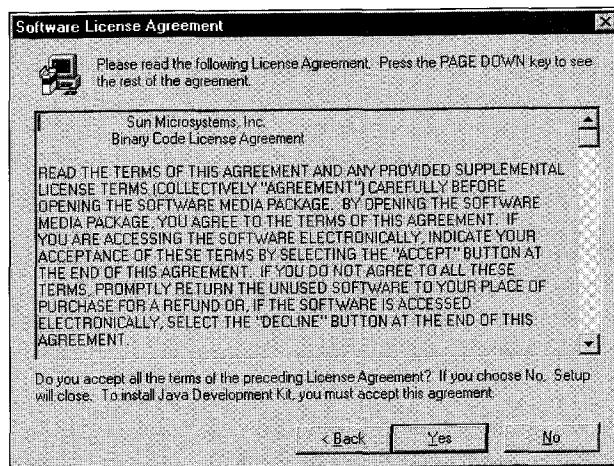
You can choose to download the file as a single download or in several smaller ones. Choose whichever is easier for you. If you download the documentation as several smaller files, you will need to concatenate them into a single file after download. Instructions for concatenating the files are included on the Download page.

9. A page similar to the JDK download page will be displayed. Click either FTP Download or HTTP Download. When prompted, select the same folder you saved the JDK to for the documentation download.

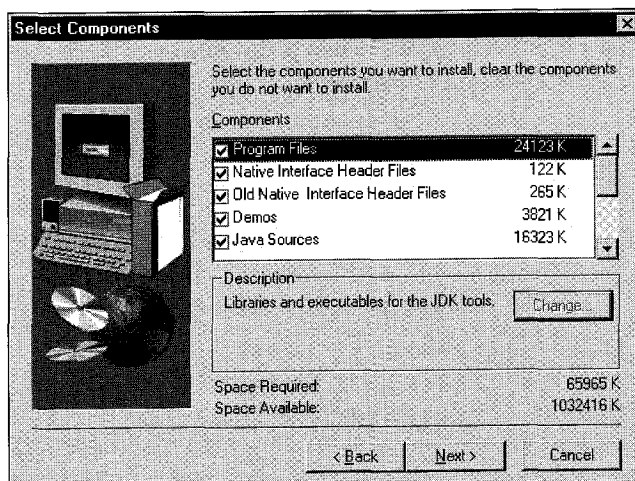
Note

The JDK file is self-extracting, but the documentation files require a compression utility, such as WinZip from Nico Mak Computing, Inc.

10. After the files are downloaded, open the folder in Windows Explorer where you saved the JDK program file (Jdk12-win32.exe) and double-click it. A dialog box will be displayed indicating the files are being unpacked; then the installation program will start.
11. When the setup Welcome dialog box is displayed, click Next. The next dialog box will display the license agreement, as shown next.



12. After reading the license agreement (use the vertical scroll bar or PAGE DOWN to see more of it), click on Yes if you agree to the terms. (If you do not agree, the software will not be installed.)
13. In the next window you can select the folder in which to install the JDK. Accept the default (C:\Jdk1.2) by clicking Next.
14. In the Select Components dialog box, shown here, select all the components and click Next. The JDK installation on your system will begin.



15. When the files are transferred, another Software License Agreement dialog box will be displayed. Click Yes after reading it.
16. The Choose Destination Location dialog box will be displayed. This sets the location of the Java Runtime Environment. Accept the default (C:\Program Files\JavaSoft\JRE\1.2) by clicking Next.
17. When the files are transferred, the Setup Complete dialog box will be displayed. Clear the Yes, I Want To View The Readme File check box and click Finish.
18. Open the Readme file in your browser by locating the file in Windows Explorer (C:\Jdk\1.2\Readme.html by default) and double-clicking it. The Readme file contains general information about the JDK and installation instructions. Review the file.
19. In the Windows Explorer, double-click on the documentation file (Jdk114doc.zip). If you have installed and associated an application for Zip-compressed files, the application will open.
20. Unzip the files to the Jdk1.2 folder. You must unzip the documentation files to the same folder structure as when they were originally compressed. In WinZip, you do this by selecting the Use Folder Names check box.
21. To save space on your hard drive, you can now delete the original compressed files in the temp folder you downloaded the JDK to. If you have the room, you may want to save the original compressed JDK and documentation files in case you need to reinstall them.

You now have the tools you need to compile Java applets from the source code files. In the next section you will use these tools to compile a sample applet included with the JDK.

Compiling Java Applets

While the Java Development Kit contains some precompiled applets as well as complete HTML pages that demonstrate them, knowing how to compile a Java applet from source code is a necessary skill for web designers who intend to use Java in their webs. Even if you never intend to write your own applets, you will probably still need to compile the applets you download from the Web, many of which are available only as Java source code. The process is relatively straightforward, but it does require using the MS-DOS interface included in Windows (commonly referred to as a "DOS window") and a basic knowledge of

using MS-DOS command syntax. The following steps will take you through the process:

1. Create a temporary folder on your hard drive for the source code file. Name the folder **Javawork**.
2. Copy the TicTacToe.java file (C:\Jdk1.2\Demo\Applets\TicTacToe\TicTacToe.java if you used the default folder structure) to the Javawork folder.
3. Open WordPad (Start menu, Programs, Accessories, WordPad), and then select Open from the File menu. In the Open dialog box, select All Documents (*.*) from the Files Of Type drop-down menu.

