Administrator's Guide

Netscape Enterprise Server

Version 3.0

Recycled and Recyclable Paper

Version 3.0

Printed in USA
98 97 96 10 9 8 7 6 5 4 3 2 1

Netscape Communications Corporation 501 East Middlefield Road, Mountain View, CA 94043

Contents

Introduction

Welcome to the Netscape Enterprise Server and the Internet. Netscape Communications Corporation is the premier provider of open software that lets people and companies exchange information and conduct commerce over enterprise networks and the Internet.

This *Administrator's Guide* documents the Netscape Enterprise Server.

What's in this book?

This manual explains how to install and configure the Netscape Enterprise Server. After configuring your server, use this manual to help maintain your server.

After you install the server, this book is available in HTML format in the server root at `manual\https\ag` in your server root directory.

Conventions used in this book

This section explains the conventions used in this book.

Italics — This typeface is used for book titles, emphasis, and any text that is a placeholder for text you need to replace for your system. For example, in a URL that contains a reference to your server's port number, the URL might contain *portnumber* in italics. Replace the words in italics with the actual value for your server.

`Monospaced font` — This typeface is used for any text that you should type. It's also used for functions, examples, URLs, filenames, and directory paths.

Sidebar text — Sidebar text marks important information. Make sure you read the information before continuing with a task.

| The vertical bar is used as a separator for user interface elements. For example, Server Status|Log Preferences means you should click the Server Status button in the Server Manager and click the Log Preferences link.

Computer system requirements

The Netscape Enterprise Server needs specific software and hardware. Before you can install a server, your computer must meet or exceed the following requirements.

- Windows NT workstation with a 486, Pentium, or a Digital Alpha chip.
- Windows NT version 3.51, running Service Pack 4, or NT version 4.0 with no Service Pack or running Service Pack 1. (Get Service Packs from Microsoft at `http://www.microsoft.com`.)
- Minimum 32MB RAM (more RAM is recommended for serving large numbers of clients or high levels of transactions).

  If you plan on running more than two separate instances of the web server on your system, each server will require an additional 16MB RAM. For example, if your system has three running web servers on it, your system would need 48MB RAM.
- Paging space at least as large as the amount of RAM (twice the amount of RAM is recommended).
- 100MB free disk space for the installation.
- 30MB free disk space for the log files (for approximately 300,000 accesses per day).
- Netscape Navigator or Navigator Gold 3.0 or higher, or Netscape Communicator. JavaScript must be enabled in the client software before you can administer your server.

Further reading

Refer to the following documents for more information on your server.

Managing Netscape Servers contains information on the administration server and global information on topics such as encryption, access control, and performance monitoring.

Web Publisher User's Guide contains information on using the web publishing system. This manual is included with your server in HTML format.

The Netscape DevEdge site contains documentation for developers, including:

- *Agent APIs*
- *JavaScript Reference*
- Netscape Internet Service Broker programmer's guides and reference guides for Java and C++
- *Notes for Java Programmers*
- *Web Publishing Client API Guide*
- *Writing Server-Side JavaScript Applications*
- *Writing Web Applications with WAI*

To access these documents, use the following URL:

```
http://developer.netscape.com/library/documentation/index.html
```

Click the Server link to see the documents pertaining to the web server.

In addition, you can find the *NSAPI Programmers' Guide*, a document that discusses the Application Programming Interface (API) used in Netscape Enterprise Server 2.0. This API is still compatible with the 3.0 server. In addition, this document also contains documentation of the `magnus.conf` and `obj.conf` files. To write plug-ins for the 3.0 server, see *Writing Web Applications with WAI*, which explains how to use the Web Application Interface (WAI).

Contacting Technical Support

For product-specific Technical Support assistance, please see the Product Support Page for the Netscape Enterprise Server at `http://help.netscape.com/products/server/enterprise/index.html`.

For general Technical Support assistance, please see the Netscape Technical Support Page at `http://help.netscape.com`.

Chapter

1

Welcome to the Internet

A *protocol* is a set of rules that computers use when exchanging information with each other. The basic protocol for the Internet is TCP/IP.

One computer by itself is a useful tool. You can work with and store a great deal of information. A *network*, or group of computers working together, is an even greater tool. You and other people can share and work with each other's information in a variety of ways. If a network can exchange information with other networks through a protocol called TCP/IP, it is part of the *Internet*. At its simplest, the Internet is a network of networks. No one person or company runs the Internet. Because of this design, the information available on the Internet changes constantly, and new computers are continually added to the network, which provides an ever-growing source of information.

Through the Internet people can:

- Send and receive email (messages and files)
- Read newsgroups ("bulletin boards" to discuss special-interest topics)
- Visit web sites (remote areas that users can get information from, or interact with)

Web sites are collections of documents and programs that can include hypertext, graphics, sounds, movies, and more. Often they contain links to other web documents, perhaps at other web sites. To a user, it hardly matters that some documents are at different sites—they are all merely a click away. All the web sites in the world are known collectively as the World Wide Web

(often called "WWW" or simply "the web"). You access the web with a *web client* like the one that's part of Netscape Navigator and Netscape Communicator.

A company can maintain an internal network of web sites, email, and other Internet services; this structure is called an *intranet*. An intranet is a particularly useful way to disseminate information within a company. Intranets can contain such information as benefits policies, sales and marketing product details, company phone lists, product plans, and online forms.

Usually, an intranet is kept separate from the Internet by hardware and software called a *firewall*, so that only people in the company can access that intranet.

When a company extends its intranet outside the firewall to allow customers, suppliers, and offsite workers access to the company's data, that structure is called an *extranet*.

Web basics

The web is based on a client/server relationship. The client program (a web browser) requests information, and the server program supplies it. When clients on the Internet connect to servers to send email, read newsgroups, or visit web sites, they communicate using other protocols in addition to TCP/IP. Multiple protocols are often thought of as layers. For example, the bottom layer might be PPP (Point-to-Point Protocol) if you use a modem, or Ethernet if you are on a corporate intranet.

TCP/IP is the middle layer in Internet communication, and it specifies exactly how all digital data is transferred from one computer to another. The next layer above might be HTTP (HyperText Transfer Protocol), which specifies what type of communication and information is sent between a web site and a web browser. HTTP is what makes web servers and web browsers speak the same language; TCP/IP is what makes sure the messages are sent back and forth reliably.

Other protocols that work "on top" of TCP/IP include:

- SMTP (Simple Mail Transfer Protocol for email)
- NNTP (Network Transfer Protocol for newsgroups)
- FTP (File Transfer Protocol for transferring files)

The various protocols are handled transparently by the client and server software. The only time people see what protocol they're using is when they type an Internet address, called a URL, into the Location field of their browser.

Understanding URLs

To get information from the Internet, you need to know the address of the information. These addresses are called Uniform Resource Locators (URLs). URLs are in the format *protocol://computer/unique identifier*. For example, the unique identifier could be a directory followed by a file. An example of a URL is `http://www.danishfurniture.com/products/orderform.html`.

Most files on the Web have the extension `html` or `htm`. HTML stands for HyperText Markup Language.

Because the protocol for accessing the web is HTTP, all web URLs begin with `http://`, as in the preceding example. The rest of the example URL is straightforward—the computer hostname is `www.danishfurniture.com`, the directory on that computer is `products`, and the file in question is `orderform.html`. Typing that URL into Navigator's Location field displays the file `orderform.html`, from `www.danishfurniture.com`, in the `products` directory. The file is sent from the server to the client through HTTP.

Note On the Internet, a computer is identified by a number, such as 198.95.251.30. This number is called a computer's Internet Protocol address, or *IP address*. To make Internet navigation easier for humans, each IP address is also associated with a hostname, such as `home.netscape.com`, or `www.danishfurniture.com`. The words you type into the Location field in Netscape Navigator are automatically translated to IP addresses by a special program called the *Domain Name System* (or DNS), which is maintained by your Internet provider or information systems department.

Understanding web servers

In the previous example, a computer named `www.danishfurniture.com` holds a file in one of its directories. This in itself is not unusual—every computer contains files in directories. What makes `www.danishfurniture.com` special is that it can "serve" its documents to clients when requested. This computer runs a server program that "speaks" HTTP and is physically connected to a TCP/IP-based network.

Sometimes the word *server* is used to mean the machine that serves the documents, but in these discussions, *server* refers to the installed program.

When you install a Netscape server, you specify a directory to contain all the files you want to serve to clients. All directories and subdirectories below that main content directory are available to clients. As mentioned previously, the most common type of content at a web site is HTML files. Graphic images and sounds are also often available on web sites. Additionally, web sites can contain programs that perform special tasks. For example, a web site might have a program that looks up stock prices, based on information a user fills out in an HTML form. Further, some web sites can hold programs that actually run on client computers, when they connect to the server.

The uses and abilities of web sites are virtually limitless. As the definition of HTML grows, so does its ability to present more varied and dynamic information. And with Java, the programming language that was created to run programs over the Internet, you have the power to make a web site into almost whatever you want. On the web you can find shopping malls, interactive stories, encyclopedias, video games, banks, and much more.

The Netscape Enterprise Server

Here is a partial list of what the Netscape Enterprise Server offers:

- Web publishing—End users can organize and publish their documents from their desktops with a web publishing interface. They can also use text search and revision control to manage content.
- Agents—You can create predefined agents that run on the server and bring information back to you, such as email notification that a specific document has changed.
- LDAP (Lightweight Directory Access Protocol) support—You can store users and groups in a centralized directory.

- Encryption—You can establish encrypted and authenticated transactions between clients and the server through the Secure Sockets Layer (SSL) 3.0 protocol.
- Access control—You can protect confidential files or directories by implementing access control by username, password, domain name, or IP address.

Chapter

2

Installing the server

This chapter tells you how to install the Netscape Enterprise Server and begin configuring it for your needs.

The administration server

When you install the Netscape Enterprise Server, the administration server is installed on the same computer. The administration server is a web-based server containing forms you use to configure your Netscape server products, including the Netscape Enterprise Server. With the administration server, you can manage multiple servers from a single interface.

During the installation of your Netscape Enterprise Server, you select a port for the administration server. Like any other server, the administration server listens to that port and responds to requests sent to it.

For more information on the administration server, see *Managing Netscape Servers*. For information on accessing your Netscape Enterprise Server through the administration server, see Chapter 3, "Managing your server."

Before you begin installation

Before you install the Netscape Enterprise Server, you should do the following:

- Make sure DNS is running. If you don't have DNS, you need to use IP addresses.
- Create an alias for the server.
- Create a user account for the server, and a group for all Netscape SuiteSpot servers.
- Choose unique port numbers for the administration and the web servers.

If you need more information on accomplishing these tasks than is included in the following sections, consult your system manager.

Make sure DNS is running

When you install the Netscape Enterprise Server, you'll be asked for a hostname or an IP address (or multiple entries of the same) as input strings.

- A hostname is a name for a specific computer in the form `machine.subdomain.domain`. For example, `www.netscape.com` is the machine `www` in the subdomain `netscape` and domain `com`.
- An Internet Protocol (IP) address is a set of numbers, separated by dots, that specifies the actual location of a computer on the Internet (for example, 198.95.251.30).

As you prepare for installation, make sure DNS is running properly. Otherwise, the server can't resolve (translate) hostnames and can't connect to any remote hosts.

To use a hostname during installation, your server must have an entry in a DNS server, and you must have entered a hostname and a domain name in the Network configuration utility in the Control Panel. You can also use an IP address during installation.

If your system is using a dynamic IP address, (for example, if your IP address is assigned by the Dynamic Host Configuration Protocol (DHCP) or your Internet service provider (ISP)), users might have difficulty accessing your web server as your IP address changes.

For more information about your system's networking configuration, see the documentation for your operating system.

Create an alias for the server

If your server will run on one machine among many in a network, you or your system administrator should set up a DNS CNAME record or an alias (such as www) that points to the actual server machine. Later, you can change the actual hostname or IP address of the server machine without having to change all URLs that point to the server machine.

For example, you might call the server `my_server.my_company.com` and then use an alias like `www.my_company.com`. So the URLs to documents on your server would always use the www alias instead of *my_server*.

Create a user account for the server

If you don't know how to create a new user on your system, consult your system documentation.

You should create a user account for your server. It should have restricted access to your system resources and run under a nonprivileged system user account (one that has a limited set of system permissions to your system). When the server starts, it runs with this user account. Any server extension modules the server uses are created with this user account as the owner.

During installation, the server uses the `LocalSystem` account, not the user account you created. Once you start and run the server, you should use the user account. You can change the user account for the server after the installation process. You can configure that user account so that is has permissions to get files on another computer, so that your server can serve files that are mounted from another computer.

In addition, the user account you create for the server should belong to a group that contains the server users for all Netscape SuiteSpot servers. With a SuiteSpot group, multiple servers can have access to shared files.

Note It's strongly recommended that you use a dedicated account for the server.

Choose unique port numbers

You need two port numbers: one for the administration server and one for the web server.

The standard web server port number is 80 and the standard SSL-enabled web server port number is 443, but you can install the server to any port. If you use a port other than the default port (port 80), the URL used to gain access to your home page will change. For example, if your computer is called `www.mozilla.com` and you choose port 9753, your server's URL will be `http://www.mozilla.com:9753/`.You should choose a random number for the administration server to make it harder for anyone to breach your server. When you configure your server, you use the administration server's port number. For example, for server `mozilla.com`, the server's URL could be `http://www.mozilla.com:2634/`.

Make sure the port you choose isn't in use. Check the file `C:\WINNT\system32\drivers\etc\services` on the server machine to make sure you don't assign a port number that is reserved for another service. If you choose a port that is being used by another service, the installation program prompts you for another port.

Migrating existing server settings

Before you migrate a 2.x server to a 3.0 server you should read the information on upgrading/migrating in *Managing Netscape Servers*.

If you currently have a 2.0 or 2.01 web server, the installation program gives you the option of migrating your server settings from your 2.x server to the 3.0 server. If you choose not to migrate settings when installing, you can do so later using the administration server. For more information on migrating using the administration server, see "Migrating a server from a previous version" on page 45.

You should stop running the 2.x web server before migrating settings. Make sure you have Netscape Navigator 3.0 or higher installed on your computer before upgrading.

When you install a 3.0 web server, the installation program checks if you have a 2.x web server installed. If you do, it offers you the option of migrating your server settings to your new 3.0 server. After you choose a destination location for your server, a Server Upgrade form appears. To migrate your settings, choose the Migrate existing server settings radio button.

If you choose to migrate, you go to the administration server migration forms. For more information, see "Migrating a server from a previous version" on page 45.

Copy any documents that you still want to use from your 2.0 document directory to your new 3.0 document directory.

Installation instructions

The following sections describe the installation of the Netscape Enterprise Server.

Warning You must install your version 3.0 servers in a separate server root directory from the server root that contains your 2.x servers. If you have beta versions of the 3.0 servers installed, you should uninstall them before installing the final version. When you uninstall, the document root is preserved.

Before installing the Netscape Enterprise Server, install Netscape Navigator 3.0 (or later) or Netscape Navigator Gold. You need one of these applications installed to configure your server. Both applications are available for downloading from `http://home.netscape.com`; Navigator Gold is included with Netscape Enterprise Server.

Note If you want to use the Netscape Enterprise Server's SNMP monitoring capabilities, make sure that SNMP is properly set up on your system before installing the server.

Any errors that occur when the server starts are logged in the Event Viewer. Once started, the server logs errors to the normal error log file.

Follow these steps to install the server:

1. Put the CD-ROM in the drive.
2. If you are using Windows NT 3.51, choose File|Run from the Program Manager. In the dialog box that appears, type `D:\NTX86\ENTPRISE\SETUP.EXE`, where *D:* is the letter of your CD-ROM drive. Press Enter. The Netscape Enterprise Server Installation dialog box appears.
3. If you are using Windows NT 4.0, click on the icon representing your CD-ROM drive. Doubleclick on the SETUP.EXE file. The Netscape Enterprise Server Installation dialog box appears.
4. Click Yes to continue. The files are extracted, and the Welcome screen appears.
5. After reading the Welcome screen, click Next.
6. The Software License Agreement appears. Click Accept to accept the license.
7. The Choose Destination Location screen appears. The default location for the server files is `C:\Netscape\SuiteSpot`, where *C:* is the letter of the drive on which you are installing the server.

   If you have no other 3.0 Netscape servers, you can use the Browse button to select a directory. However, if you already have other Netscape 3.0 servers installed, you need to install all servers into the same directory. In that case, the Browse button will be disabled.

   Click Next.
8. If you have a 2.x Netscape Enterprise Server installed, the Server Upgrade screen appears. You can choose either to migrate your existing 2.x server settings to your new 3.0 server, or to create new settings for the server you are installing. Click Next.
9. The Administration Server LDAP Configuration screen appears. If you want to use LDAP, click the checkbox and specify the server name, server port, and distinguished name (DN). If you do not want to use LDAP, leave the checkbox unchecked. Click Next.

10. The Administration Server Authentication screen appears. Type the user name for administration server access; the default is `admin`. Type the administration server access password; type it again for verification. Click Next.

    If you are using LDAP-based authentication, you must make sure that this user has access permissions to the LDAP server to perform user/group management tasks.

11. The Administration Server Port Selection screen appears. Type the port the administration server runs on. This can be any number from 1 to 65535. The URL for administration access is displayed. You might want to make a note of this URL. The default port is a randomly generated number from 1024 to 65535. Click Next.

12. If you are not migrating existing settings, the Enterprise Server Configuration screen appears. Type the path for your primary document directory, where your server's content files will be stored. The default document directory is `C:\Netscape\SuiteSpot\docs`. Click Browse to navigate your file system.