Note

Notepad is not suitable for opening or editing the Java source code files included with the JDK. These files contain characters that do not display properly in Notepad.

4. Open your Javawork folder from the Look In drop-down menu and then the file and folder list, and double-click TicTacToe.java. The TicTacToe.java source code file will be displayed as shown in Figure 19-13.

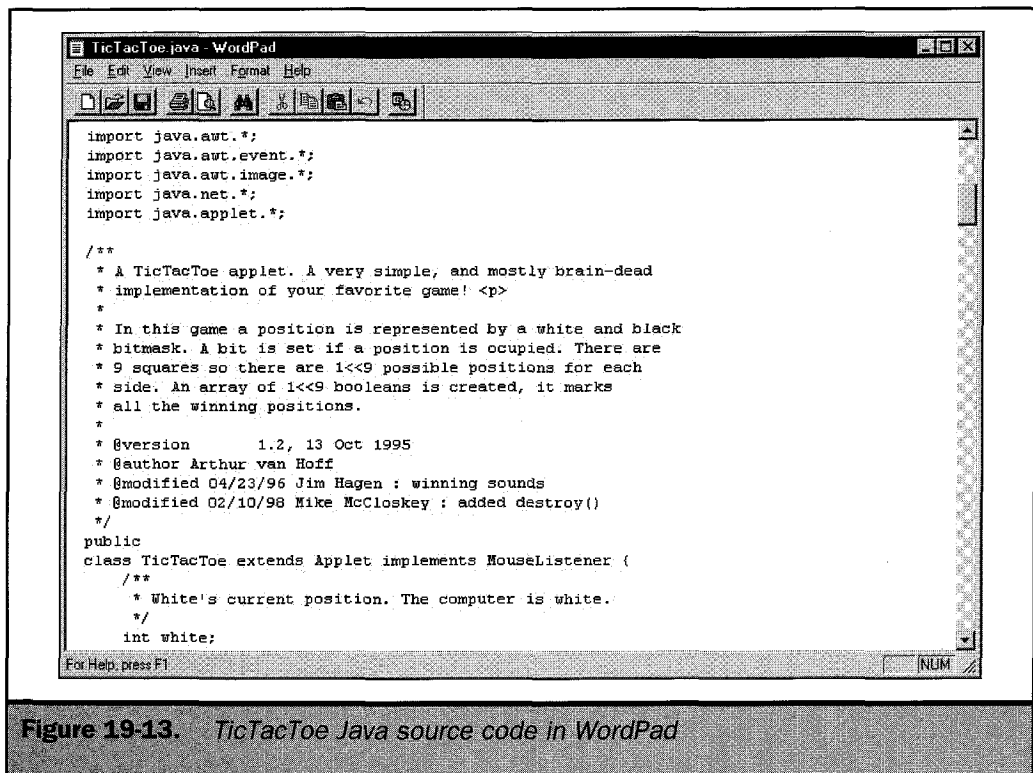


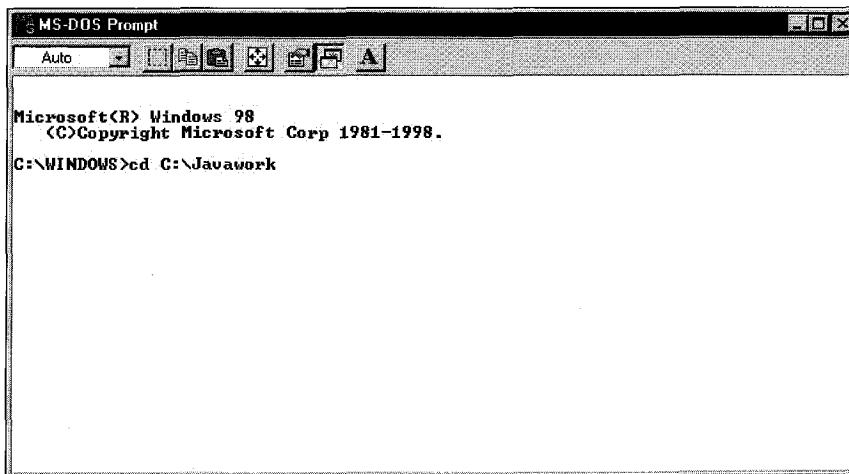
Figure 19-13. TicTacToe Java source code in WordPad

The source code includes a number of comments that describe the function of each Java statement in the file. Comments are all the text between the `/**` and `*/` markers. For example, in the lines