Note If you enter an existing document root, the sample HTML files won't be copied to the existing document root.

    Type a web server port number. If you use a port other than the default port (port 80), the URL used to gain access to your home page will change. For example, if your computer is called `www.mozilla.com` and you choose port 9753, your server's URL will be `http://www.mozilla.com:9753/`.

    Click Next.

13. The Server Configuration Summary screen appears. This screen contains information about the settings for your Netscape Enterprise Server and administration server. This screen gives you the opportunity to review your settings before the installation is complete. If they are correct, click Next.

14. The server files are installed.

15. The Netscape Enterprise Server 3.0 screen appears. Choose whether you want to read the Readme file and whether to connect to the home page (creating new server settings) or the upgrade page (migrating existing server settings). Click Finish. If you have requested it, the server's home page or upgrade page appears.

If you migrated existing server settings, and you chose to go to the upgrade page, the Choose Servers to Import page appears in a browser window. You have to enter your administration user name and password to access it. For information on migrating settings, see "Migrating a server from a previous version" on page 45.

Your web server can now be configured. For more information, see Chapter 3, "Managing your server."

What the installation process does

After you complete the installation forms, the actual configuration takes place. Some temporary files are written to *C*:\TEMP and removed after installation. In addition, several .dll files are copied to the *C*:WINNT\system32 directory, and several registry key entries are created.

The installation process places all the files under the server root directory that you specified during installation.

The following directories are created under the server root directory.

- admin-serv contains administration server directories:
  - config contains the administration server's configuration files.
  - logs contains the administration server's log files.
- adminacl contains the files that store access control configuration information for the administration server.
- authdb contains user databases in the 2.x format, if you have any. Databases for the current release are in the userdb directory.
- alias contains the key and certificate files for all Netscape servers.

- `bin` contains the binary files for the server, such as the actual server, the administration forms, and so on.
- `docs` is the server's default primary document directory, where your server's content files are usually kept. If you are migrating settings from an existing server, this directory doesn't appear until you finish the migration process.
- `extras` contains a log analysis tool.
  - The `flexanlg` directory contains a command-line log analyzer. This log analyzer analyzes files in flexlog format.
  - The `log_anly` directory contains the log analysis tool that runs through the Server Manager. This log analyzer analyzes files in common log format only.
- `httpacl` contains the files that store access control configuration information in the `generated.`*`server-identifier`*`.acl` and `genwork.`*`server-identifier`*`.acl` files. The file `generated.`*`server-identifier`*`.acl` contains changes you make using the Server Manager access control forms after saving your changes; `genwork.`*`server-identifier`*`.acl` contains your changes *before* you save your changes. In addition, this directory contains the `agents.acl` file, which contains the access control configuration information for agents.
- `https-`*`identifier`* are the directories for each server you have installed on the machine. Each server directory has the following subdirectories and files:
  - `agents-db` contains files used by agents.
  - `catalog` contains files used by the auto-catalog feature.
  - `config` contains the server's configuration files: `agent.conf`, `csid.conf`, `filter.conf`, `magnus.conf`, `magnus.conf.clfilter`, `mime.types`, `obj.conf`, `obj.conf.clfilter`, `process.conf rdm.conf`, `robot.conf`, `webpub.conf`, and `webpub.conf.clfilter`. Working copies are kept here. For complete information on `magnus.conf` and

`obj.conf`, see the *NSAPI Programmer's Guide* on the DevEdge site at `http://developer.netscape.com/library/documentation/index.html`.

- `conf_bk` contains backup copies of the server's configuration files.
- `db` contains the RDM (Resource Description Messaging) schema used by the auto catalog feature.
- `logs` contains any agent, error, and access log files.

- `include` contains header files.
- `install` contains files needed for migrating server settings and default configuration files needed for backward compatibility.
- `js` contains the Application Manager and the samples for server-side JavaScript.
- `lib` contains shared libraries.
- `manual` contains the online manuals for the product.
- `ns-icons` contains icons for FTP listings and Gopher menus used in "fancy" indexing lists. For more information about fancy indexing lists, see "Configuring document preferences" on page 53.
- `nsapi` contains header files and example code for creating your own functions using NSAPI. For more information, see Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/index.html`.
- `plugins` contains directories for Java, agents, search, and other plugins.
- `userdb` contains user databases and related information.
- `wai` contains information and sample code for using the Web Application Interface (WAI).

Changing the user account

After installing the administration and web servers, you might want to change the user account that the administration server uses. By default, the user account is `LocalSystem`.

To change the administration server user account after installation:

1. Create a user with the Windows NT Users Manager. The user must have "Log in as a service" rights.
2. Stop the server.
3. From the Windows Control Panel, choose Services.
4. Select the Administration Server 3.0 service.
5. In the Service pop-up, in the Log on As section, click the This Account radio button.
6. Type the user account you want the administration server to use.
7. Type the password for that account; type it again for confirmation.
8. Click OK.
9. Restart the server using the Services program or the Server Administration page.

Troubleshooting installation

This section describes the most common installation problems and explains how to solve them.

I accidentally denied all access to the Server Manager forms.
Log in to the system with the server's user account. In the `admin-serv\config` directory, edit the `ns-admin.conf` file. There's a line for allowed hosts or allowed addresses. Use wildcard patterns, or modify the lines to include your host and address, save the file, and then restart the administration server.

Clients can't locate the server.
First, try using the hostname. If that doesn't work, use the fully qualified name (such as www.domain.com), and make sure the server is listed in DNS. If that doesn't work, use the IP address.

The port is in use.
Most likely, you didn't shut down a server before you migrated the settings. Shut down the old server, then manually start the new one.

The port might also be used by another installed server. Make sure the port you've chosen isn't already being used by another server.

The installation could not proceed.
Error: Can't open file for writing.
Can't write to file *filename*. The error was error code 32.
Make sure you don't already have a web server running on your system.

This error message might also appear if you partially completed the installation so that the executable was running and you're trying to install the server again. In the Control Panel Services dialog box, shut down all the web server services and reinstall the server.

I've forgotten the administration user name and password.
In the `admin-serv\config` directory in your server root directory, edit the `admpw` text file, which contains a line of text similar to the following:

```
admin:lnOVeixulqkmU
```

The text before the colon is the administration user name (in this case, `admin`); the text following the colon is the password, which is encrypted.

Delete everything after the colon and save the file. Shut down the administration server, and restart it. When prompted for the administration password, leave the password field blank. You should be able to access the administration server now. Be sure to create a new password for the administration server. For more information on creating a new password for the administration server, see *Managing Netscape Servers*.

Using uninstall

You can remove the Netscape Enterprise Server from your computer by using the Uninstall application. The Uninstall application determines which Netscape servers are installed on your machine and allows you to choose which of those servers you want to delete.

To remove your Netscape Enterprise Server:

1. If you are running a Windows NT 3.51 system, double-click the Uninstall Enterprise icon in the Netscape Servers program group.

   If you are running a Windows NT 4.0 system, from the Start menu, choose Control Panel. Double-click on the Add/Remove Programs icon. Highlight Netscape Enterprise Server 3.0 from the list and click the Add/Remove button.

2. The Netscape Server Uninstall form appears. This form lists all of the Netscape SuiteSpot servers installed in the same location as the Enterprise server.

   Choose which server you would like to remove by checking the box next to its product name. You may choose to remove more than one server.

3. Click the Uninstall button. If the server you are removing shares files with other applications on your system, the File Delete Confirmation form will appear on your screen. On this form, you can determine which shared files the Uninstaller removes.

Note The uninstall program does not remove every server file. For example, it does not remove keys and certificates or the `docs` directory.

Chapter

3

Managing your server

This chapter describes how to configure and manage your server using the Netscape Server Administration page and the Server Manager.

During installation, you specified a port number for the administration server. The administration server helps you manage your Netscape Enterprise Server (or multiple servers) from a single interface—the Netscape Server Administration page. From this page you can access the Server Manager, which is a collection of forms you use to change options and control your web server.

This chapter contains sections on using the Server Administration page and Server Manager.

Using the Server Administration page

You configure your administration server and access the configuration forms for other Netscape servers (including the Netscape Enterprise Server) with the Netscape Server Administration page. This page contains links to the Server Managers for the Netscape servers you have installed.

You can perform the following web server tasks from the Server Administration page:

- Choose a server to configure.
- Install another web server on the machine.
- Remove a server from the list of servers you can configure.
- Start and stop a web server.
- Import or migrate from a 2.x web server to a 3.0 web server.

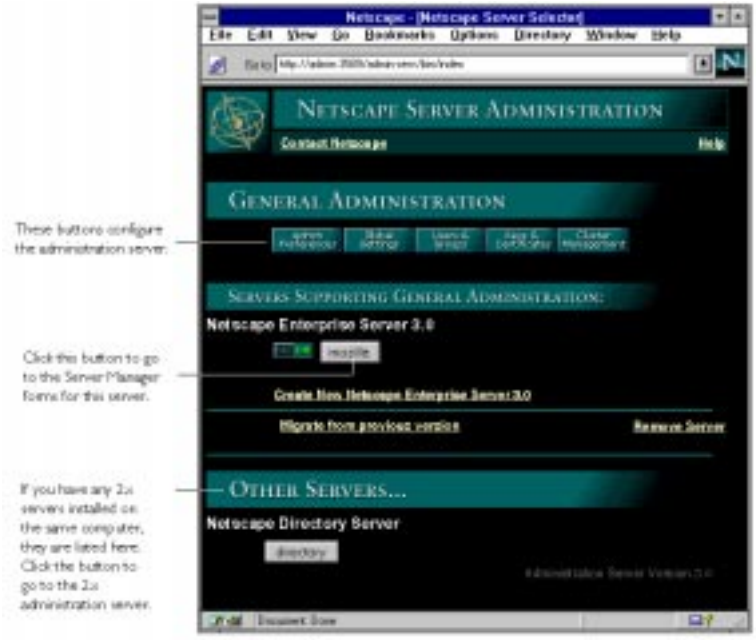
In addition, you can perform tasks for the administration server. For more information see *Managing Netscape Servers*.

Accessing the Server Administration page

In Windows NT 3.51, the installation program creates a program group with several icons. Use the Administration icon in the program group to connect to the administration server. In NT 4.0, click the Start button in the taskbar and select Programs|Netscape SuiteSpot|Administration to connect to the administration server.

Type the administration username and password you specified during installation. The Server Administration page appears, as seen in Figure 3.1. To configure a server using the Server Manager, click the name of the server you want to configure.

Figure 3.1 The Server Administration page



The administration server runs as a service. You can use the Control Panel to start this service directly. Once the administration service is running, you can use any browser that has access to the administration server to configure your servers.

Remotely accessing the Server Administration page

As long as you have access to client software such as Navigator, you can access the Server Administration page to configure your web server.

To remotely access the Server Administration page:

1. Using a browser that supports frames and JavaScript, such as Netscape Navigator, type the URL for the administration server:

   `http://servername.your_domain.domain:port_number/`

   Use the port number for the administration server that you specified during installation; this is not the port number for the web server.

2. You'll be prompted for a username and password. Type the administration server username and password you specified during the installation.

   The Server Administration page appears.

The Server Administration page lists all the servers you've installed on this system. Click the name of the server you want to configure.

Starting and stopping a web server

You can start and stop the servers listed in the Server Administration page by clicking the On/Off icon located to the left of the server's name.

Setting up multiple servers

There are three ways you can have multiple web servers on your system:

- Install multiple instances of the server
- Use hardware virtual servers
- Use software virtual servers

Each approach has its strengths and weaknesses; you should choose the one that's right for your situation.

If you install multiple instances of the server, you can have separate configuration information. For example, one instance of the server could have security features or web publishing enabled while another server could have them disabled. However, each instance of the server takes substantial resources of RAM, disk space, and swap space.

Hardware virtual servers allow you to map multiple IP addresses to multiple document roots. For example, if you have two IP addresses, you could map the first IP address to one document root and the second IP address to a second document root. While hardware virtual servers take fewer system resources than multiple instances of the server, they must share the same configuration information. For example, if one hardware virtual server has enabled security features or web publishing, they all must have it enabled.

Software virtual servers give you the ability to map a single IP address to multiple server names. Each software virtual server can have its own home page. One use for this is to host multiple web sites from one IP address. However, in order for software virtual servers to work correctly, the users accessing the server must be using client software that supports the HTTP Host header. Like hardware virtual servers, software virtual servers all must have the same configuration.

For more information on virtual servers, see "Setting up hardware virtual servers" on page 58 and "Setting up software virtual servers" on page 59.

Installing multiple instances of the web server

You can install another instance of the web server on your current computer without going through the installation program. Your web server software license allows you to have as many web server instances as you want on one system. Each web server you have installed can run on any TCP/IP port on your system, but you cannot run two web servers on the same port at the same time unless they are configured to respond to different IP addresses. Contact your system's vendor for information on how to configure your system to respond to different IP addresses.

If your system is configured to listen to multiple IP addresses, for each server you install enter one of the IP addresses that your system is hosting.

If you installed your server before configuring your system to host multiple IP addresses, configure your system to respond to different IP addresses. Then you can either install hardware virtual servers or change the server's bind address using the Server Manager (see "Configuring network settings" on page 79) and install separate instances of the server for each IP address.

To install another web server with its own separate configuration files:

1. Click Create New Netscape Enterprise Server from the Server Administration page.

2. Enter values for the following:

   - Server name—Type the fully qualified domain name for your server. If you are installing a second server for a custom domain, enter the domain here.
   - Bind address—If you're installing another server in order to have your machine answer to multiple IP addresses, enter the IP address that this instance of the server should listen to. Your system should already be configured to listen to multiple IP addresses. If you're not going to use multiple IP addresses, you can leave this field blank.
   - Server port—Type the number of the port that you want this server to listen to.
   - Server identifier—Type the server identification that the administration server will use for your web server (for example, `marketing_server`). This name is used to identify the server's subdirectory in the server root.
   - MTA host—Type the Message Transfer Agent (MTA) host. You must enter a valid MTA host if you want to use the agent email function.
   - NNTP host—Type the NNTP Network News Transfer Protocol (NNTP) host. You must enter a valid NNTP host if you want to use agents with the capability to post to news.
   - Document root—Type this server's document root, which is the directory that contains most of your server documents.

3. Click OK.

After installing, you will have several subdirectories in your server root directory, one for each server you installed. Each of these servers can be managed from the Server Administration page.

Migrating a server from a previous version

You can migrate a server from 2.x to 3.0 using the administration server. Your 2.x server is preserved, and a new 3.0 server using the same settings is created.

You should stop running the 2.x server before migrating settings. Make sure you have Netscape Navigator 3.0 or later installed on your computer before migrating settings.

1. From the Netscape Server Administration page, click the "Migrate from previous version" link. The Choose 2.x Server Root to Import From page appears.
2. Enter the directory that contains the server from which you want to import settings. Click the Find Servers button. The Choose Servers to Import page appears.
3. Click the checkbox next to the server with the settings you want to import. The next page displayed depends upon the server or servers you chose to import settings from. You may need to click an Import button to continue the import.
4. When the import of the server is complete, click the Dismiss button.

Removing a server from your system

To remove a server from your system, use the Server Administration page. Be sure that you will not need the server anymore. You can also remove the administration server for the type of server you are deleting. For example, if you are removing an Enterprise Server and no longer plan to install any Enterprise Servers, you could remove the administration server too.

To remove a server:

1. Shut down the server before removing it by clicking the On/Off icon to the left of the server name in the Server Administration page.
2. Click Remove a Server from the Server Administration page.

3. Select the server that you want to remove.

4. Select whether you want to remove the administration binaries, which include the administration server's configuration files and binaries.

Caution! Do not remove the binaries for the administration server if more than one server is installed.

5. Verify that you want to remove the server and the administration binaries by clicking the Yes checkboxes for each one you want to remove.

6. Click OK.

Using the Server Manager

The Server Manager is the collection of forms you use to change options and control your server. From the Server Administration page, which lists all the servers installed on your system according to identifier, access the Server Manager by clicking the button showing the server name (located next to the On/Off icon). The Server Manager appears. (You can use the Server Manager from any remote system as long as the system you're working on is one of the hosts that can access the administration server; you don't need to be working at the system the server is installed on.)

Note When changing server information, you must save and apply your changes in order for your changes to take place. After you submit a form, you get a pointer to a script that allows you to save and apply your changes.

You can return to the Server Administration page by clicking the Admin button in the upper-right corner of the Server Manager. The Server Manager is shown in Figure 3.2.

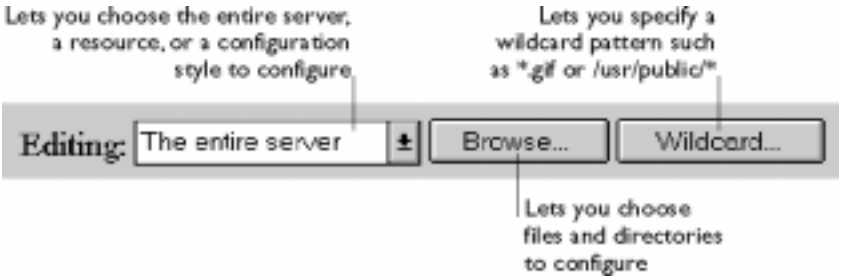
Figure 3.2 The Server Manager

Use the server configuration buttons in the top frame to configure the server. After clicking a button, you'll see a list of items on the left. Click one of these links; the corresponding form comes up in the main frame. If you need more information about a form, click Help for context-sensitive help. In the help window, click Contents to see a list of online manuals you can view.

Using the Resource Picker

Most of the Server Manager forms configure the entire server. Some forms can configure either the entire server or files or directories that the server maintains. These forms have the Resource Picker, shown in Figure 3.3, at the top. The Resource Picker lets you specify what resource to configure.

Figure 3.3 Resource Picker



Pick a resource from the drop-down list for configuration. Click Browse to browse your primary document directory; clicking Options allows you to choose other directories. Click Wildcard to configure files with a specific extension.

Wildcards used in the Resource Picker

In many parts of the server configuration, you specify wildcard patterns to represent one or more items to configure. Please note that the wildcards for access control and text search may be different from those discussed in this section.

Wildcard patterns use special characters. If you want to use one of these characters without the special meaning, precede it with a backslash (\) character.

Table 3.1 Resource Picker wildcard patterns

Pattern	Use
*	Match zero or more characters.
?	Match exactly one occurrence of any character.
|	An *or* expression. The substrings used with this operator can contain other special characters such as * or $. The substrings must be enclosed in parentheses, for example, (a|b|c), but the parentheses cannot be nested.
$	Match the end of the string. This is useful in *or* expressions.
[abc]	Match one occurrence of the characters a, b, or c. Within these expressions, the only character that needs to be treated as a special character is]; all others are not special.

Table 3.1 Resource Picker wildcard patterns

Pattern	Use
`[a-z]`	Match one occurrence of a character between `a` and `z`.
`[^az]`	Match any character except `a` or `z`.
`*~`	This expression, followed by another expression, removes any pattern matching the second expression.

Table 3.2 Resource Picker wildcard examples

Pattern	Result
`*.netscape.com`	Matches any string ending with the characters `.netscape.com`.
`(quark|energy).netscape.com`	Matches either `quark.netscape.com` or `energy.netscape.com`.
`198.93.9[23].???`	Matches a numeric string starting with either `198.93.92` or `198.93.93` and ending with any 3 characters.
`*.*`	Matches any string with a period in it.
`*~netscape-*`	Matches any string except those starting with `netscape-`.
`*.netscape.com~quark.netscape.com`	Matches any host from domain `netscape.com` except for a single host `quark.netscape.com`.
`*.netscape.com~(quark|energy|neutrino).netscape.com`	Matches any host from domain `netscape.com` except for hosts `quark.netscape.com`, `energy.netscape.com`, and `neutrino.netscape.com`.
`*.com~*.netscape.com`	Matches any host from domain `com` except for hosts from subdomain `netscape.com`.

Chapter 4

Managing server content

You can use the Server Manager to help manage your server's content. You create HTML pages and other files such as graphics, text, sound, or video, and then you store those files on your server. When clients connect to your server, they can view your files provided they have access to them. This chapter describes how you can configure and manage your server's content.

Setting the primary document directory

You probably don't want to make all the files on your file system available to remote clients. An easy way to restrict access is to keep all of your server's documents in a central location—known as the document root or primary document directory.

Another benefit of the document directory is that you can move your documents to a new directory (perhaps on a different disk) without changing any of your URLs because the paths specified in the URLs are relative to the primary document directory.

For example, if your document directory is `C:\Netscape\SuiteSpot\docs`, a request such as `http://www.mozilla.com/products/info.html` tells the server to look for the file in `C:\Netscape\SuiteSpot\docs\products\info.html`. If you change the document root (that is, you move all the files and subdirectories),

you only have to change the document root that the server uses, instead of mapping all URLs to the new directory or somehow telling clients to look in the new directory.

To set your server's primary document directory:

1. From the Server Manager, choose Content Management|Primary Document Directory.
2. In the Primary Directory field, type the full pathname of the directory that you want to make the primary document directory.
3. Click OK.
4. Click Save and Apply to confirm your changes.

Setting additional document directories

Most of the time, you keep all of your documents in the primary document directory. Sometimes, though, you may want to serve documents from a directory outside of your document root. You can do this by setting additional document directories. By serving from a document outside of your document root, you can let someone manage a group of documents without giving them access to your primary document root.

To add an additional document directory you first need to choose the URL prefix to map. Clients send this URL to the server when they want documents. Next, you specify the directory to map those URLs to. Finally, you might want to use an existing configuration style to specify how this directory should be configured.

To add additional document directories:

1. Choose Content Management|Additional Document Directories.
2. Type the URL prefix you want to map. (For example, a mapped URL could be `http://www.mozilla.com/`*`marketing/`*`index.html` where *`marketing/`* is the prefix you specify.)

3. Type the absolute path of the directory you want the URL prefix to map to. For example, it could be `C:\Netscape\marketing\pubdocs\index.html`.

4. If you want to, select a configuration style to apply to this directory's configuration.

5. Click OK.

Note When you update information, but don't save and apply changes, your information is retained so that you can view and edit it, even though the changes have not taken effect.

By default, the server has three additional document directories. They have the following prefixes:

- `/search-ui`
- `/webpub-ui`
- `/publisher`

You should restrict access to these directories so that users cannot write to them.

Configuring document preferences

You can configure the following document preferences from the Server Manager by selecting Content Management | Document Preferences.

Index filenames

If a document name is not specified in the URL, and the server finds a file with this name in a document directory, it assumes that file is the index file. The server automatically displays this file when no specific file is requested. The defaults are `index.html` and `home.html`. If more than one name is specified, the server looks in the order in which the names appear in this field

until one is found. For example, if your index filenames are `index.html,home.html`, the server first looks for `index.html`, and if it doesn't find it looks for `home.html`.

Directory indexing

In your document directory, you'll probably have several subdirectories. For example, you might create a directory called `products`, another called `people`, and so on. It's often helpful to let clients access an overview (or *index*) of these directories.

The server indexes directories using the following process:

- The server first searches the directory for an index file called `index.html` or `home.html`, which is a file you create and maintain as an overview of the directory's contents. (Note that these defaults are configurable for the whole server, so your server's files may vary. For more information, see "Index filenames" on page 53). You can specify any file as an index file for a directory by naming it one of these default names, which means you can also use a CGI program as an index if CGI is activated.
- If an index file isn't found, the server generates an index file that lists all the files in the document root. The generated index has one of the following formats:
    - "Fancy" directory indexing is fairly detailed. It includes a graphic that represents the type of file, the date the file was last modified, and the file size.
    - Simple directory indexing is less detailed but takes less time to generate.
    - You can also specify that no dynamic directory listing be generated if the server looks for index files and cannot find any. If the server does not find any index files, it will not create a directory listing to show the user and will return an error message.

Server home page

When users first access your server, they usually use an URL such as `http://www.mozilla.com/`. When the server receives a request for this document, it returns a document called a *home page*. Usually this file has general information about your server and links to other documents.

By default the server finds the index file specified in the Index filename field and uses that for the home page. However, you can also specify a file to use as the home page by setting the radio button to Home page and entering the file name for the home page in the field next to the radio button.

Default MIME type

When a document is sent to a client, the server includes a section that identifies the document's type, so the client can present the document in the right way. However, sometimes the server can't determine the proper type for the document because the document's extension is not defined for the server. In those cases, a default value is sent. For information about maintaining your server's MIME types, see "Configuring MIME types" on page 78.

The default is usually text/plain, but you should set it to the type of file most commonly stored on your server. Some common MIME types include the following:

- `text/plain`
- `text/richtext`
- `image/jpeg`
- `application/x-tar`
- `application/x-gzip`
- `text/html`
- `image/tiff`
- `image/gif`
- `application/postscript`
- `audio/basic`

Parsing the accept language header

When clients contact a server using HTTP 1.1, they can send header information describing the languages they accept. You can configure your server to parse this language information.

For example, if you store documents in Japanese and English, you could choose to parse the accept language header. When clients that have Japanese as the accept language header contact the server, they receive the Japanese version of the page. When clients that have English as the accept language header contact the server, they receive the English version.

If you do not support multiple languages, you should not parse the accept language header.

For more information on using the accept language header, see "Using the accept language header" on page 272.

Setting document preferences

To set document preferences:

1. From the Server Manager, choose Content Management|Document Preferences.
2. Type a new index filename, or add a file in the Index Filenames field.
3. Select the kind of directory indexing you want.
4. Select whether you want users to see a specified home page or an index file when they access your server.
5. Type the default MIME type you want the server to return if a client accesses a file with an extension that has not been set up as a MIME type on your server.
6. Choose whether to parse the accept language header.
7. Click OK.

Forwarding URLs

Redirection is a method for the server to tell a user that a URL has changed (for example, because you have moved files to another directory or server). You can also use redirection to seamlessly send a person who requests a document on one server to a document on another server.

To map a URL to another server, you must first specify the prefix of the URL you want the server to redirect. Then, you need to choose which URL to redirect to. You can redirect to a URL prefix if the directory on the new server is the same as in the mapped URL; you can also redirect to a fixed URL (hostname, directory, and filename).

To forward URLs:

1. From the Server Manager, choose Content Management | URL Forwarding.

2. Type the URL prefix you want to redirect. (For example, if the URL you want to map is `http://www.netscape.com/info/movies`, you'd type `/info/movies` in the field.)

3. Choose whether you want to forward requests to a URL prefix or to a fixed URL. If you forward to a URL prefix, the forwarding keeps the full path name, and substitutes one prefix for another. For example, if you forward `http://www.netscape.com/info/movies` to a prefix `mozilla.com`, the URL `http://www.netscape.com/info/movies` redirects to `http://mozilla.com/info/movies`.

   However, if the directory structure on the new server is not the same as in the mapped URL, you could forward the URL to a fixed URL. For example, you could forward `http://www.netscape.com/info/movies` to `http://mozilla.com/new-files/info/movies`.

   Sometimes you may want to redirect requests for all the documents in one sub-directory to a specific URL. For example, if you had to remove a directory because it was causing too much traffic, or because the documents were no longer to be served for any reason, you could direct a request for any one the documents to a page explaining why the

documents were no longer available. For example, a prefix on `/info/movies` could be redirected to `http://www.netscape.com/explain.html`.

4. Click OK.

Setting up hardware virtual servers

A hardware virtual server is a way to have your server respond to multiple IP addresses without installing multiple servers. With hardware virtual servers you map multiple IP addresses to multiple document roots. For example, if you have two IP addresses, you could map the first IP address to one document root and the second IP address to a second document root. The Netscape Enterprise Server can respond to up to 256 IP addresses.

Hardware virtual servers share the same server configuration information. For example, if you turn on encryption for one hardware virtual server, any other hardware virtual servers you create would also have encryption turned on.

If you need servers that respond to different IP addresses and require that they have separate configuration information, install separate instances of the server with specific IP addresses. For more information, see "Installing multiple instances of the web server" on page 43.

To set up hardware virtual servers:

1. From the Server Manager, choose Content Management|Hardware Virtual Servers.
2. Type the document root and its corresponding IP address in the appropriate fields.
3. Click OK.
4. Click Save and Apply to save and apply your changes.
5. Repeat the previous steps for each hardware virtual server.

Setting up software virtual servers

A software virtual server is a way to host several web sites on one computer without needing to have more than one IP address on it. For example, you can set up your system so that both `www.mozilla.com` and `www.netscape.com` resolve to 192.3.4.5, then set up software virtual servers to handle both server names (for example, `http://www.mozilla.com/` and `http://www.netscape.com)`.

The server can respond to requests differently depending upon the URL, even though the server only has one IP address. For example, one server can serve different pages for `http://www.mozilla.com/` and `http://www.netscape.com`.

The following example shows how software virtual servers might be used. An Internet service provider (ISP) installs a web server and then wants to set up a software virtual server for each of its customers (for example, customers *aaa*, *bbb*, and *ccc*) so that each customer can have an individual domain name.

The ISP first configures DNS to recognize that a customer's URL, `www.aaa.com`, resolves to the ISP's IP address. The ISP then creates a subdirectory for each company (`aaa`, `bbb`, and `ccc)` in the document root. These subdirectories contain the files for that company, including the home page, `aaa/home.html`. Next the ISP sets up software virtual servers. The URL host would be `www.aaa.com` and the home page would be `aaa/home.html`. The ISP would do this for all the companies.

Because software virtual servers use the HTTP Host header to direct the user to the correct page, not all client software works with software virtual servers. Only client software (such as Netscape Navigator) which supports the HTTP Host header works. In the previous example, the ISP would set up the `index.html` file in the document root to be an index page that links to all the virtual servers hosted by the system, so all users could access the home pages.

To set up a software virtual server:

1. From the Server Manager, choose Content Management|Software Virtual Servers.

2. Type the URL host whose custom home page you want to set up.

3. Type the path to the home page you want to use for this virtual server. If you type a full path, the server uses that specific document. If you type a partial path, the server interprets it as relative to your primary document directory.

4. Click OK.

5. Click Save and Apply to apply your changes.

Assigning a character set

The character set of a document is determined in part by the language it is written in. You can override Netscape Navigator's default character set setting for a document, a set of documents, or a directory by selecting a resource and entering a character set for that resource.

Netscape Navigator can use the MIME type `charset` parameter in HTTP to change its character set. If the server includes this parameter in its response, Netscape Navigator changes its character set accordingly. The following are some character set examples:

- `Content-Type: text/html;charset=iso-8859-1`
- `Content-Type: text/html;charset=iso-2022-jp`

The `charset` names recognized by Netscape Navigator are specified in RFC 1700 (except for the names that begin with `x-`). These `charset` names include the following:

- `us-ascii`
- `iso-2022-jp`
- `x-euc-jp`
- `iso-8859-1`
- `x-sjis`
- `x-mac-roman`

Additionally, the following aliases are recognized for `us-ascii`:

- `ansi_x3.4-1968`
- `ansi_x3.4-1986`
- `ascii`
- `us`
- `cp367`
- `iso-ir-6`
- `iso_646.irv:1991`
- `iso646-us`
- `ibm367`

The following aliases are recognized for `iso_8859-1`:

- `latin1`
- `iso_8859-1:1987`
- `ibm819`
- `iso_8859-1`
- *`iso-ir-100`*
- `cp819`

To change the character set:

1. From the Server Manager, choose Content Management | International Characters.
2. From the Resource Picker, choose the server resource for which you want to change the character set.
3. In the Character Set field, enter one of the character sets mentioned in the previous paragraphs.
4. Click OK.
5. Click Save and Apply for your changes to take effect.

Specifying a document footer

You can specify a document footer, which can include the last-modified time, for all the documents in a certain section of your server without using server-parsed HTML. This footer works for all files except output of CGI scripts or parsed HTML (`.shtml`) files. If you need your document footer to appear

on CGI-script output or parsed HTML files, enter your footer text into a separate file and add a line of code or another server-side include to append that file to the page's output.

To specify a document footer:

1. From the Server Manager, choose Content Management|Document Footer.
2. From the Resource Picker, choose the resource to which you want to apply the document footer.
3. Type the kind of files you want to include in the footer. The default is `text/html`.
4. Choose the time format from the drop-down list, or type a custom date format in the Custom Date Format field.
5. Enter the footer text. The maximum number of characters for a document footer is 765. Type the string `:LASTMOD:` if you want to include the date the document was last modified.

Note Any entities (for example, `&copy`) are contracted after you save your changes.

6. Click OK.
7. Click Save and Apply to confirm your changes.

When you change the document footer for an HTML page, the last-modified date doesn't change.

Customizing parsed HTML

HTML is normally sent to the client exactly as it exists on disk without any server intervention. However, the server can search HTML files for special commands (that is, it can *parse* the HTML) before sending documents. If you want the server to parse these files and insert request-specific information or files into documents, you must first enable HTML parsing.

To customize parsed HML:

1. From the Server Manager, choose Content Management | Parse HTML.
2. From the Resource Picker, choose the server resource to edit.
3. Choose whether you want to activate parsed HTML. If you activate it, you need to choose whether to activate it with or without the `exec` tag. The `exec` tag allows an HTML file to execute an arbitrary program on the server. You might not want to allow the `exec` tag for security or performance reasons.
4. Choose which files to parse.
   - The common (and default) choice is to parse only files with the extension `.shtml`. In this case, all files you want to parse must have the `.shtml` extension.
   - You can have the server parse all of its HTML files. Choosing this option can slow your server's performance.

Using cache-control directives

Cache-control directives are a way for the Netscape Enterprise Server to control what information is cached by a proxy server. Using cache-control directives, you override the default caching of the proxy to protect sensitive information from being cached, and perhaps retrieved later. For these directives to work, the proxy server must comply with HTTP 1.1.

For specific directories in your server, you can set the cache-control directives to one of the following levels:

- Public. The response is cacheable by any cache.
- Private. The response is only cacheable by a private (nonshared) cache.
- No cache. The response must not be cached anywhere.
- No store. The cache must not store the request or response anywhere in nonvolatile storage.

- Must revalidate. The cache entry must be revalidated from the originating server.
- Maximum age (in seconds). The client does not accept a response that has an age greater than this age.

To set the cache-control directives:

1. Choose Content Management|Cache Control Directives.
2. From the Resource Picker, choose the directory or directories for which you want to set cache-control directives.
3. Choose the level of control you want to set. The default is public.
4. Click OK.

For more information HTTP 1.1, see the Hypertext Transfer Protocol—HTTP/1.1 specification (RFC 2068) at:

`http://www.ietf.org/html.charters/http-charter.html`

Working with configuration styles

Configuration styles are an easy way to apply a set of options to specific files or directories that your server maintains. For example, you can create a configuration style that sets up access logging. When you apply that configuration style to the files and directories that you want to log, you don't have to individually configure access logging for all the files and directories.

Creating a configuration style

To create a configuration style:

1. From the Server Manager, choose Configuration Styles|New Style.
2. Type the name you want to give the configuration style.
3. Click OK. The Edit Configuration Style form appears.