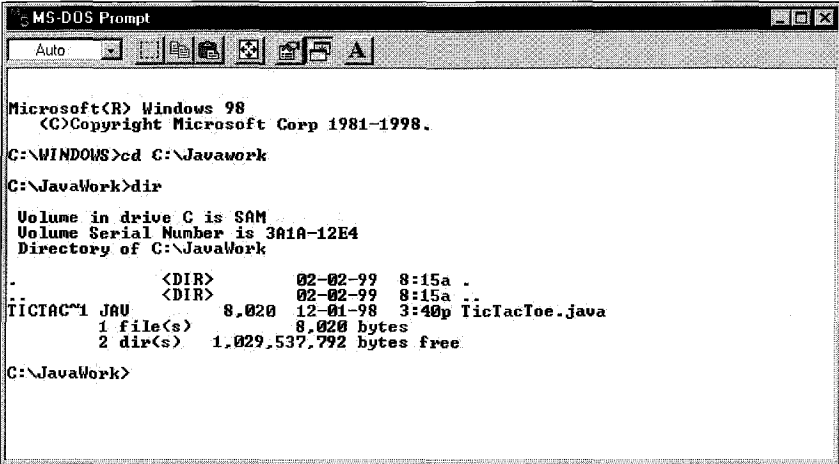
```
/**
 * White's current position. The computer is white.
 */
int white;
```

the comment (White's current position...) explains that the Java statement `int white;` is an integer that represents the current grid position of white's move. The variable name, `white`, contains the integer value. This variable is used to pass the current position of white to other functions in the applet. It is an axiom of programming that one of the best ways to learn a new programming language is to study other people's code. This is equally true of Java. If you are familiar with programming languages such as C++, the Java code will look familiar to you.

5. Close WordPad without changing the file.
6. Open a DOS window by opening the Start menu, pointing to Programs, and then clicking MS-DOS Prompt.
7. At the `C:\WINDOWS>` prompt in the DOS window, type `cd C:\Javawork` (or use the path where you created the Javawork directory), as shown here, and then press ENTER. CD is the DOS command for Change Directory. This will make the Javawork directory the current directory, and the prompt will read `"C:\Javawork>"`.



8. Type **dir** and then press ENTER. This is the command to display the contents of the current directory. Your DOS window should display a listing similar to this:



```

MS-DOS Prompt
Auto
Microsoft(R) Windows 98
(C)Copyright Microsoft Corp 1981-1998.
C:\WINDOWS>cd C:\Javawork
C:\JavaWork>dir
Volume in drive C is SAM
Volume Serial Number is 3A1A-12E4
Directory of C:\JavaWork

    <DIR>          02-02-99   8:15a  .
    <DIR>          02-02-99   8:15a  ..
TICTAC~1 JAV      8,020    12-01-98   3:40p  TicTacToe.java
1 file(s)          8,020 bytes
2 dir(s)        1,029,537,792 bytes free
C:\JavaWork>

```

Note

The directory listing displays two names for each file. On the left is the DOS name of the file in 8.3 format, in this case TICTAC~1 JAV, and on the right, the long filename supported by Windows, TicTacToe.java.

9. Compile the TicTacToe.java source file by typing **C:\Jdk1.2\bin\javac TicTacToe.java** at the C:\Javawork> prompt and pressing ENTER. The name of Sun's Java compiler is javac. The command you entered instructed it to convert the source file TicTacToe.java into the executable applet TicTacToe.class. Java names are case sensitive and must be entered exactly as shown.
10. When the C:\Javawork> prompt reappears in the DOS window, type **dir** and press ENTER. In the listing that is displayed you will see that there are now two files in the Javawork directory—the original source file and TicTacToe.class, the executable applet.

In the preceding steps you used the simplest form of the javac command to generate the applet. There are also a number of options you can use with javac to control the compiling or to generate messages. In the next step you will use the -verbose option to generate a list of all the steps the compiler is taking to generate the applet. This will overwrite the applet you just created.

11. At the C:\Javawork> prompt type **C:\Jdk1.2\bin\javac -verbose TicTacToe.java** and press ENTER. As each step is executed, the compile event and the time it takes to compile are displayed.

A complete listing of the options for the javac compiler can be found in the documentation web pages at *servername/jdk1.2/docs/tooldocs/win32/javac.html*. You can open this page and find information about the other tools included with the JDK. You open the documentation home page (*servername/jdk1.2/docs/index.html*) and click the Tool Docs hyperlink. This takes you to the JDK Tool Documentation index. Clicking Tool Documentation opens the JDK Tools page. Scroll down this page to the Basic Tools heading, shown in Figure 19-14, which contains a brief description of each tool and a hyperlink to the page where the tool is explained in detail.

12. Close the DOS window by typing **exit** at the C:\Javawork> prompt and pressing ENTER.

The final step in compiling a Java applet is to test it on a web page. Do that now with these instructions.

1. In FrontPage, open your Active web home page in Page view, if it isn't already open.
2. Create a new folder in your Active web with the name **TicTacToe**. Make sure the folder is selected.
3. Open the File menu and choose Import. In the Import dialog box, click Add File.