4. From the drop-down list, choose a configuration style to edit and click "Edit this Style."

5. From the list of links available, click the category you want to configure for your style. You can configure the following information:

   - CGI file type—Allows you to activate CGI as a file type. For more information about CGIs, see "Installing CGI programs" on page 131.

   - Character Set—Allows you to change the character set for a resource. For more information about character sets, see "Assigning a character set" on page 60.

   - Default Query Handler—Allows you to set a default query handler for a server resource. For more information about query handling, see "Using the query handler" on page 138.

   - Document Footer—Allows you to add a document footer to a server resource. For more information about document footers, see "Specifying a document footer" on page 61.

   - Dynamic Configuration—Allows you to give people a subset of configuration options without giving them access to the Server Manager. For more information about dynamic configuration, see "Working with dynamic configuration files" on page 82.

   - Error Responses—Allows you to customize the error responses that clients see when they encounter an error from your server. For more information about error responses, see "Customizing error responses" on page 81.

   - Log preferences—Allows you to set preferences for access logs. For more information about log preferences, see "Setting log preferences" on page 152.

   - Restrict Access—Allows you to restrict access to the entire server or parts of it. For more information about access control, see "Restricting access" on page 96.

   - Server Parsed HTML—Allows you to specify whether the server parses files before they are sent to the client. For more information about using parsed HTML, see "Customizing parsed HTML" on page 62.

6. Fill out the form that appears, and then click OK.
7. Repeat Step 5 and Step 6 to make any other configuration changes to the configuration style.
8. Click OK.
9. Click Save and Apply to confirm your changes to the configuration style.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker.

Editing a configuration style

To edit a configuration style:

1. From the Server Manager, choose Configuration Styles|Edit Style.
2. From the drop-down list, choose a configuration style to edit.
3. Click "Edit this style."
4. From the list of links available, click the category you want to configure for your style. For more information on these categories, see "Creating a configuration style" on page 64.
5. Fill out the form that appears, and then click OK.
6. Repeat Step 4 and Step 5 to make any other changes to the configuration style.
7. Click OK.
8. Click Save and Apply to confirm your changes to the configuration style.

When you choose a style to edit, your Resource Picker lists configuration styles instead of other resources. After you have finished editing a style, click OK and Save and Apply. The Resource Picker exits the styles mode. You can also choose to exit the styles mode by choosing Exit styles mode from the Resource Picker.

Applying a configuration style

Once you've created a configuration style, you can apply it to files or directories in your server. You can specify either individual files and directories or wildcard patterns (such as `*.gif`).

To apply a configuration style:

1. Choose Configuration Styles|Assign Configuration Style.
2. Enter the prefix of the URL to which you are applying this configuration style. If you choose a directory inside the document root, only enter the path after the document root. If you enter `/*` after the directory, you apply the configuration style to all of the directory's contents.
3. Select the configuration style you want to apply. To remove any configuration style previously applied to the resource, apply the `None` configuration style.
4. Click OK.

Removing a configuration style

Before removing a configuration style, apply the `None` configuration style to any files or directories that had the configuration style applied to them. If you do not do this before removing the configuration style, you must manually edit your `obj.conf` file, searching for the configuration style in the file and replacing it with `None`. If you don't do this search and replace, anyone who accesses the files or directories that had the deleted configuration style applied will get a server misconfiguration error message.

To remove a configuration style:

1. Choose Configuration Styles|Remove Configuration Style.
2. Select the configuration style you want to remove.
3. Click OK. The configuration style is removed.

Listing configuration style assignments

After you have created configuration styles and applied them to files or directories, you can get a list of the configuration styles and where you applied them.

To list configuration style assignments:

1. From the Server Manager, choose Configuration Styles|List Assignments. The List Assignments form appears, showing the configuration styles you applied to server resources.
2. To edit a configuration style assignment, click the Edit link next to the configuration style name.

Chapter

5

Configuring server preferences

This chapter describes how to configure server preferences for your Netscape Enterprise Server by using the Server Manager configuration forms.

Starting and stopping the server

Once installed, the server runs constantly, listening for and accepting requests. If your server is running, you'll see the On icon and its green light (to the left of the server's name) in the Server Administration page, as shown in the graphic at left. You can start and stop the server by clicking the icon. You can also start, restart, and stop the server from the Server Manager or the Control Panel.

To start or stop the server from the Server Manager:

1. Choose Server Preferences | On/Off.
2. Click the On or Off button.

   If your server is on and you click Server On, the server will restart. If you've turned on or restarted your server, access it as a client by clicking "Access *server_name* as a client."

Note After you shut down the server, it may take a few seconds for the server to complete its shut-down process and for the status to change to "Off."

If your machine crashes or is taken offline, the server stops and any requests it was servicing may be lost.

Setting the termination timeout

When you stop your server, the server stops accepting new connections. Then it waits for all outstanding connections to complete. The time the server waits before timing out is configurable in the `magnus.conf` file. By default it is set to 3 seconds. You probably do not need to change this value. If you do need to change the value, add a line in `magnus.conf` as follows:

```
TerminateTimeout seconds
```

where *seconds* represents the number of seconds you want to wait before timing out.

The advantages to configuring this value is that if for some reason you want to wait longer for connections to complete, you can. However, because most servers often have connections open from nonresponsive clients, if you increase the time the server waits, you will almost always have to wait the full time before your server shuts down.

Restarting the server

You can restart the server using one of the following methods:

- Use the Control Panel Services to restart any server.
- Use the Control Panel Services to configure the operating system to restart the server or the administration server each time the machine is restarted.

  For Windows NT 3.51:

  1. In the Main group, double-click the Control Panel icon.
  2. Double-click the Services icon.
  3. Scroll through the list of services and select the service for your server.

4. Check Automatic to have your computer start the server each time the computer starts or reboots.

5. Click OK.

For Windows NT 4.0:

1. From the Start menu, choose Settings|Control Panel.

2. Double-click the Services icon.

3. Scroll through the list of services and select the service for your server.

4. Check Automatic to have your computer start the server each time the computer starts or reboots.

5. Click OK.

Note You can also use the Services dialog box to change the account the server uses. For more information about changing the account the server uses, see "Changing the user account" on page 35.

Normally, you can't start an SSL-enabled server automatically because you have to enter its password. There is a way to have an SSL-enabled server start without having to enter a password if you keep the password in plain text in a text file. This practice is *not* recommended.

Caution! Leaving your SSL-enabled server's password in a text file on your system is a large security risk. In essence, you are trading security for convenience. Anyone who can access the file has access to your SSL-enabled server's password. Consider whether you can afford the security risks before keeping your SSL-enabled server's password in plain text on your system.

If the security risk is not a concern for you, follow these steps to start your SSL-enabled server automatically:

1. Using a text editor, such as Notepad, create a new text file called `password.txt` in *`server_root`*`\https-`*`server_identifier`*`\config`. For a default web server installation, `password.txt` would be stored in the `C:\Netscape\SuiteSpot\https-`*`server_id`*`\config` directory.

2. Type your private-key password in the first line, making sure not to put carriage returns or linefeeds after the password. The file must contain only the password.

   When you start your SSL-enabled server, it first tries to read the password in `password.txt`. If the file does not exist, you will be prompted for the password. If `password.txt` exists but the password is incorrect, the server will add an entry to the error log and exit.

Caution! If you have an NTFS file system, you should protect the directory that contains `password.txt` by restricting its access, even if you do not use the file. The directory should have read/write permissions for the administration server user and the web server user. Protecting the directory prevents others from creating a false `password.txt` file.

On FAT file systems, you cannot protect directories or files by restricting access to them.

Using the automatic restart utility

The server is automatically restarted by a server-monitoring utility if the server crashes. On systems that have debugging tools installed, a dialog box with debugging information appears if the server crashes. To help debug server plug-in API programs (WAI or NSAPI programs), you can disable the auto-start feature by setting a very high timeout value. You can also turn off the debugging dialog boxes by using the Registry Editor.

Changing the time interval

To change the time interval that elapses between startup and the time the server can automatically restart:

1. Start the Registry Editor.
2. Select your server's key (in the left side of the Registry Editor window, located in `HKEY_LOCAL_MACHINE\SOFTWARE\Netscape\`).
3. Choose Edit|Add Value. The Add Key dialog box appears.
4. In the Value Name box, type `MortalityTimeSecs`.

5. Select REG_DWORD from the Data Type pull-down list.
6. Click OK. The DWORD Editor dialog box appears.
7. Type the time interval (in seconds) that will elapse between startup and the time the server can restart automatically. The interval can be in binary, decimal, or hexadecimal format.
8. Click the numerical format for the value you entered in the previous step (binary, decimal, or hexadecimal).
9. Click OK. The `MortalityTimeSecs` value appears in hexadecimal format at the right side of the Registry Editor window.

Turning off the debugging dialog box

If you've installed an application (such as a compiler) that has modified the system debugging settings and the server crashes, you might see a system-generated application error dialog box. The server will not restart until you click OK.

To turn off the debugging dialog box that appears if the server crashes:

1. Start the Registry Editor.
2. Select the `AeDebug` key, located in the left side of the Registry window in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT \CurrentVersion`.
3. Double-click the Auto value in the right side of the window. The String Editor dialog box appears.
4. Change the string value to 1.

Viewing server settings

You can view your server's technical and content settings from the Server Manager. You can also see if your server is running. The technical settings come from `magnus.conf`, and the content settings come from `obj.conf`. These files are located in the server root, in the directory

`https-`*`server_name`*`\config`. For more information about the `magnus.conf` and `obj.conf` files, see The NSAPI Programmer's Guide on Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

The following list explains the server's technical settings:

- **Server Root** is the directory where the server binaries are kept. You first specified this directory during installation.
- **Hostname** is the URL clients use to access your server.
- **Port** is the port on your system that the server listens to for HTTP requests.
- **Error log** is the name and path of the server's error log file.
- **MTA host** is the name of the mail server (used by agents).
- **NNTP host** is the name of the news server (used by agents).
- **DNS** shows whether DNS is enabled or disabled.
- **Security** shows whether SSL is enabled or disabled.
- **Asynch DNS** shows whether asynchronous DNS is enabled or disabled.

The server's content settings depend on how you've configured your server. Common server content settings include the server's document directory, its index filenames, name and location of its access log, and default MIME type.

Restoring backup configuration files

You can view or restore a backup copy of your configuration files (`https-`*`server_id`*`.acl`, `magnus.conf`, `obj.conf`, `webpub.conf`, `agent.conf`, `mime.types`, `.acl` files, `rdm.conf`, `csid.conf`, `process.conf`, `robot.conf`, and `filter.conf`).

To view or restore a backup copy of your configuration files:

1. From the Server Manager, choose Server Preferences | Restore Configuration.
2. Set the number of backups. The number you type here is the number of backups displayed on the form. Enter the number and click Change to change the number.
3. If you want to view a backup version, click the View button next to the version you want. Click Restore if you want to restore that version. To restore all files to their state at a particular time, click the Restore to *time* button, which lists the specific time to which you want to restore.

Tuning server performance

You can configure the server's technical options, including the number of maximum simultaneous requests, listen-queue size, and DNS usage.

Configuring maximum simultaneous requests

You can set the number of maximum simultaneous requests, which is the number of active requests allowed for the server at one time. However, for general purpose Internet or intranet use, you probably will not need to change the default value (128 requests).

To get the number of simultaneous requests, the server counts the number of active requests, adding 1 to the number when a new request arrives, subtracting 1 when it finishes the request. When a new request arrives, the server checks to see if it is already processing the maximum number of requests. If it has reached the limit, it defers processing new requests until the number of active requests drops below the maximum amount.

In theory, you could set the maximum simultaneous requests to 1 and still have a functional server. Setting this value to 1 would mean that the server could only handle one request at a time, but since HTTP requests generally have a

very short duration (response time can be as low as 5 milliseconds), processing one request at a time would still allow you to process up to 200 requests per second.

However, in actuality, Internet clients frequently connect to the server and then do not complete their requests. In these cases, the server waits 30 seconds or more for the data before timing out. Also, some sites do heavyweight transactions that take minutes to complete. Both of these factors add to the maximum simultaneous requests that are required. If your site is processing many requests that take many seconds, you may need to increase the number of maximum simultaneous requests.

If you need to change the number of maximum simultaneous requests, set the number before starting the server. To reset the number:

1. Choose Server Preferences|Performance Tuning.
2. Type the number of requests.
3. Click OK.
4. Click Save and Apply.

Enabling Domain Name System lookups

You can configure the server to use Domain Name System (DNS) lookups during normal operation. By default, DNS is not enabled; if you enable DNS, the server looks up the host name for a system's IP address. Although DNS lookups can be useful for server administrators when looking at logs, they can impact performance. When the server receives a request from a client, the client's IP address is included in the request. If DNS is enabled, the server must look up the hostname for the IP address for every client making a request.

DNS causes multiple threads to be serialized when you use DNS services. If you do not want serialization, enable asynchronous DNS. You can enable it only if you have also enabled DNS. Enabling asynchronous DNS can improve your system's performance if you are using DNS.

Note Turning off DNS lookups on your server has the following consequences: host name restrictions won't work, and hostnames won't appear in your log files. Instead, you'll see IP addresses.

You can also specify whether to cache the DNS entries. If you enable the DNS cache, the server can store hostname information after receiving it. If the server needs information about the client in the future, the information is cached and available without further querying. You can specify the size of the DNS cache and an expiration time for DNS cache entries. The DNS cache can contain 32 to 32768 entries; the default value is 1024 entries. Values for the time it takes for a cache entry to expire can range from 1 second to 1 year (specified in seconds); the default value is 1200 seconds (20 minutes).

Configuring listen-queue size

The listen-queue size is a socket-level parameter that specifies the number of incoming connections the system will accept for that socket. The default setting is 100 incoming connections.

Note Normally, you should *not* change the listen-queue size. The default setting is sufficient in most cases.

If you manage a heavily used web site, you should make sure your system's listen-queue size is large enough to accommodate the listen-queue size setting from the Server Manager form. If you do change the listen-queue size, make sure that your system supports the new size. The listen-queue size set from the Server Manager form changes the listen-queue size requested by the server. If the server requests a listen-queue size larger than the system's maximum listen-queue size, the size defaults to the system's maximum.

Caution! Setting the listen-queue size too high can degrade server performance. The listen-queue size was designed to prevent the server from becoming overloaded with connections it cannot handle. If your server is overloaded and you increase the listen-queue size, the server will only fall further behind.

Configuring the HTTP persistent connection timeout

With HTTP 1.1, a connection can be set to be persistent (similar to keep alive in HTTP 1.0). However, even if a connection is persistent, it still needs to have a timeout setting, or it may consume system resources.

Note Normally, you should *not* change the persistent connection timeout. The default setting is sufficient in most cases.

If you need to change the setting:

1. From the Server Manager, choose Server Preferences | Performance Tuning.
2. Enter a number in seconds in the HTTP Persistent Connection Timeout field.
3. Click OK.
4. Save and apply your changes.

Configuring MIME types

MIME (Multi-purpose Internet Mail Extension) types control what types of multimedia files your mail system supports. You can also use MIME types to specify what file extensions belong to certain server file types, for example to designate what files are CGI programs. For more information on using file extensions with programs, see "Installing CGI programs" on page 131.

To add a new MIME type:

1. Choose Server Preferences|Mime Types.
2. Select the category and enter the content type and file suffix.
3. Click New Type.

To edit a MIME type:

1. Click Edit next to the type you want to edit.
2. Change the category, content type, and file suffix as needed.
3. Click Change MIME Type to update.

To remove a MIME type, click Remove next to the type you want to remove.

Note Do not put spaces between the file suffixes when you are adding or editing a MIME type. If you put a space between them, you may receive an error or your server may not restart. If this happens, edit your `mime.types` file to delete the

space. The `mime.types` file is in your server root in the `https-`*`identifier`*`\config` directory. After you have edited the file, from the Server Manager, use the Apply button to apply your manual changes.

Configuring network settings

You can change your server's network settings using the Server Manager.

Changing the server's user account

By using a specific user account (other than `LocalSystem`), you can restrict or enable system features for the server. For example, you can use a user account that can mount files from another machine.

To change the web server user account after installation:

1. Create a user with the Windows NT Users Manager. The user must have "Log in as a service" rights.
2. Stop the server.
3. From the Windows Control Panel, choose Services.
4. Select the Netscape Enterprise Server 3.0 service.
5. In the Service pop-up, in the Log on As section, click the This Account radio button.
6. Type the user account you want the web server to use.
7. Type the password for that account; type it again for confirmation.
8. Click OK.
9. Restart the server using the Services program or the Server Administration page.

Changing the server name

The server name is the full hostname of your server machine. When clients access your server, they use this name. The format for the server name is *machinename.yourdomain.domain*. For example, if your full domain name is `netscape.com`, you could install a server with the name `www.netscape.com`.

If your system administrator has set up a DNS alias for your server, use that alias on the Network Preferences form. If not, use the machine's name combined with your domain name to construct the full hostname.

Changing the server port number

On the Network Preferences form. Server Port Number specifies the TCP port that the server listens to. The port number you choose can affect your users—if you use a nonstandard port, then anyone accessing your server must specify a server name and port number in the URL. For example, if you use port 8090, the user would specify something like this URL:

```
http://www.netscape.com:8090
```

Port numbers for the most commonly used network-accessible services are maintained in the file `\WINNT\System32\drivers\etc\services`. The standard unsecure web server port number is 80; the standard secure web server port number is 443. Technically, the port number can be any port from 1 to 65535.

Changing the server binding address

At times you'll want the server machine to answer to two URLs. For example, you might want to answer both `http://www.netscape.com/` and `http://www.mozilla.com/` from one machine.

If you have already set up your system to listen to multiple IP addresses and want to use this feature, on the Network Preferences form use the Bind To Address field to tell the server which IP address is associated with this hostname.

Changing the server's MTA host

To change the MTA (Message Transfer Agent) host, use the MTA Host field on the Network Preferences form to change the name of the SMTP mail server. You must enter a valid MTA host if you want to use the agent email function.

Changing the server's NNTP host

To change the NNTP (Network News Transfer Protocol) host, use the NNTP Host field to change the name of the news server. You must enter a valid NNTP host if you want to use agents with the capability to post to news.

Customizing error responses

You can specify a custom error response that sends a detailed message to clients when they encounter errors from your server. You can specify a file to send or a CGI program to run.

You might want to change the way a the server behaves when it gets an error for a specific directory. Instead of sending back the default file, you might want to send a custom error response instead. For example, if a client tries repeatedly to connect to a part of your server protected by access control, you might return an error file with information on how to get an account.

What are the errors?

You can customize the response to several different kinds of errors:

- **Unauthorized**. This error occurs when users without access permissions try to access a document on the server that is protected by access control. You might send information on how they can get access.
- **Forbidden**. This error occurs when the server doesn't have file system permissions to read something, or if the server is not permitted to follow symbolic links.

- **Not Found**. This error occurs when the server can't find a document or when it has been instructed to deny the existence of a document.
- **Server Error**. This error occurs when the server is not configured properly or when a catastrophic error occurs, such as the system running out of memory or producing a core dump.

Setting up the response

Before you can set up the response, you need to write the HTML file to send or create the CGI program to run. After you do this, set the response by doing the following:

1. From the Server Manager, choose Server Preferences|Error Responses.
2. From the Resource Picker, choose the server resource you want to configure.
3. Select the error response you want to customize.
4. Type the absolute pathname to the file or CGI script that you want to return for that error code. Check the CGI box if the file is a CGI program that you want to run.

   Repeat this process for each of the error responses you want to customize.
5. Click OK.
6. Click Save and Apply to confirm your changes.

To remove a customization, return to the form and delete the filename from the text box next to the error code.

Working with dynamic configuration files

Server content is seldom managed entirely by one person. You may need to allow end users to access a subset of configuration options so that they can configure what they need to, without giving them access to the Server Manager. The subset of configuration options are stored in dynamic configuration files.

Two types of dynamic configuration files are supported by the Netscape Enterprise Server: `.htaccess` and `.nsconfig`. You can enable `.nsconfig` files in the Server Manager; you have to manually enable `.htaccess` files.

Note There is no support for LDAP or the 3.0 Netscape user databases in the dynamic configuration files. You should not use dynamic configuration files if you use LDAP. You must use NCSA-style user databases to use `.htaccess` files. You must use either NCSA-style user databases or Enterprise 2.x DBM-format user databases with `.nsconfig` files. For more information on user databases, see *Managing Netscape Servers*.

If you already use `.nsconfig` files, you might want to continue using them. However, Netscape also provides a utility for converting your `.nsconfig` files to `.htaccess` files.

Using .htaccess files

The files that support `.htaccess` are in the server root, in `plugins\htaccess`. These files include a plug-in that enables you to use `.htaccess` files and a script for converting `.nsconfig` files to `.htaccess` files.

Activating .htaccess checking

To use `.htaccess` files, you must first modify the server's `obj.conf` file to load, initialize, and activate the plug-in. At the top of the `obj.conf` file, after the other `Init` directives, add the following lines:

```
Init fn="load-modules" funcs="htaccess-init,htaccess-find" \
shlib="server_root\plugins\htaccess\htaccess.dll"
Init fn="htaccess-init"
```

These lines load and initialize the module when the server is started. *`server_root`* is the path to your server root.

To activate `.htaccess` file processing for all directories managed by the server, add the `PathCheck` directive:

```
PathCheck fn="htaccess-find"
```

to the default server object, which is delimited by:

```
<Object name="default">
...
</Object>
```

Generally, the directive to activate `.htaccess` processing should be the last `PathCheck` directive in the object.

To activate `.htaccess` file processing for particular server directories, place the `PathCheck` directive in the corresponding object definition in `obj.conf`.

If you want to name your `.htaccess` files something other than `.htaccess`, you need to specify the filename in the `PathCheck` directive using the following format:

```
PathCheck fn="htaccess-find" filename="filename"
```

Replace *`filename`* with the filename you are using.

After editing the configuration file, stop and start your server. Apply your configuration file changes in the Server Manager by clicking the Apply button. Subsequent accesses to the server will be subject to `.htaccess` access control in the specified directories.

To restrict write access to .`htaccess` files, create a configuration style for them, and apply access control to that configuration style. For more information, see "Working with configuration styles" on page 64, and Chapter 6, "Controlling access to your server."

Converting existing .nsconfig files to .htaccess files

The Netscape Enterprise Server includes a script for converting your existing `.nsconfig` files to `.htaccess` files. To convert your files, at the command prompt, enter the path to Perl on your system, the path to the script, and the path to your `obj.conf` file. For example you might type the following (it should all be on one line when you type it):

```
server_root\install\perl server_root/plugins/htaccess/htconvert server_root/
https-server_name/config/obj.conf
```

The script converts all `.nsconfig` files to `.htaccess` files, but does not delete the `.nsconfig` files.

Supported .htaccess directives

The following `.htaccess` directives are supported in this release:

- `AuthName`
- `AuthType`
- `AuthUserFile`
- `AuthGroupFile`
- `Limit`
  - `order`
  - `deny`
  - `allow`
  - `require`

The only `AuthType` supported is `Basic`.

Example of an .htaccess file

The following example shows an `.htaccess` file:

```
<Limit GET POST>
order deny,allow
deny from all
allow from all
</Limit>
<Limit PUT DELETE>
order deny,allow
deny from all
</Limit>
AuthName mxyzptlk.kawaii.com
AuthUserFile /server_root/mxyz-docs/service.pwd
AuthGroupFile /server_root/mxyz-docs/service.grp
```

Using .nsconfig files

With `.nsconfig` files, you can allow end users to apply access control or customize error messages without allowing them to use CGI or parsed HTML. The format and capability of these dynamic configuration files is described in "Writing .nsconfig files" on page 87.

When a request is made for a resource in which dynamic configuration is enabled, the server must search for the configuration files within one or more directories of that resource. This search can be an expensive operation in terms of performance, so the server lets you configure how much flexibility you need, weighing it against the efficiency cost.

You can provide a base directory to the server, in which case the server starts its search for configuration files from the filesystem directory. Alternatively, you can provide no base directory, in which case the server attempts to infer the base directory from the URL. That is, if the requested URL is to a file in the document root, the server starts searching from the document root.

You also specify the name of the configuration file to search for within the base directory.

If you centralize all of your configuration information for the subdirectories of the base directory in the base directory's configuration file, the server is more efficient because it doesn't have to search for configuration files in the subdirectories.

However, you may sometimes want the server to search the subdirectories. If you do, the server searches for `.nsconfig` files starting from the top level directory and searching downward until reaching the directory in which the referenced resource resides. The server processes `.nsconfig` files in the order it encounters them. If a top level file restricts a user's access, the server does not give the user access, even though a lower level file might allow access.

The server processes all restrictions based on IP address and DNS entry (`RestrictAccess` directive) as it finds them in a file. If the server finds a file that denies a user access because of IP address or DNS entry, it stops processing files. The server collects restrictions based on user name (`RequireAuth` directive) and processes them at the end, unless the request has already been denied because of IP address or DNS entry.

For example, if you selected the base directory inferred from URL translation, selected `.nsconfig` for your filename, and chose to search subdirectories, the following search would occur.

When a user requests the filesystem path `C:\Netscape\SuiteSpot\docs\icons\gfx\logo.gif`, instead of searching for `C:\Netscape\SuiteSpot\docs\.nsconfig` the server would search all of the subdirectories:

```
C:\Netscape\SuiteSpot\docs\.nsconfig
C:\Netscape\SuiteSpot\docs\gfx\.nsconfig
C:\Netscape\SuiteSpot\docs\gfx\icons\.nsconfig
```

You can also enter a wildcard pattern of file types you want to disable in directories where dynamic configuration is enabled. To disable CGI programs and parsed HTML, for example, use * (`cgi|parsed-html`).

To configure `.nsconfig` files:

1. Choose Server Preferences | Dynamic Configuration Files.
2. Choose a resource from the Resource Picker.
3. Choose whether to base the directory from the URL or from a specified directory.
4. Type the filename.
5. Choose whether to search only the base directory.
6. Type the disabled types.
7. Click OK
8. Click Save and Apply.

Writing .nsconfig files

The `.nsconfig` files consist of sets of directives that control the server. These directives are surrounded by `Files` directives that tell the server which files in the configuration file's directory the directives apply to. For example:

```
<Files PATTERN1>
... directives ...
</Files>
<Files PATTERN2>
... directives ...
</Files>
```

PATTERN1 and *PATTERN2* are wildcard patterns that tell the server which filesystem pathnames to apply the directives to. For example, * would apply the directive to all filesystem pathnames. Any pattern given is first prefixed with the directory containing the configuration file to ensure that the directives are applied only to subdirectories. There can be as many sets of `Files` directives in the `.nsconfig` file as you need.

The file can contain blank lines. Lines beginning with # are treated as comments.

All paths must use the forward slash (/) instead of the backwards slash (\), otherwise you receive a server "path not found" error.

Each directive can take a variable number of parameters. The `Files` directives include:

- `AddType exp=`*`SHEXP`* `type=`*`mime-type`* `enc=`*`http-encoding`*

  `AddType` assigns the give type or encoding to the paths represented by the wildcard pattern *SHEXP* . One or both of `type` and `encoding` can appear, but only one expression can be used. You cannot apply two MIME types or encodings to the same pattern of the files.

- `ErrorFile reason=`*`error-string`* `code=`*`error-code`* `path=`*`html-file`*

  `ErrorFile` causes the HTML file described by the URL suffix *path* to be sent in place of the server's default error message. The file is substituted when an error described by one or both of `reason` and `code` occurs. `path` is a valid URL to the local server but without the `http://`*`server`* prefix. The error codes are the standard HTTP error codes:

  - 401 Unauthorized
  - 403 Forbidden

  - 404 Not found
  - 500 Server error

- `RequireAuth dbm=`*`dbmfile`* `userfile=`*`database_name`* `realm=`*`string`* `userpat=`*`PATTERN`*

  `RequireAuth` lets you ask the user for a username and a password when accessing the directory. `dbm` is a user database. Please note that `dbm` can only be used on a 2.x Enterprise user database. `userfile` is an NCSA-style user database filename. The file consists of lines in the format *`user:encrypted_password`*`.` `realm` is a unique string to tell your users which password they should use. `userpat` determines which users from the given `dbm` or `userfile` are allowed access. `userpat` is a wildcard pattern or list of user names. For example, you can use the syntax `userpat="user1"` or `userpat="(user1|user2)"` for specifying a user or a list of users.

- `RestrictAccess method=HTTP-method type=allow|deny ip=addrpattern dns=hostpattern return-code=403|404`

  `RestrictAccess` applies access control to the directory and restricts certain users. `method` is an optional parameter specifying a wildcard pattern of HTTP methods to protect (no method specified means all of them). `type` determines whether the IP address wildcard pattern or hostname wildcard pattern is allowed or denied access. If the only `RestrictAccess` directives in a `Files` set are of type `allow`, then all hosts not specified by the patterns are denied. `ip` must be typed in lowercase for the directive to work.

  More than one `RestrictAccess` can appear in the file. The order in which these lines appear is important; later `RestrictAccess` lines override earlier ones.

Example of an .nsconfig file

The following example shows an `.nsconfig` file:

```
<Files *>
ErrorFile reason="Unauthorized" code="401" path="/errors/unauthorized.html"
ErrorFile reason="Forbidden" code="403" path="/errors/forbidden.html"
ErrorFile reason="Not Found" code="404" path="/errors/notfound.html"
ErrorFile reason="Server Error" code="500" path="/errors/server-error.html"
RestrictAccess method="(GET|HEAD|POST)" type="allow" ip="*"
RestrictAccess method="(GET|HEAD|POST)" type="deny" ip="198.95.251.30" return-code="404"
</Files>
<Files *.gif>
AddType exp=*.gif type=application/octet-stream
</Files>
<Files *.txt>
RequireAuth dbm="server_root/authdb/default" realm=Text userpat="user*"
</Files>
```

Chapter

6

Controlling access to your server

You can control who accesses the files on your web site. This chapter discusses the various methods you can use to determine who has access to what files or directories on your web site. If you want to control who can configure the web server itself and who can access the server configuration files, see *Managing Netscape Servers*. You should also ensure the security of your web server's computer, as discussed in Chapter 7, "Using encryption and SSL."

What is access control?

Access control lets you determine who can access the server. You can use two attributes for controlling access:

- **User-Group** requires users to enter a username and password before accessing the server. Or, the server can use client authentication by checking an LDAP directory for a security certificate before giving access to a file or set of files on your web site.
- **Host-IP** requires the user to view your web site from a specific computer, where the server recognizes the computer by either its hostname or its IP address.

User-Group authentication

You can require users to *authenticate* themselves before getting access to your web site. Authentication means that users verify their identity either by entering a username and password or by using a client certificate installed in their network browser, such as Netscape Navigator or Netscape Communicator. The first method of getting the username and password is the "Basic" method, which can be done with or without encryption. The second method of using client certificates is the "SSL" method, which must be done with encryption on.

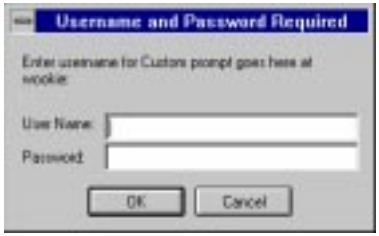
Username and password authentication

If you require users to enter a username and password to get access to your web site, you store the list of users and groups in an LDAP database, which can be either a file stored on the web server computer or an LDAP server on a remote computer (for example, a computer running Netscape Directory Server).

When users attempt to access a file or directory that has User-Group authentication, the web browser displays a dialog box asking the user to enter a username and password. The server can get this information encrypted or not, depending on whether encryption is turned on for your server.

After entering the information, the user either sees the file or directory listing requested or a message denying access. (You can customize the access-denied message that they see.) Figure 6.1 shows the authentication window. This window shows a custom message.

Figure 6.1 Users see this window when authenticating themselves to the server.



Note If your server doesn't use SSL encryption, the username and password that the end user types are sent unencrypted across the network. Someone could intercept the network packets and read the username and password being sent to the web server. For this reason, User-Group authentication is most effective when combined with SSL encryption or Host-IP authentication, or both.

Client certificate authentication

You can confirm users' identities with security certificates before giving the users access to your web site. You can do this in two ways:

- The server can use the information in the certificate as proof of identity.
- The server can verify the certificate itself provided the certificates are published in an LDAP directory.

When a request comes in and you have client authentication on, the server performs these actions in order:

1. When the browser sends the certificate, the server checks if the certificate is from a trusted CA. If not, the server ends the transaction.
2. If the certificate is from a trusted CA, the server maps the certificate to a user's entry using the `certmap.conf` file. (See *Managing Netscape Servers* for more information on setting up the certificate mapping file.)
3. If the certificate maps correctly, then the web server follows the ACL rule specified for that user. The rule can deny or allow the request.

New in 3.x In version 2.0, users had to enter a username and password along with sending a certificate to the server. In 3.x, the web server looks up the entry in an LDAP directory, so the access appears seamless to the end user.

Requiring client authentication for controlling access to specific resources is different than requiring client authentication for *all* connections to the server. To require client authentication for the entire server, you check a box in the Encryption Preferences form. To require client authentication with access control, you use the SSL authentication method.

Users must first install a client certificate in their network browser and then optionally publish the certificate in an LDAP directory, such as Netscape Directory Server.

Host-IP authentication

You can limit access to files and directories on your web site by making them available only to people using specific computers. You specify hostnames or IP addresses for the computers that you want to allow or deny. You can use wildcard patterns to specify multiple computers or entire networks. If you want to use this feature, you must have DNS running in your network and your computer must be configured to use it.

End-user access to a file or directory using Host-IP authentication appears seamless. Users can access the files and directories immediately without entering a username or password. If the computer doesn't have access, the user will get a message denying access. (You can also customize this message.)

Note It's possible for more than one person to have access to a particular computer. For this reason, Host-IP authentication is most effective when combined with User-Group authentication. If both methods of authentication are used, the end user will have to enter a username and password before getting access.

Access control files

When you use access control on your web server, the settings are stored in a file with the extension `.acl`. Access control files are stored in the directory *`server_root/server_type`*`acl` where *`server_type`* is the name of the server. For example, the administration server uses the directory `adminacl`. Netscape Enterprise Server uses `httpacl`.

The main ACL file name is `generated-https-`*`server-id`*`.acl`; the temporary working file is called `genwork-https-`*`server-id`*`.acl`. If you use the Server Manager forms to restrict access, you'll have these two files. However, if you want to do more complex restrictions, you can create multiple files and reference them from the `magnus.conf` file. There are also a few features available only by editing the files. For example, you can restrict access to the server depending on the time of day or day of the week.

You also manually create and edit `.acl` files if you want to customize access control. For example, you might want to use an Oracle or Informix database of users instead of an LDAP database. To do this type of customizing, you need to

use the access control API to program a hook into the server's access control structure. This API is written in C. For more information on the API, see the Netscape DevEdge Online site at `http://developer.netscape.com`.

For more information on access control files and their syntax, see the online help.

How does access control work?