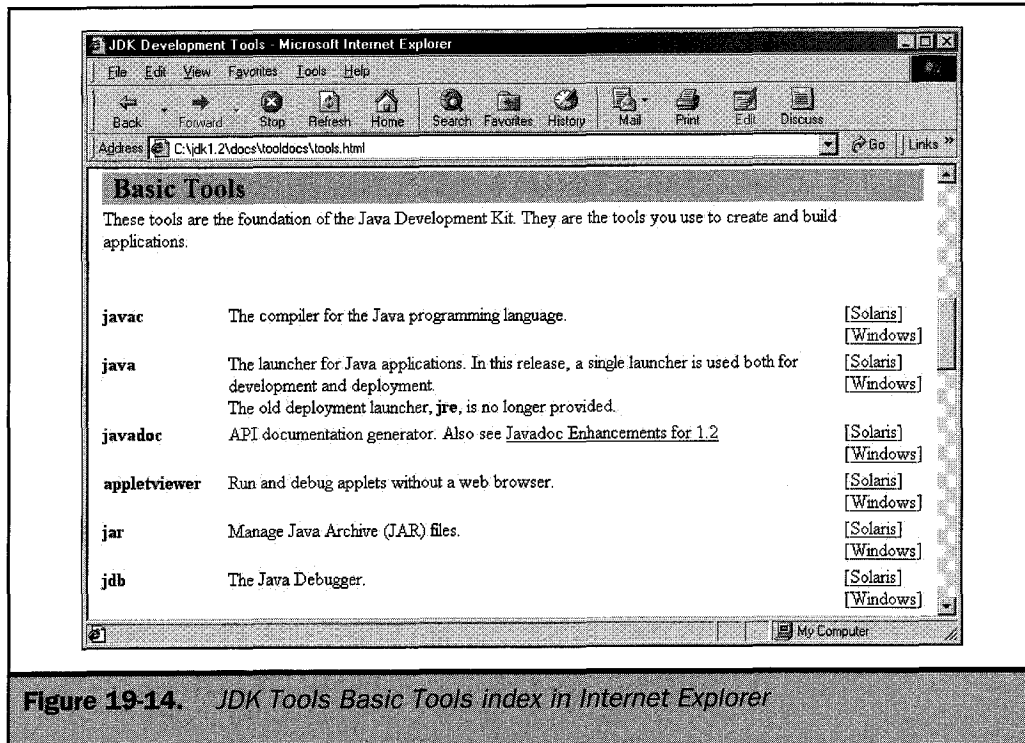


Figure 19-14. JDK Tools Basic Tools index in Internet Explorer

4. In the Add File To Import List dialog box, select your Javawork folder in the Look In drop-down list. Select TicTacToe.class and click Open; then click OK.
5. With the TicTacToe folder still selected, open the File menu and choose Import. In the Import dialog box, click Add Folder.
6. In the Browse For Folder dialog box, select the folder in the JDK that contains the sound files the applet uses. The default path is C:\Jdk1.2\Demo\Applets\TicTacToe\Audio. Click OK.
7. Repeat step 6 to import the folder containing the graphic files the applet uses. The default path is C:\Jdk1.2\Demo\Applets\TicTacToe\Images. Click OK twice.
8. In Page view, place the cursor on the line below your Clock applet.
9. Open the Insert menu and choose Advanced | Java Applet.
10. In the Java Applet Properties dialog box, type **TicTacToe.class** in the Applet Source text box and **TicTacToe** in the Applet Base URL text box.
11. Type **150** in both the Width and Height text boxes, and then click OK.
12. Save your work. In Page view, your Active web home page should look similar to Figure 19-15.

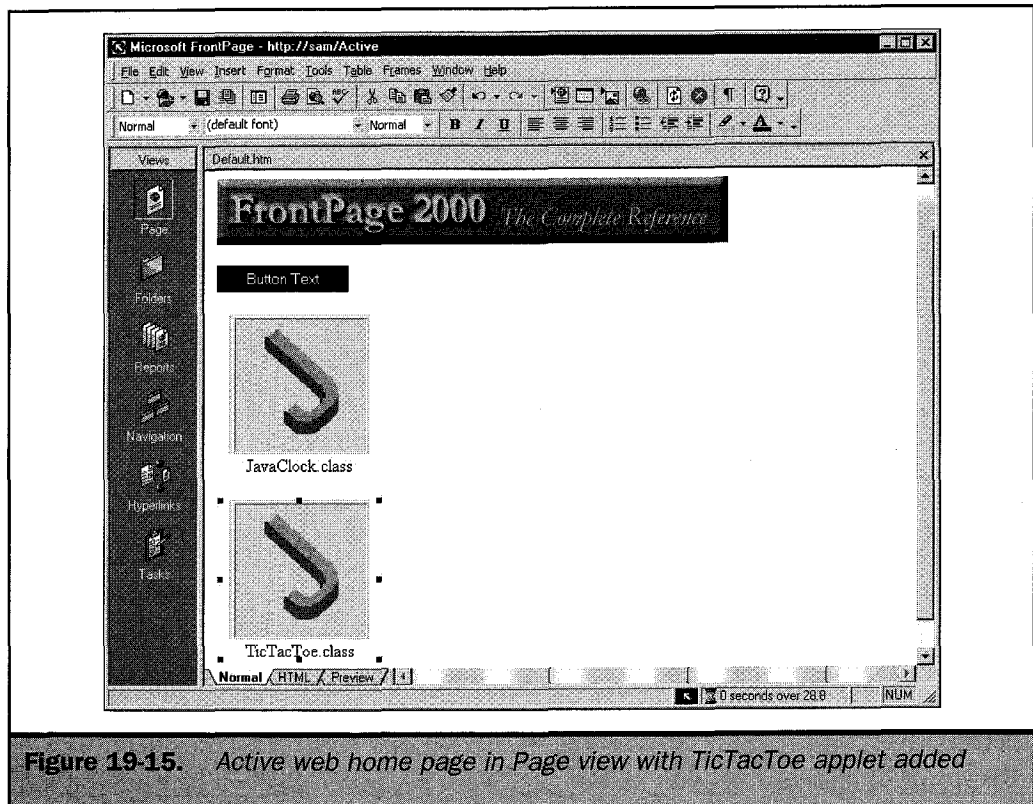
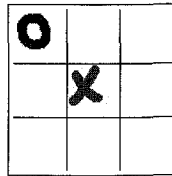


Figure 19-15. Active web home page in Page view with TicTacToe applet added

13. Open the page in your browser. Figure 19-16 shows the TicTacToe applet in Netscape Navigator.
14. Start the game by clicking in any empty cell. An "X" will be placed in that cell, and the applet will counter by placing an "O" in a cell that will block you, as shown here.