You can control access to the entire server or to parts of the server (that is, directories, files, file types). When the server evaluates an incoming request, it determines access based on a hierarchy of rules called access-control entries (ACEs), and then it uses the matching entries to determine if the request is allowed or denied. Each ACE specifies whether or not the server should continue to the next ACE in the hierarchy. The collection of ACEs is called an access-control list (ACL).When a request comes in to the server, the server looks in `obj.conf` for a reference to an ACL, which is then used to determine access. By default, the server has one ACL file that contains multiple ACLs.

For example, suppose someone requests the following URL:

```
http://www.mozilla.com/my_stuff/web/presentation.html
```

The server would first check access control for the entire server. If the ACL for the entire server was set to continue, the server checks to see if there is an ACL for the file type (`*.html`). Then, it checks for an ACL for the directory, `my_stuff`. If one exists, it checks the ACE and then moves on to the next directory. The server continues traversing the path either until it reaches an ACL that says not to continue or until it reaches the final ACL for the requested URL (in this case, the file `presentation.html`).

To set up access control for this example using the Server Manager forms, you could create an ACL for the file only or for each resource leading to the file. That is, one for the entire server, one for the `my_stuff` directory, one for the `my_stuff/web` directory, and one for the file.

The following sample ACL file illustrates one way to control access to this resource.

```
# File automatically written
#
# You may edit this file by hand
#
version 3.0;

# This ACL allows everyone in the local database or LDAP directory
acl "agents";
authenticate (user,group) {
   prompt = "Enterprise Server";
};
deny (all)
    user = "anyone";
allow absolute (all)
    user = "all";

# This ACL denies all access to the my_stuff directory
acl "path=C:\Netscape\SuiteSpot\docs\my_stuff";
deny (all)
    user = "anyone";

# This ACL allows access to anyone in the user database
acl "path=C:\Netscape\SuiteSpot\docs\my_stuff\web";
allow (all)
    user = "anyone";

# This ACL allows access to the file to anyone in the "my_group" group
acl "path=C:\Netscape\SuiteSpot\docs\my_stuff\web\presentation.html";
allow (all)
    user = "anyone";
    group = "my_group"

# This is the default ACL and denies access to anyone
acl "default";
deny (all)
```

Restricting access

This section takes you through the process of restricting access to documents on your web site. The sections following this one describe in detail each option available when using access control. Keep in mind that most access-control rules use only a subset of the available options.

There is also a section of examples you can review on page 106.

To create an access-control rule:

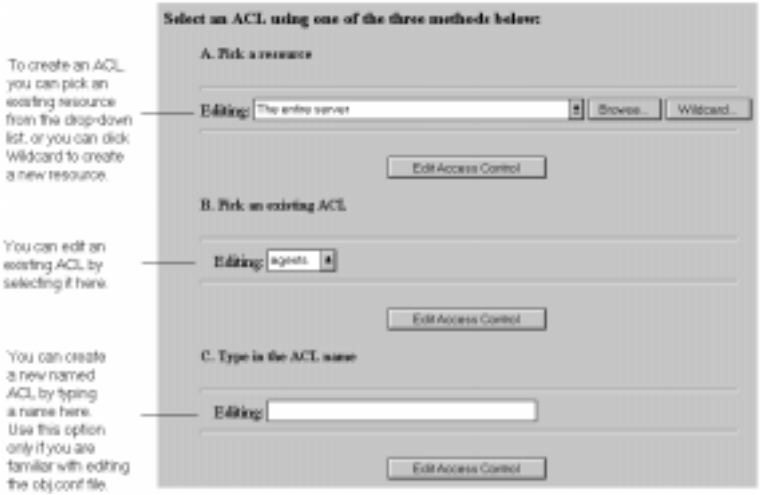
1. Go to the Server Manager and choose System Preferences|Restrict Access. A form appears where you select and edit an existing access-control rule or specify a new rule by either choosing the resource you want to apply to the rule (that is, the file, directory, or wildcard pattern you want to control) or typing a name to assign to the ACL. There are three sections to this main form:

   - **Pick a resource** lets you specify a wildcard pattern for files or directories to restrict access to (such as `*.html`), or you can choose a directory or a filename to restrict. You can also browse for a file or directory by using the Browse button.

   - **Pick an existing ACL** is a drop-down list of all the ACLs you've created.

   - **Type in the ACL name** is a field you can use to create named ACLs. Use this option only if you're familiar with ACL files and the `obj.conf` configuration file—you'll need to manually edit `obj.conf` if you want to apply named ACLs to resources.

Figure 6.2 The Restrict Access form has three sections.



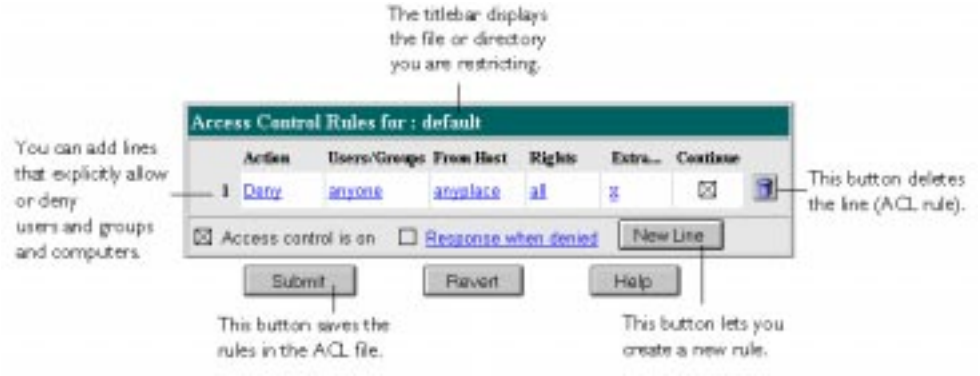
2. Specify the part of your web site (the resource) that you want to control. For example, you can select Entire Server to set up access control for your entire server.

   The following items are some common examples of resources you might use for access control.

Resource wildcard	What it means
Entire Server	One set of rules determines the access to your entire web site, including any virtual servers you have running. To restrict access to a virtual server, specify the path of its document root.
*.html	Controls access to all files with the `.html` extension
*.cgi	Controls access to all files with the `.cgi` extension
usr/ns-home/cgi-bin/*	Controls access to all files and directories in the cgi-bin directory. Note that the path is absolute. On NT, the path must include the drive letter.
agents	A named ACL that restricts access to all agents. The web server contains this ACL by default
uri="/sales"	Controls access to the `sales` directory in the document root. To specify URIs, create a named ACL.

3. Click the Edit Access Control button. The right frame divides into two frames that you use to set the access control rules. If the resource you chose already has access control, the rules will appear in the top frame. The following figure briefly describes the form elements.

Figure 6.3 The ACL form contains links that, when clicked, display another form in the bottom frame (not shown).



4. Click the New Line button. This adds a default ACL rule to the bottom row of the table. You can use the up and down arrows in the left column to move the rule, if needed.

5. Select the action you want to apply to the rule by clicking the Deny link. The bottom frame displays a form where you can check if you want to deny or allow access to the users, groups, or hosts you'll specify in the following steps. Check the option you want, and then click Update.

6. Specify User-Group authentication by clicking the "anyone" link listed under the Users/Groups column. The bottom frame displays a form for configuring User-Group authentication. By default, there is no authentication, meaning anyone can access the resource.

   Check the options you want, and then click Update.

7. Specify the computers you want to include in the rule by clicking the "anyplace" link. The bottom frame displays a form where you can enter wildcard patterns of host names or IP addresses to allow or deny.

   Check the options you want, and then click Update.

8. Specify the access rights you want to include in the rule by clicking the "all" link. Check the access rights in the bottom frame, and then click Update.

9. If you are familiar with ACL files, you can enter a customized ACL entry by clicking X under the Extra column. This area is useful if you use the access control API to customize ACLs.

10. Check Continue if you want the access-control rule to continue in a chain. This means the next line is evaluated before the server determines if the user is allowed access. When creating multiple lines in an access-control entry, it's best to work from the most general restrictions to the most specific ones.

11. Repeat steps 4 through 10 for each rule you need. If you want the user to be redirected to another URL if their request is denied, check "Response when denied." Click the link to specify the URL for redirection.

12. Click the Submit button to store the new access-control rules in the ACL file. If you click Revert, the server removes any changes you made to the rules from the time you first opened the two-frame window. Be cautious when using Revert because you can't restore your edits. In most cases, it's probably better to delete the rule lines individually.

The following sections describe the options that appear in the bottom frame of the access-control window.

Setting access-control actions

You can specify the action the server takes when a request matches the access-control rule.

- **Allow** means the people or computers can access the requested resource.
- **Deny** means the people or computers cannot access the resource.

The server goes through the list of ACEs to determine what the access is. For example, the first ACE is usually to deny everyone. If the first ACE is set to "continue," the server checks the second ACE in the list. (If continue is *not* checked, everyone would be denied access to the resource.) If the second entry matches, then the next ACE is used. The server continues down the list until it reaches either an ACE that doesn't match or that matches but is set to not continue. The last ACE that matches is used to determine if access is allowed or denied. For example, in Figure 6.4 any user in the database can view a file (read access), but they must be in the "pubs" group if they want to publish a file to the server.

Figure 6.4 You can combine Deny and Allow statements in an ACL.

Specifying users and groups

You can restrict access to your web site based on the user who requests a resource. With user and group authentication, users are prompted to enter a username and password before they can access the resource specified in the access-control rule.

The web server uses a list of users, who might be sorted into groups, to determine access rights for the user requesting a resource. The list of users (and the groups they are included in) are stored either in a database on the web server computer or in an LDAP server, such as Netscape Directory Server. You should make sure the database has users and groups in it before you set access control.

You can allow or deny access to everyone in the database, or you can allow or deny specific people by using wildcard patterns or lists of users or groups.

To configure access control with users and groups, follow the general directions for restricting access. When you click the Users/Groups field, a form appears in the bottom frame. The following list describes the options in the form.

- **Anyone (No Authentication)** is the default and means anyone can access the resource without having to enter a username or password. However, the user might be denied access based on other settings, such as host name or IP address.
- **Authenticated people only** means all users requesting the resource will have to enter a username and password before getting access. If the username they enter isn't in the database, the access-control rule won't

apply to them. However, if the rule says deny and then a group is listed, that group is denied, but everyone else in the database could be allowed depending on if there is another ACL that matches their request.

- **All in the authentication database** matches any user who has an entry in the database. To use this option, you must also check "Authenticated people only."
- **Only the following people** lets you specify certain users and groups to match. You can list the users and groups of users individually by separating the entries with commas. Or, you can enter a wildcard pattern. To use this option, you must also check "Authenticated people only."
    - **Group** matches all users in the groups you specify.
    - **User** matches the individual users you specify.
- **Prompt for authentication** lets you specify message text that appears in the authentication window. You can use this text to describe what the user needs to enter. Depending on the operating system, the user will see about the first 40 characters of the prompt. Netscape Navigator and Netscape Communicator cache the username and password and associate them with the prompt text. This means that if the user accesses areas (files and directories) of the server that have the same prompt, the user won't have to retype usernames and passwords. Conversely, if you want to force users to reauthenticate for various areas, you simply need to change the prompt for the ACL on that resource.
- **Authentication Methods** specifies the method the server uses when getting authentication information from the client.
    - **Default** uses the default method you specify in the `obj.conf` file, or "Basic" if there is no setting in `obj.conf`. If you check Default in this form, the ACL rule doesn't specify a method in the ACL file. Default is the best choice because you can easily change the methods for all ACLs by editing one line in the `obj.conf` file.
    - **Basic** uses the HTTP method to get authentication information from the client. The username and password are only encrypted if encryption is turned on for the server.

  - **SSL** uses the client certificate to authenticate the user. If you use this method, SSL must be turned on for the server. If you have encryption on, you can combine Basic and SSL methods.

  - **Other** uses a custom method you create using the access control API.

- **Authentication Database** lets you select a database that the server uses to authenticate users. The default setting means the server looks for users and groups in the either the local database or an LDAP directory, depending on the setting specified in the administration server. However, you can configure individual ACLs to use different databases. You can specify different databases and LDAP directories in the file *`server_root/`*`userdb/dbswitch.conf`. Then, you can reference the database you want to use in the ACL by typing the name of the database in the Other field in the User/Group form.

Specifying host names and IP addresses

You can restrict access to your web site based on which computer the request comes from. You specify this restriction by using wildcard patterns that match the computers' host names or IP addresses. For example, to allow or deny all computers in a specific domain, you would enter a wildcard pattern that matches all hosts from that domain, such as `*.netscape.com`.

To specify users from hostnames or IP addresses, follow the general directions for restricting access. When you click the From Host field (the link called "anyplace"), a form appears in the bottom frame. Check the "Only from" option and then type either a wildcard pattern or a comma-separated list of hostnames and IP addresses. Restricting by hostname is more flexible than by IP address—if a user's IP address changes, you won't have to update this list. Restricting by IP address, however, is more reliable—if a DNS lookup fails for a connected client, hostname restriction cannot be used.

The hostname and IP addresses should be specified with a wildcard pattern or a comma-separated list. The wildcard notations you can use are specialized; you can only use the `*`. Also, for the IP address, the `*` must replace an entire byte in the address. That is, `198.95.251.*` is acceptable, but `198.95.251.3*` is not. When the `*` appears in an IP address, it must be the right-most character. For example, `198.*` is acceptable, but `198.*.251.30` is not.

For hostnames, the * must also replace an entire component of the name. That is, `*.netscape.com` is acceptable, but `*sers.netscape.com` is not. When the * appears in a hostname, it must be the left-most character. For example, `*.netscape.com` is acceptable, but `users.*.com` is not.

Setting access rights

You can set access rights to files and directories on your web site. That is, in addition to allowing or denying all access rights, you can specify a rule that allows or denies partial access rights. For example, you can give people read-only access rights to your files, so they can view the information but not change the files. This is particularly useful when you use the web publishing feature to publish documents.

When you create an access-control rule, the default access rights are set to all access rights. To change access rights, click the Rights link in the top frame, and then check or uncheck the access rights you want to set for a particular rule. The following list describes each access right you can check.

- **Read** access lets a user view a file. This access right includes the HTTP methods GET, HEAD, POST, and INDEX.
- **Write** access lets a user change or delete a file. Write access right includes the HTTP methods PUT, DELETE, MKDIR, RMDIR, and MOVE.
- **Execute** access applies to server-side applications, such as CGI programs, Java applets, and agents.
- **Delete** access means a user can delete a file or directory.
- **List** access means the user can get directory information. That is, they can get a list of the files in that directory. This applies to Web Publisher and to directories that don't contain an `index.html` file.
- **Info** access means the user can get headers (`http_head` method). This is mainly used by the Web Publisher.

Writing customized expressions

You can enter custom expressions for an ACL. You can use this feature if you are familiar with the syntax and structure of ACL files. There are a few features available only by editing the ACL file or creating custom expressions. For example, you can restrict access to your server depending on the time of day, day of the week, or both.

The following customized expression shows how you could restrict access by time of day and day of the week. This example assumes you have two groups in your LDAP directory: the "regular" group gets access Monday through Friday, 8:00am to 5:00pm. The "critical" group gets access all the time.

```
allow (read)
{
   (group=regular and dayofweek="mon,tue,wed,thu,fri");
   (group=regular and (timeofday>=0800 and timeofday<=1700));
   (group=critical)
}
```

For more information on valid syntax and ACL files, see the online help.

When "Access control on" is checked

You can turn off access control for any part of the server a user accesses. For example, you could create an ACL that restricts access to the resource `*.html`, and then you could have an ACL for the entire server that is turned off. In this case, the only time access-control is used is when a user requests any file or directory with the .html extension.

When you uncheck this option, you'll get a prompt asking if you want to erase records in the ACL. When you click OK, the server deletes the ACL entry for that resource from the ACL file.

Responding when access is denied

You can choose the response a user sees when denied access. You can vary the message for each access-control object. By default, the user is sent a message that says the file wasn't found (the HTTP error code 404 Not Found is also sent).

To change what message is sent for a particular ACL:

1. In the ACL form, click the link called "Response when denied."
2. In the lower frame, check the radio button called "Respond with the following file."
3. In the text field, type a URL or URI to a text or HTML file in your server's document root that you want to send to users when they are denied access. Make sure the file doesn't contain references to other files (such as style sheets) or images because they won't be sent. Click Update.

Note Make sure any users who get the response file have access to that file. That is, if you have access control on the response file and the user is denied access to both the original resource and the response file, the server will send the default denied response.

4. Make sure you submit the access -control rule by clicking the Submit button in the top frame.

Examples

This section describes some common examples for restricting access to a web server and its contents.

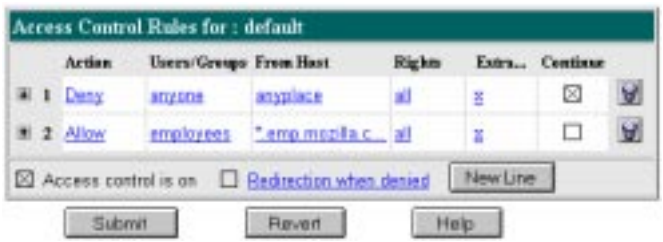
Restricting access to the entire server

This example allows access to users in a group called "employees" who access the server from computers in a subdomain. There are no access-control rules for other resources on the server. You might use this example if you have a server for a department and you only want users to access the server from computers in a specific subdomain of your network.

To restrict access to the entire server:

1. In the Server Manager, choose Server Preferences | Restrict Access.
2. In the section called Pick a Resource, select "The entire server" from the Editing drop-down list. Click Edit Access Control. The two-frame forms appear.
3. Click New Line. The default rule appears, which denies all access to the server. Typically, you should deny all access to your server, and then allow specific access to users, groups, and computers; however, you might change this if you want to deny access only to a small group of users or groups. Click New Line again to create a second rule.
4. Click the Deny link in the second rule. In the bottom form that appears, check Allow, and then click Update.
5. Click the "anyone" link in the second rule. In the bottom form, type the group you want to have access to the server. For this example, type `employees` in the Group field. Note that the two options called "Authenticated people only" and "Only the following people" are checked automatically. Click Update.
6. Click the "anyplace" link in the second rule. In the bottom form, type a wildcard pattern for the host names of the computers you want to allow. For example, type `*.emp.mozilla.com` in the Host Names field. Click Update.
7. Uncheck the Continue box in the top frame, and then click Submit. The form should look like the one in Figure 6.5. Save and apply your changes.

Figure 6.5 Restricting access to the entire server



Be sure to restart the server for the changes to take affect. The following text is the ACL file for this example.

```
# File automatically written
#
# You may edit this file by hand
#

version 3.0;

acl "default";
deny (all)
    user = "anyone";
allow absolute (all)
    user = "employees" and
    dns = "*.emp.mozilla.com";
```

Restricting access to a directory (path)

This example lets users in a group called “executives” have read access to a directory and its subdirectories and files on the server. The user called “ceo” has full permissions to the directory.

You might use this example if you have a directory on your server that one person owns (that is, they publish to this directory) and you want one group of users to read the files. For example, you might have a project owner who publishes status information for the project team to review.

To restrict access to a directory on the server:

1. In the Server Manager, choose Server Preferences|Restrict Access.
2. In the section called Pick a Resource, click the Browse button. In the form that appears, click the link for the directory you want to restrict. The directories listed in this form are in the servers document root. Once you click a link, the Editing drop-down list displays the absolute path to the directory.

Note If you want to view all files in your server root, click the Options button and check the box labeled “List files as well as directories” and then click OK.

3. Click Edit Access Control. The two-frame forms appear.

4. Click New Line twice to create two rules. Don't edit the default values for the first rule—they deny all access to the directory. You'll edit the second rule to allow read access to the "executives" group.

5. Click the Deny link in the second rule. In the bottom form that appears, check Allow, and then click Update.

6. Click the "anyone" link in the second rule. In the bottom form, type the group you want to have access to the server. For this example, type `executives` in the Group field. Click Update.

7. Click the "all" link in the top frame. Uncheck the Write and Delete access rights. This means the users in the executives group can't add or remove files, but they can view them and run any applications in the directories. Click Update.

8. Click New Line to create a rule for the "ceo" user. Check Allow for the third rule.

9. Click the "anyone" link. In the bottom form, type `ceo` in the User field. Click Update.

10. Uncheck Continue for both the second and the third rules. This means that the server ignores any ACLs for directories or files under the directory you specified in Step 2. The form should look like the one in Figure 6.6. Save and apply your changes.

Figure 6.6 Restricting access to a path in the server

The entry in the `generated.https-serverid.acl` file for this example looks like this:

```
acl "path=d:/netscape/suitespot/docs/senior-staff/";
deny (all)
    user = "anyone";
allow absolute (read,execute,list,info)
    group = "executives";
allow absolute (all)
    user = "ceo";
```

Restricting access to a URI (path)

This example uses a URI to control access to a single user's content on the web server. URIs are paths and files relative to the server's document root directory. Using URIs is an easy way to manage your server's content if you frequently rename or move all or part of it (for example, for disk space). It's also a good way to handle access control if you have additional document roots.

This example gives anyone read access to files and directories in the path specified by the URI `/my_directory`. Only one user ("me" in this example) has full access to the directories and files.

You might use this example if you have several users who publish their content on your server. The users want to have write access to their content, and they want anyone to have read/execute access.

To restrict access to a URI:

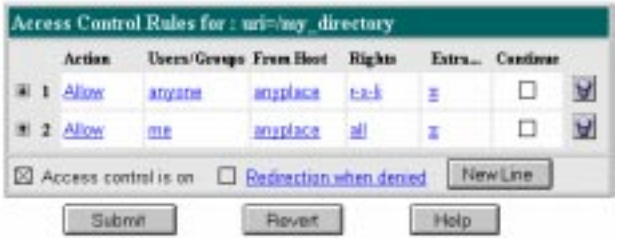
1. In the Server Manager, choose Server Preferences|Restrict Access.
2. In the section called "Type in the ACL name," type the URI you want to control. For example, type `uri=/my_directory`. Click Edit Access Control. The two-frame forms appear.
3. Click New Line to create the first rule that allows all users read access.
4. Click the Deny link. In the bottom form that appears, check Allow, and then click Update.
5. Click the "all" link in the top frame. Uncheck the Write and Delete access rights. This means users can't add or remove files, but they can view them and run any applications in the directories. Click Update.

6. Click New Line to create a rule for the owner of the directory. Check Allow for the second rule.

7. Click the "anyone" link. In the bottom form, type `me` in the User field. Click Update.

8. Uncheck Continue for both the first and second rules. This means that the server ignores any ACLs for other URIs, directories, or files under the URI you specified in Step 2. The form should look like the one in Figure 6.7. Save and apply your changes.

Figure 6.7 Restricting access to a URI (path) in the document root



The entry in the `generated.https-`*`serverid`*`.acl` file for this example looks like this:

```
acl "uri=/my_directory";
allow absolute (read,execute,list,info)
    user = "anyone";
allow absolute (all)
    user = "me";
```

Restricting access to a file type

This example controls write and delete access to all files with the extension `.cgi`. You might use this example if you only want specific users to create programs that run on your server. In this example, anyone can run the programs, but only users in the "programmers" group can create or delete them.

To restrict access to a file type:

1. In the Server Manager, choose Server Preferences|Restrict Access.
2. In the section called "Pick a resource," click the Wildcard button. In the prompt that appears, type `*.cgi` and click OK. This wildcard pattern matches any request that contains a file or directory with the `.cgi` extension.
3. Click Edit Access Control. The two-frame forms appear.
4. Click New Line to create the first rule that will allow all users read access.
5. Click the Deny link. In the bottom form that appears, check Allow, and then click Update.
6. Click the "all" link in the top frame. Uncheck the Write and Delete access rights. This means users can't add or remove files or directories with the `.cgi` extension. Click Update.
7. Click New Line to create a rule that allows write and delete access to the "programmers" group. Check Allow for the second rule.
8. Click the "anyone" link. In the bottom form, type `programmers` in the Group field. Click Update. The form should look like the one in Figure 6.8. Save and apply your changes.

Figure 6.8 Restricting access to a file type—in this case, to files with the .cgi extension

In this example, both continue boxes are checked. This means that if a request for a file comes in, the server will first look at the ACL for the file type, and then it will continue to look for another ACL that matches (for example, am ACL on the URI or the path). The server checks ACLs in this order:

1. Pathcheck functions in `obj.conf`. For example, these could be wildcard patterns for files or directories. The entry in the ACL file would appear as follows: `acl “*.cgi”;`
2. URIs. For example, a path relative to the document root. The entry in the ACL file would appear as follows: `acl “uri=/my_directory”;`
3. Pathnames. For example, an absolute path to a file or directory. The entry in the ACL file would appear as follows:
   `acl “path=d:\netscape\suitespot\docroot1\sales/”;`

The entry in the `generated.https-`*`serverid`*`.acl` file for this example looks like this:

```
acl "*.cgi";
allow (read,execute,list,info)
    user = "anyone";
allow (all)
    group = "programmers";
```

Restricting access based on time of day

This example restricts write and delete access to the server during working hours. You might use this example if you don’t want people publishing documents at times when people might be accessing the files. This example allows users to publish during the evening during the week (between 6:00pm and 6:00am, Monday through Friday) and all time during the weekend.

To restrict access based on time of the day and day of the week:

1. In the Server Manager, choose Server Preferences|Restrict Access. In the section called Pick a Resource, select “The entire server” from the Editing drop-down list. (You can select any resource.) Click Edit Access Control. The two-frame forms appear.
2. Click New Line.

3. Click the Deny link. In the bottom form that appears, check Allow, and then click Update.

4. Click the "all" link in the top frame. Uncheck the Write and Delete access rights. This means that if a user wants to add, update, or delete a file or directory, this rule won't apply and the server will search for another rule that matches. Click Update.

5. Click New Line to create a rule that restricts the write and delete methods. Check Allow for the second rule.

6. Click the X link to create a customized expression. In the bottom form, type the following lines:

   ```
   user = "anyone" and
   dayofweek = "sat,sun" or
   (timeofday >= 1800 and
   timeofday <= 600)
   ```

   You might want to select the entire text element and copy to memory—if there are errors, you'll have to reenter the text. Click Update. The top form will display "Unrecognized expressions" in the Users/Groups and From Host fields.

7. Click Submit. If you made any errors in the custom expression, you'll get a JavaScript alert. Correct any changes and click Submit again.

Restart your server for the changes to take affect.

Converting a 2.0 ACL file

If you have a 2.0 server installed when you install the 3.0 server, you'll have the option of converting your 2.0 ACL file to a 3.0 format. You can also convert them after installation. For example, if you have the access-control files on a different computer, you can copy them to 3.0 server and then convert them.

To convert 2.0 ACL files:

1. Choose System Settings|Convert 2.0 ACL File.
2. Be default, the server assumes you copied the 2.0 file to the directory *server_root*/`httpacl` and named the file `generated.https-`*`serverid`*`.acl`. If you have copied and renamed the file, check the Default radio button and then click OK. If you haven't copied and renamed the file, follow the rest of these steps.
3. Type the absolute path to the 2.0 access control file.
4. Type a file name you want to assign to the 3.0 file. If you don't specify a 3.0 file name, the 2.0 file is converted and saved with the default name `generated.https-`*`serverid`*`.acl`.

Warning If you don't specify a 3.0 path and filename, the conversion will overwrite the default 3.0 file if one exists.

5. Click OK.
6. If your 2.0 ACL files contain references to databases, convert the database entries to the 3.0 local database or LDAP directory. You might need to edit the userdb/dbswitch.conf file to reference the database files.
7. Restart the web server. (From the Server Manager, choose Server Preferences|On/Off, and then click Server On to restart the web server.)

Once the file is converted, you can manually edit it. Or if you converted to the default file `generated.https-`*`serverid`*`.acl`, you can use the Server Manager forms to edit the settings.

Chapter

7

Using encryption and SSL

Netscape servers use an encryption system called Secure Sockets Layer (SSL) to ensure privacy when communicating with other SSL-enabled products, such as Netscape Navigator and Netscape Communicator.

For a complete discussion of encryption and SSL, see *Managing Netscape Servers*.

What is encryption?

Encryption is the process of transforming information so it can't be decrypted or read by anyone except the intended recipient. This encrypted information is called *ciphertext*. It is the ciphertext that you send across the network. For example, suppose you have a financial report stored at your web site. If SSL is enabled on your server, your server encrypts the report and sends the ciphertext to a client, who then decrypts the ciphertext back into the financial report.

Decryption reverses the process, turning the ciphertext back into the original message. Only the recipient can decrypt the text because only the recipient has the *key*. In fact, both the encryption and the decryption processes require keys.

Client-server communication with SSL uses two keys. The encryption key is called the *public key*, and the decryption key is the *private key*. SSL uses the two keys as follows:

- A client and server exchange public keys.
- The client generates a symmetric encryption key that is used only for this transaction. This key is called a *session key*.
- The client encrypts the session key with the server's public key and sends it to the server.
- For the rest of that transaction, the client and server can use the same key for encryption and decryption.

Increasing server security

There are other security risks besides someone trying to break your encryption. The modern network faces risk from external and internal hackers, using a variety of tactics to gain access to your server and the information on it.

So in addition to enabling SSL on your server, you should take extra security precautions. The following list describes the most important things you can do to make your server more secure. For more information on server security see *Managing Netscape Servers*.

- Limit physical access. Keep the server machine in a locked room that only authorized people can enter. This prevents anyone from hacking the server machine itself.
- Limit administration access. If you plan on remotely configuring your server, be sure to use your administration server's access control to allow administration from a very small number of locations. You should also make the administrative connection a mandatory SSL connection.
- Choose good passwords. It's important to choose passwords that are difficult to guess and never to reveal them to anyone. Your most important passwords should not contain words from any language because numerous password-cracking programs exist that can run through millions of possible word combinations in seconds. Your important passwords also should be at least eight characters long, and contain a mix of uppercase and lowercase letters, punctuation marks, mathematical symbols, or numerals.

- Secure your private key. Make sure your private key file is protected. Store the key file in a directory that only you or authorized administrators have access to. It's also important to know whether the file is stored on backup tapes or is otherwise available for someone to intercept. If so, you must protect your backups as much as you protect your server.
- Limit other applications on the server. It's possible to circumvent your server's security by exploiting holes in other programs running on your server. Therefore, it is wise to disable all unnecessary programs and services. Some applications that you should be cautious of are telnet, rlogin, and rdist. Also be careful about which CGI, Java, and JavaScript programs are on your server.
- Know your server's limits. A server can't control the security of information once it reaches the client, nor can it control which individuals have access to directories and files on the server. Therefore, it is your responsibility to secure any information clients send to you through SSL.

Preventing clients from caching SSL files

You can prevent pre-encrypted files from being cached by a client by adding the following line inside the `<HEAD>` section of a file in HTML:

```
<meta http-equiv="pragma" content="no-cache">
```

Enabling SSL on your server

To enable SSL on your server, you must complete these steps:

1. Generate your server's key-pair file (public and private keys). For more information on key pairs, see *Managing Netscape Servers*.
2. Request a certificate from a Certification Authority (CA). For more information on Certification Authorities, see *Managing Netscape Servers*.
3. Install the certificate the CA sends to you.
4. Turn on SSL encryption for your server.

Note Steps 1 through 3 must be done through the administration server. For more information about using the administration server to enable SSL, see *Managing Netscape Servers*.

Activating SSL

After you have generated a key-pair file and installed your certificate, you can activate SSL for your server.

1. In the Server Manager, choose Server Preferences|Encryption On/Off. The Encryption On/Off form appears.
2. Check the On radio button.
3. Type the port number you want your server to use, if it's different from the one you specified upon setup. Note that the standard port is 443. If you choose another port, clients will have to specify it when they type your URL, as in `https://www.mozilla.com:80`.
4. From the alias drop-down list, select the name of the alias that you want to use for encryption. This alias maps to the key-pair file and certificate file you associated it with in the administration server.
5. Click OK. Save and apply your changes.

Note Most of the time, you want your server to run with SSL enabled. You might, at other times, want to disable it. If you temporarily disable SSL, make sure you re-enable it before processing transactions that require confidentiality, authentication, or data integrity.

Now that SSL is enabled on your server, you can configure your overall SSL preferences. See "Setting encryption preferences" on page 121 for information on configuring encryption preferences for your server.

Setting encryption preferences

You can set a number of system-wide preferences for SSL. To do so, choose Server Preferences|Encryption Preferences in the Server Manager. After you make your changes, click OK and confirm your changes. You can configure settings for SSL version, client certificates, and ciphers.

SSL version

You can specify which versions of SSL your server can communicate with. The latest and most secure version is SSL version 3, but many older clients use only SSL version 2. You will probably want to enable your server to use both versions.

Client certificates

You can configure the web server so that it refuses any client who doesn't have a client certificate from a trusted CA. This differs from access control in that all requests must be through SSL connections and they must be from clients who have certificates from trusted CAs. (See *Managing Netscape Servers* for details on configuring trusted CAs.)

With client certificates on, the server asks the client to send its certificate before the server will grant the request. The server doesn't care who the user is as long as that user has a valid certificate from a trusted CA. However, you can combine client certificates with access control so that in addition to being from a trusted CA, the user associated with the certificate must match the access-control rules. See Chapter 6, "Controlling access to your server," for more information.

To enable client authentication using trusted certificates:

1. Using the administration server, install a server certificate so that the web server can use SSL encryption, and then trust the CAs whose certificates you want to allow. The server will grant requests only to clients that have a certificate from a trusted CA.

2. Go to the Server Manager for the web server and choose Server Preferences|Encryption Preferences.

3. In the form that appears, choose Yes to require client certificates, and then click OK.

4. Turn on SSL for the web server. Save and apply your changes, and then restart the server.

If a user accessing the server doesn't have a certificate from a trusted CA, the server returns an error stating, "The server cannot verify your certificate." The server logs an error, but it doesn't list the untrusted CA. If you want to know which CA issued the certificate to the client, you need to use access control in addition to client authentication. With access control and certificate mapping, the server will log an error that lists the CA's distinguished name.

Ciphers

A *cipher* is an algorithm used in encryption. Some ciphers are more secure, or *stronger*, than others. Generally speaking, the more bits a cipher uses during encryption, the harder it is to decrypt the data. The list of available ciphers doesn't appear on the Encryption Preferences form unless you've enabled SSL.

When initiating an SSL connection with a server, a client lets the server know what ciphers it prefers for encrypting information. In any two-way encryption process, both parties must use the same ciphers. Because a number of ciphers are available, your server needs to be able to use the most popular ones.

You can choose ciphers from the SSL 2 protocol, as well as from SSL 3. To specify which ciphers your server can use, check them in the list. Unless you have a compelling reason not to use a specific cipher, you should check them all.

Another reason for not enabling all ciphers is to prevent SSL connections with less than optimal encryption. International versions of Netscape products are limited to 56-bit encryption keys. Therefore, international clients might be using only 56-bit or 40-bit encryption, which is not as difficult to crack as 128-bit. Unchecking all 56-bit ciphers effectively restricts access to browsers available only in the United States.

Warning! You might not want to check No Encryption, only MD5 message authentication. If no other ciphers are available on the client side, the server will use this, and no encryption will occur.