This has been only a brief trip through the world of Java. It is a powerful programming language that opens new doors for the web designer. Because Java is a true programming language, it also requires a comprehensive knowledge of the language and its capabilities to be used effectively. If you are a C++ programmer, the transition will be smooth. If Java is your first real programming language, you will

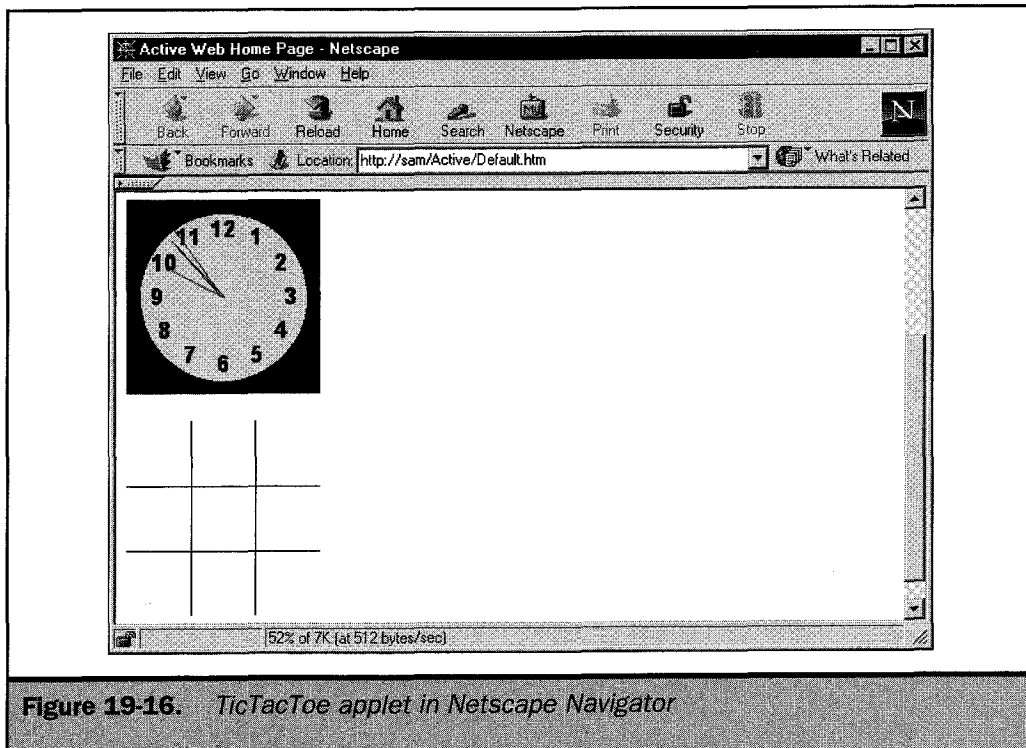


Figure 19-16. TicTacToe applet in Netscape Navigator

have to devote a significant amount of time and energy to learning it. It is a language that will play an ever-increasing role in web design, so your time may be well spent. If you don't want to learn a programming language, you can find a growing number of precompiled Java applets on the Web. The information in this section has presented the basics you need to download and use applets you find on the Web.

In the next section you will learn about ActiveX, Microsoft's technology that is designed to offer even more advantages to the web designer than Java.

ActiveX

Java is a new technology. In a sense, its designers began with a clean sheet of paper. (An interesting personal history of Java is included *The Java Handbook* [Osborne/McGraw-Hill, 1996] by Patrick Naughton, one of the original team at Sun that created the language.) ActiveX is more evolutionary, having its roots in Microsoft's OLE (object linking and embedding) technology.

Note

By this point you already have experience using ActiveX controls on the Web. All the ASP components you've used in the previous chapters are ActiveX controls.

OLE was developed as a method of sharing text and graphics, generally called objects, between applications on a computer. If an object was linked between documents, a pointer was created in the receiving document pointing to the object in the original document. When the original object was updated, the linked object was also updated. For example, you could link a spreadsheet to a word processing document. When the data in the spreadsheet changed, it would be reflected in the word processing document. When an object was linked, it actually existed only in the original document. The receiving document only contained a pointer to the original. When an object was embedded, an actual copy was placed in the receiving document.

This method, while it worked fine when both documents were on a single computer, had its shortcomings in a networked environment. This led Microsoft to develop new technologies, such as Component Object Model (COM), Distributed Component Object Model (DCOM), and OLE Control Extensions (OCX).

Essentially, ActiveX is object-oriented programming for the Web. In the section of this chapter on object-oriented programming, the concept of objects was introduced. An object was defined as a process that accepts information, processes it, and then outputs the result. ActiveX brings the same modular concept to a web page, with the addition that an object can also be a data file. This is a greatly simplified explanation of the technology, but it avoids turning this chapter into a programming handbook, rather than a guide for web designers who want to add the latest features to their FrontPage webs.

Tip

For those who do want a programming handbook for ActiveX, ActiveX from the Ground Up, by John Paul Mueller (Osborne/McGraw-Hill, 1997), is an excellent choice. It contains a thorough explanation of the evolution of ActiveX from OLE, as well as all you need to know to create your own ActiveX controls.

The ActiveX equivalent of a Java applet is an ActiveX control. Because ActiveX has evolved from OLE, ActiveX controls can be used with many different programming languages, including all the Microsoft programming and database languages. This means you can use the same control with your Access database as you do with your web page. This is also an area where ActiveX differs from Java.

ActiveX is a very useful technology, but it does have its drawbacks. In particular, it is a Windows-based technology. This leaves Mac and UNIX users out of the picture, at least for the present. If and when support is added for these platforms, the code will need to be compiled separately for each. This would lead to maintaining separate web pages for each operating system. Of course, Windows computers make up the majority of the market, and with corporate intranets, the operating system can be controlled. It also is less secure than Java, which is balanced by the fact it is potentially more powerful.

Note

Neither Java nor ActiveX is totally secure. Creative programmers can almost always find a way to get around security safeguards. When implementing any web technology, you must use caution and pay attention to what is happening on your web site.

These issues are not new to computing. At every stage of growth in the industry, the question of features versus compatibility has arisen. Often, as with ActiveX, compatibility has suffered in order to increase the usefulness of software. These are simply issues you need to consider before using ActiveX with your webs.

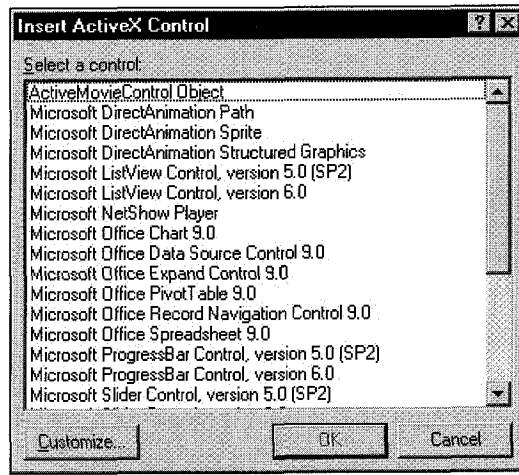
Since ActiveX controls are commonly written in Visual Basic, this section will not go into writing your own controls. Instead, you will use ActiveX controls included with FrontPage and readily available on the Web. Additional information about ActiveX and other Internet technologies can be found on the Microsoft Site Builder Network (<http://microsoft.com/sitebuilder/>).

The next section will take you right into using ActiveX with your FrontPage webs.

ActiveX and FrontPage

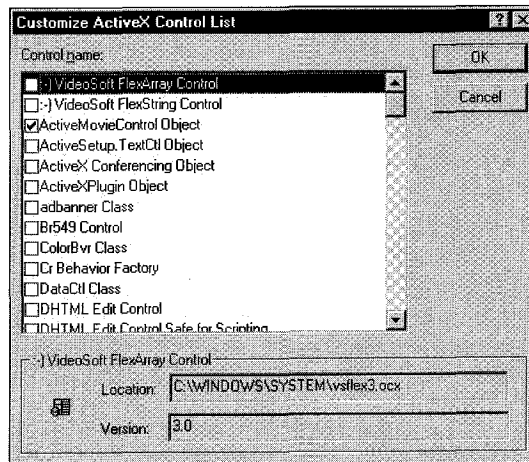
FrontPage comes with a number of ActiveX controls you can add to your webs, as you will see next. Your Active web home page should still be open in FrontPage.

1. In Page view, move the cursor to the line below your TicTacToe Java applet.
2. Open the Insert menu and choose Advanced | ActiveX Control. The Insert ActiveX Control dialog box, shown here, is opened.



This looks like quite a collection of ActiveX controls, but there are actually more.

3. Click Customize in the Insert ActiveX Control dialog box. The Customize ActiveX Control List dialog box shown next is opened.



4. Click Cancel to close the Customize ActiveX Control List dialog box. In the Insert ActiveX Control dialog box, select Microsoft Office Spreadsheet 9.0 and

click OK. A functioning spreadsheet is placed on your page, as shown in Figure 19-17.

5. Save your page and then open it in Internet Explorer.
6. In Internet Explorer, scroll down until the ActiveX spreadsheet is visible. In cell A1, type 2, and then press DOWN ARROW.
7. In cell A2, type 2 and press ENTER. Cell A3 should now be selected.
8. Click the AutoSum button on the spreadsheet's toolbar and then press ENTER. Cell A3 now contains the formula `=Sum(A1:A2)` and displays the total of the numbers in those cells, as shown in Figure 19-18.

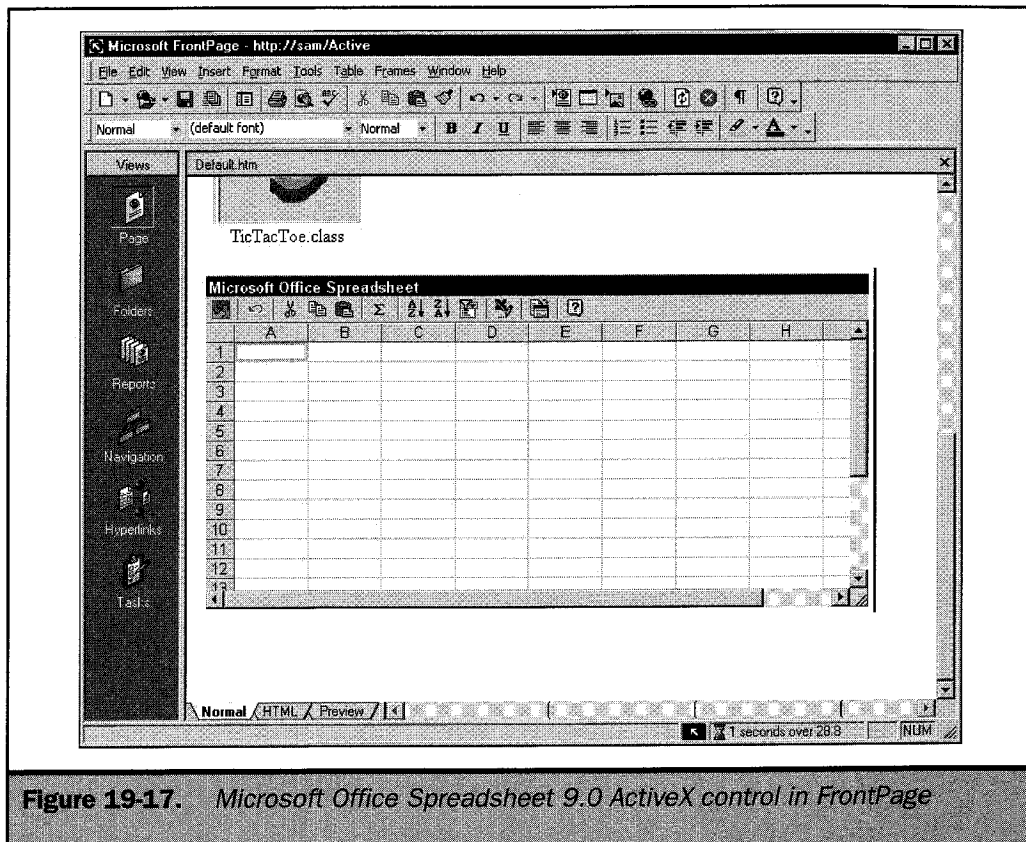


Figure 19-17. Microsoft Office Spreadsheet 9.0 ActiveX control in FrontPage

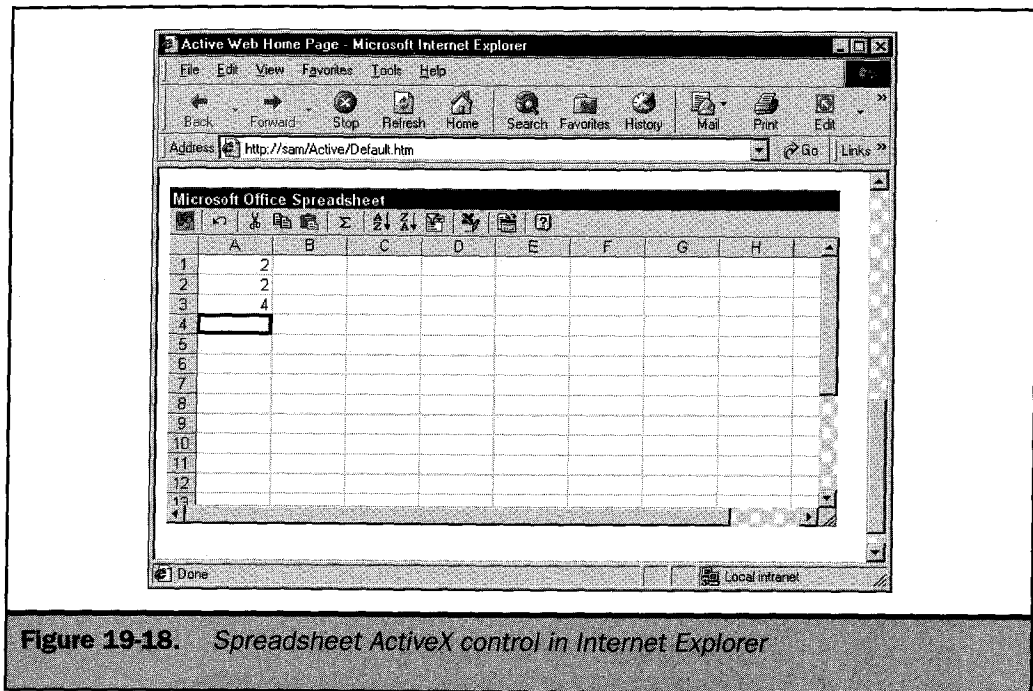
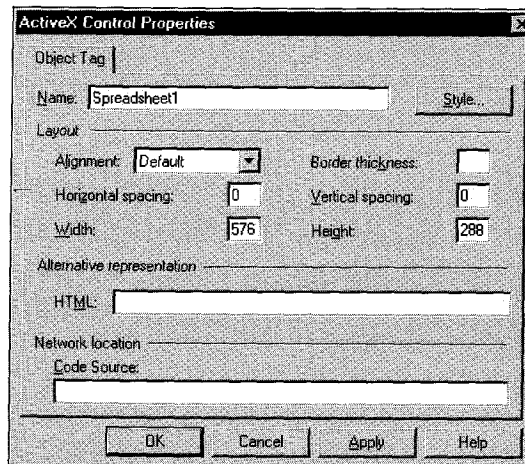


Figure 19-18. Spreadsheet ActiveX control in Internet Explorer

9. In FrontPage Page view, right-click the spreadsheet ActiveX control in an area outside of the cell matrix and select ActiveX Control Properties. This opens the ActiveX Control Properties dialog box, shown here.



The options available in the ActiveX Control Properties dialog box will depend on the specific ActiveX control, though the choices shown here will be available for most controls.

10. In the HTML Alternative Representation text box, type **<H1>This browser doesn't support ActiveX.</H1>** and click OK.
11. Save your page and then open it in Netscape Navigator, as shown in Figure 19-19.

The preceding example illustrates one of the problems with ActiveX controls—Netscape Navigator doesn't support them without a separate plug-in. Since Netscape still holds a large share of the browser market, this currently makes using ActiveX controls in your webs a risky business. However, this situation will change as support for ActiveX grows. On an intranet, if you can control the browsers used, this limitation doesn't apply.

The Microsoft Office Spreadsheet ActiveX control is one of a number of ActiveX controls that are included with FrontPage. All the ActiveX controls installed on your system (both those included with FrontPage and those you download and install separately) are displayed in the Customize ActiveX Control List dialog box. The controls that are selected in the Customize ActiveX Control List dialog box are also available in the Insert ActiveX Control dialog box.

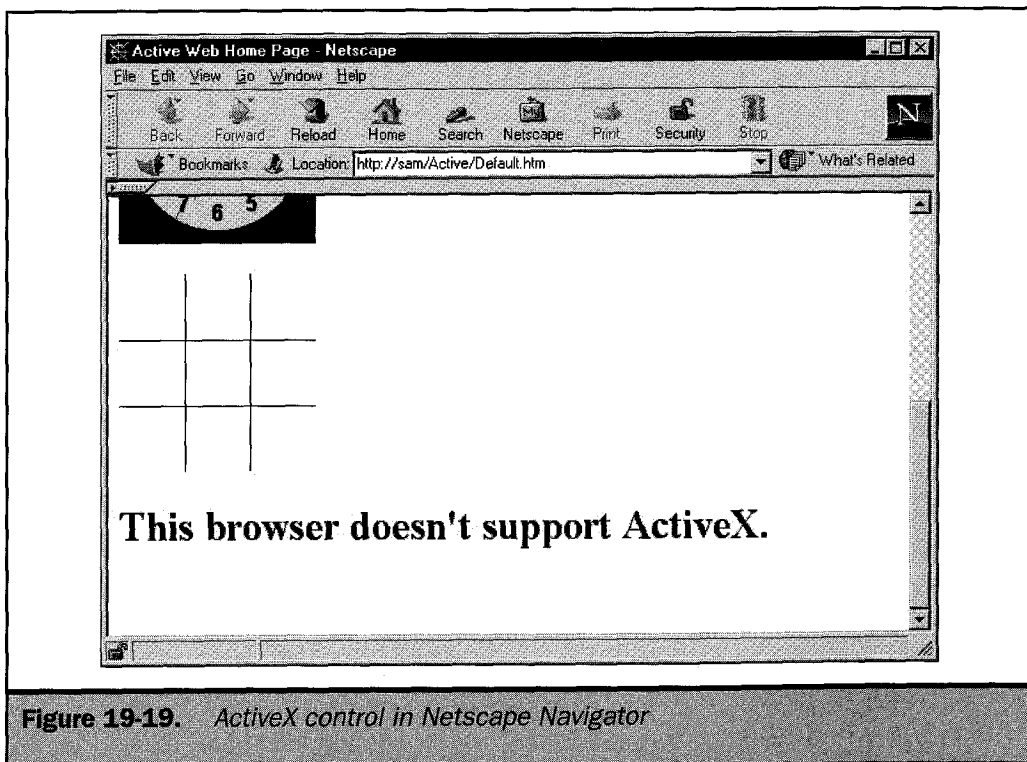


Figure 19-19. ActiveX control in Netscape Navigator

Each of the technologies covered in this chapter could easily fill a book of its own, and in fact each does. The purpose here was to give you an overview of them so that you could make an informed decision about which technologies you want to add to your own webs. Both Java applets and ActiveX controls require programming that is not for the faint of heart. If you already program in C++, the step up to these new languages will not be difficult. The Sun (<http://www.sun.com>), Netscape (<http://home.netscape.com>), and Microsoft (<http://www.microsoft.com>) web sites all have a variety of programming tools, documentation, and examples to get you started.

The coming months will see these technologies added to a growing number of web sites, as well as advances in the technologies themselves. If you want your webs to stand out in the crowd, these tools are part of your future.