The SSL 2.0 ciphers are:

- RC4 cipher with 128-bit encryption and MD5 message authentication. RC4 ciphers are the fastest ciphers. This cipher, because it has 128-bit encryption, is the second strongest cipher next to Triple DES (Data Encryption Standard) with 168-bit encryption. It has approximately $3.4 * 10^{38}$ possible keys, making it very difficult to crack. As added security, all SSL 2.0 ciphers use MD5 (Message Digest 5) message authentication. MD5 message authentication detects attempts to modify data while it is in transit.

- RC4 cipher with 40-bit encryption and MD5 message authentication. This cipher is also an RC4 cipher, making it one of the fastest available ciphers. It has 40-bit encryption, which has approximately $1.1 * 10^{12}$ (a trillion) possible keys, making it easier to crack than encryption with more possible keys, such as 128-bit encryption. This cipher also uses MD5 message authentication to detect attempts to modify data in transit.

- RC2 cipher with 128-bit encryption and MD5 message authentication. The RC2 ciphers are slower than the RC4 ciphers. This RC2 cipher, because it has 128-bit encryption, is the second strongest cipher next to Triple DES with 168-bit. It has approximately $3.4 * 10^{38}$ possible keys, making it very difficult to crack. This cipher also uses MD5 message authentication to detect attempts to modify data in transit.

- RC2 cipher with 40-bit encryption and MD5 message authentication. This cipher is also an RC2 cipher, making it is slower than the RC4 cipher. It has 40-bit encryption, which is not as strong as 168-bit, 128-bit, or 56-bit encryption. 40-bit encryption has approximately $1.1 * 10^{12}$ (a trillion) possible keys. This cipher also uses MD5 message authentication to detect attempts to modify data in transit.

- DES with 56-bit encryption and MD5 message authentication. DES (Data Encryption Standard) is a U.S. government standard for data encryption. This cipher does not have as many possible keys as does 128-bit encryption, and therefore is not as strong. 56-bit encryption is the strongest encryption that Netscape Communications is permitted to export under U.S. law. 56-bit encryption has approximately $7.2 * 10^{16}$ possible keys. This cipher also uses MD5 message authentication to detect attempts to modify data in transit.

- Triple DES with 168-bit encryption and MD5 message authentication. Triple DES is the strongest cipher available, but it is not as fast as RC4. Triple DES uses a key three times as long as the key for standard DES. Because the key size is so large, there are more possible keys than for any other cipher - approximately $3.7 * 10^{50}$. This cipher also uses MD5 message authentication to detect attempts to modify data in transit.

The SSL 3.0 ciphers are:

- RC4 with 128-bit encryption and MD5 message authentication. This cipher is the same as the SSL 2.0 version of RC4 with 128-bit encryption but uses a more secure implementation of MD5 message authentication to detect attempts to modify data in transit.

- RC4 with 40-bit encryption and MD5 message authentication. This cipher is the same as the SSL 2.0 version of RC4 with 40-bit encryption but uses a more secure implementation of MD5 message authentication to detect attempts to modify data in transit.

- Triple DES with 168-bit encryption and SHA message authentication. This cipher is the same as the SSL 2.0 version of Triple DES with 168-bit encryption, but uses SHA (Secure Hash Algorithm) message authentication instead of MD5 message authentication. SHA is a government standardized algorithm that is used to construct a message authentication code that detects attempts to modify data while it is in transit. SHA is slower than MD5, but it is stronger.

- DES with 56-bit encryption and SHA message authentication. This cipher is the same as the SSL 2.0 version of DES with 56-bit encryption but uses SHA message authentication instead of MD5 message authentication.

- RC2 with 40-bit encryption and MD5 message authentication. This cipher is the same as the SSL 2.0 version of RC2 with 40-bit encryption but uses a more secure implementation of MD5 message authentication to detect attempts to modify data in transit

- No encryption, only MD5 message authentication. This cipher uses only MD5 message authentication to secure data. Any data sent using this cipher is not encrypted. The data may be protected from modification, but it can be viewed by eavesdroppers.

Specifying stronger encryption ciphers

You can specify cipher bit size for specific files or directories on your server. That is, for users to access a resource, they must have a browser that supports the specific cipher size you specify for that resource.

Note This feature is only available in the U.S. domestic version of Netscape Enterprise Server.

To specify a cipher for a resource:

1. In the Server Manager, choose Server Preferences|Stronger Ciphers. The Enforce Strong Security Requirements form appears.
2. Choose the resource to apply the stronger ciphers to. As discussed in the previous section, you might want to make any or all of your server accessible to only those clients who have strong encryption ciphers available.
3. Choose the minimum cipher size allowed to connect to the resource you picked.
4. Click OK and confirm your changes. Restart your server for the changes to take affect.

Effects of an SSL-enabled server

This section describes what effects you need to know about while running an SSL-enabled server.

Secure URL construction

URLs to an SSL-enabled server are constructed using https instead of simply http. URLs that point to documents on an SSL-enabled server have this format:

`https://servername.domain.dom/pathname/document`

Secure server document root

After SSL is installed and enabled on a server, all communications between the server and SSL-enabled browsers (such as Netscape Navigator) are private, authenticated, and checked for message integrity. Thus, any document sent to a user with an SSL-enabled browser is automatically encrypted.

Note Browsers not enabled with SSL can't communicate with an SSL-enabled server because they can't initiate an SSL transaction. However, they can communicate with the server when the server isn't using SSL.

Unprotected server document directory

If you want to have both protected and unprotected servers, you should operate the unprotected server on a different machine from the protected one. If your resources are limited and you must run an unprotected server on the same machine as your protected server, do the following.

- Assign proper port numbers. Make sure that the protected server and the unprotected server are assigned different port numbers. The registered default port numbers are 443 for the protected server and 80 for the unprotected one.

Changes to the magnus.conf file

Installing an SSL-enabled server creates new directive entries in the `magnus.conf` file (the server's main configuration file). These new directives are briefly described in the following sections.

Security

The Security directive tells the server whether SSL is enabled or disabled.

Syntax `Security` *`value`*

`value` specifies if SSL is on or off. Set the value parameter to `on` to enable SSL; and to `off` to disable SSL.

SSL2

The SSL2 directive tells the server whether SSL2 is enabled or disabled.

Syntax `SSL2 value`

value specifies whether SSL version 2 is enabled or disabled. Set the value parameter to `on` to enable SSL 2 and to `off` to disable SSL 2.

SSL3

The SSL3 directive tells the server whether SSL3 is enabled or disabled.

Syntax `SSL3 value`

value specifies whether SSL version 3 is enabled or disabled. Set the value parameter to `on` to enable SSL 3, and to `off` to disable SSL 3.

KeyFile

The KeyFile directive tells the server where the key file is located.

Syntax `KeyFile keyfile`

keyfile is the server's key file, specified as an absolute path from the server root.

CertFile

The CertFile directive specifies where the certificate file is located.

Syntax `CertFile certfile`

certfile is the server's certificate file, specified as an absolute path from the server root.

Ciphers

The Ciphers directive specifies the ciphers enabled for your server. For a discussion of these ciphers, refer to "Setting encryption preferences" on page 121.

Syntax `Ciphers +rc4 +rc4export -rc2 -rc2export +idea +des +desede3`

A + means the cipher is active, and a - means the cipher is inactive. Any cipher with `export` as part of its name is not stronger than 40 bits.

SSL3Ciphers

The SSL3Ciphers directive specifies which SSL 3 ciphers are enabled for your server. For a discussion of these ciphers, refer to "Setting encryption preferences" on page 121.

Syntax `SSL3Ciphers +rsa_rc4_128_md5 +rsa_3des_sha +rsa_des_sha +rsa_rc4_40_md5 +rsa_rc2_40_md5 -rsa_null_md5`

A + means the cipher is active, and a - means the cipher is inactive. Any cipher with `40` as part of its name is 40 bits.

SSLClientAuth

The SSLClientAuth directive specifies whether a client must have a certificate in order to communicate with the server. You don't need to turn on this directive to use client authentication with access control.

Syntax `SSLClientAuth` *`value`*

`value` specifies if certificates are always required. Set the value parameter to `on` to require certificates, and to `off` to specify that certificates are not required.

Chapter

8

Extending your server with programs

In addition to serving HTML documents, your server can run programs that interact with clients. These applications that run on the server computer are called *server-side applications*. (*Client-side applications*, which are downloaded to the client, run on the client machine.)

Your server can run these types of server-side applications:

- Java applets
- CGI programs
- JavaScript applications
- Plug-in programs that use the server plug-in APIs: Web Application Interface (WAI) and Netscape Server Plug-in (NSAPI)

This chapter describes how to install Java applets, CGI programs, and JavaScript applications onto your server. Plug-ins extend or replace your server's features. For example, you can use plug-ins to provide a different way to control access, or different logging mechanisms.

For information on writing and installing plug-ins, see the DevEdge site at `http://developer.netscape.com/library/documentation/`.

Additionally, your server can send these types of client-side applications to clients:

- Java applets
- JavaScript programs

This chapter deals mainly with the installation and configuration of server-side programs, although you'll find some information on client-side applications as well. For more detailed programming information, see the Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

Additionally, this chapter describes the steps for specifying a default query handler CGI program. (A query handler processes text sent to it via the `ISINDEX` tag in an HTML file.)

Installing server-side programs

Java applets, JavaScript applications, and CGI programs have different strengths and uses. Java is a full-featured programing language for creating network applications. CGI (*Common Gateway Interface*) programs can be written in C, Perl, or other programming languages; what makes them all CGI programs is the standard way they pass information between clients and servers. JavaScript applications are written in JavaScript, an object-based scripting language; it is easier to learn than an object-oriented programing language and lends itself to rapid application development.

Each type of program is installed onto the server differently. The following list summarizes the procedures:

- For Java applets, copy all your server-side applets into a specified directory and enable your server's Java interpreter.
- For CGI programs, configure your server to recognize certain files as CGI—all files with certain filename extensions, or all files in specified directories.
- For JavaScript applications, check in each application individually through the Application Manager, which you can access from the Server Manager or separately.

These installation procedures are described in the following sections.

Installing server-side Java applets

Java is a programming language designed for platform-independent application development. If you have a server-side Java applet that works on a Unix server, you can also use it with a server running on Windows NT. Like a CGI program, a server-side Java applet is triggered by a client sending or requesting information from a URL. However, server-side Java applets and client-side Java applets have different formats and are not interchangeable.

To use server-side Java applets with your server, you must enable your server's Java interpreter and copy all Java applets into a specified directory. All server-side Java applet files must be named in the standard format *name*.class. For more information on creating Java applets that work with your server, refer to Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

To enable your server to use server-side Java applets:

1. From the Server Manager, choose Programs|Java. The Java form appears.
2. Click the Yes button to enable the Java interpreter.
3. Specify a Java applet directory by typing it in the Java applet directory field.
4. Click the OK button.
5. Save and apply your changes.

Be sure to copy all server-side Java applets into the directory you've specified.

Installing CGI programs

Common Gateway Interface, or CGI, programs can be created with any number of programming languages. On a Unix machine, you're likely to find CGI programs written as Bourne shell or Perl scripts. On a Windows computer, you might find CGI programs written in C++ or batch files. CGI programs written in a Windows-based programming language such as Visual Basic use a different mechanism to operate with the server. They are called Windows CGI programs. See "Installing Windows CGI programs" on page 134 for information about Windows CGI.

Regardless of the programming language, all CGI programs accept and return data in the same manner, as described in Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

There are two ways to store CGI programs on your server machine:

- Specify a directory that contains only CGI programs. All files are run as programs regardless of the file extensions.
- Specify that CGI programs are all a certain file type. That is, they will all use the file extensions `.cgi`, `.exe`, or `.bat`. The programs can be located in any directory the server can serve from.

There are benefits to either implementation. If you want only a specific set of users to be able to add CGI programs, keep the CGI programs in specified directories and restrict access to those directories. If you want to allow anyone who can add HTML files to be able to add CGI programs, use the file type alternative. Users can keep their CGI files in the same directories as their HTML files.

If you choose the directory option, your server will attempt to interpret any file you place in that directory as a CGI program. By the same token, if you choose the file type option, your server will attempt to process any files with the file extensions `.cgi`, `.exe`, or `.bat` as CGI programs. If a file has one of these extensions but is not a CGI program, an error occurs when a user attempts to access it.

Specifying a CGI directory

To specify a CGI-only directory:

1. From the Server Manager, choose Programs|CGI Directory. The CGI Directory form appears.
2. In the URL Prefix field, type the URL prefix you want to use for this directory. That is, the text you type appears as the directory for the CGI programs in URLs.

   For example, if you type `cgi-bin` as the URL prefix, then all URLs to these CGI programs have the following structure:

   `http://`*`yourserver.domain.com`*`/cgi-bin/`*`program-name`*

Note The URL prefix you specify can be different from the real CGI directory you specify in the next step.

3. In the CGI Directory text field, type the location of the directory as an absolute path. Note that this directory doesn't have to be under your document root. This is the reason that you need to specify a URL prefix in the previous step.
4. Click OK.
5. Save and apply your changes.

To remove an existing CGI directory, click that directory's Remove button in the CGI Directory form. To change the URL prefix or CGI directory of an existing directory, click that directory's Edit button.

Copy your CGI programs into the directories you've specified. Remember—any files in those directories will be processed as a CGI file, so you don't want to put HTML files in your CGI directory.

Specifying CGI as a file type

To specify CGI programs as a file type:

1. From the Server Manager, choose Programs|CGI File Type. The CGI as a File Type form appears.
2. From the Resource Picker, choose the resource you want this change to apply to.
3. Click the Yes radio button.
4. Click OK.
5. Save and apply your changes.

The CGI files must have the file extensions `.bat`, `.exe`, or `.cgi`. Any non-CGI files with those extensions will be processed by your server as CGI files, causing errors.

Downloading executable files

If you're using `.exe` as a CGI file type, users will not be able to download `.exe` files as executables.

One solution to this problem is to compress the executable files that you want users to be able to download, so that the extension is not `.exe`. This solution has the added benefit of making the download time shorter.

Another possible solution is to remove `.exe` as a file extension from the `magnus-internal/cgi` type and add it instead to the `application/octet-stream` type (the MIME type for normal downloadable files). You can do this through the Server Manager, by choosing Server Preferences|MIME Types. However, the disadvantage to this method is that after making this change you cannot use `.exe` files as CGI programs.

Another solution is to edit your server's `obj.conf` file to set up a download directory, where any file in the directory is downloaded automatically. The rest of the server won't be affected. For directions on setting up this directory, see the technical note at `http://help.netscape.com/kb/server/960513-130.html`.

Installing Windows CGI programs

Windows CGI programs are handled much as other CGI programs. You specify a directory that contains only Windows CGI programs, or you specify that all Windows CGI programs have the same file extension. Note that like other CGI programs, you can use both methods at the same time if you want to. For example, you can create a directory for all your Windows CGI programs, and specify a Windows CGI file extension.

Although Windows CGI programs behave like regular CGI programs, your server processes the actual programs slightly differently. Therefore, you need to specify different directories for Windows CGI programs. If you enable the Windows CGI file type, it uses the file extension `.wcg`.

Netscape web servers support the Windows CGI 1.3a informal specification, with the following differences:

- The following keywords have been added to the [CGI] section to support Netscape security methods:

— HTTPS—Its value is on or off, depending on whether the transaction is conducted through SSL.

— HTTPS Keysize—When HTTPS is on, this value reports the number of bits in the session key used for encryption.

— HTTPS Secret Keysize—When HTTPS is on, this value reports the number of bits used to generate the server's private key.

- The keyword Document Root in the [CGI] section might not refer to the expected document root because the server does not have a single document root. The directory returned in this variable is the root directory for the Windows CGI program.
- The keyword Server Admin in the [CGI] section is not supported.
- The keyword Authentication Realm in the [CGI] section is not supported.
- Forms sent with multipart/form-data encoding are not supported.

Specifying a Windows CGI directory

To specify a Windows CGI-only directory:

1. From the Server Manager, choose Programs|Win CGI Directory. The WinCGI Directory form appears.
2. In the URL Prefix text field, type the URL prefix you want to use for this directory. That is, the text you type appears as the directory for the Windows CGI programs in URLs.

   For example, if you type `wcgi-programs` as the URL prefix, then all URLs to these Windows CGI programs have the following structure:

```
http://yourserver.domain.com/wcgi-programs/program-name
```

Note The URL prefix you specify can be different from the real Windows CGI directory you specify in Step 4.

3. Choose whether you want to enable script tracing.

   CGI parameters are passed from the server to Windows CGI programs through files, which the server normally deletes after the Windows CGI program finishes execution. If you enable script tracing, these files are

retained in a `/temp` directory or wherever the environment variables TMP and TEMP are pointing. Also, any window that the Windows CGI program brings up is shown when script tracing is enabled.

4. In the WinCGI Directory text field, type the location of the directory as an absolute path. Note that this directory doesn't have to be under your document root. This is the reason that you need to specify a URL prefix in Step 2.

5. Click OK.

6. Save and apply your changes.

To remove an existing Windows CGI directory, click that directory's Remove button in the Windows CGI Directory form. To change the URL prefix or Windows CGI directory of an existing directory, click that directory's Edit button.

Copy your Windows CGI programs into the directories you've specified. Remember—any file in those directories is processed as a Windows CGI file.

Specifying Windows CGI as a file type

To specify a file extension for Windows CGI files:

1. From the Server Manager, choose Server Preferences|MIME Types.

2. On the MIME Types page, add a new MIME type. Set the Type field to `type`, and the content type to `magnus-internal/wincgi`.

3. In the File Suffix field, enter the file suffixes that you want the server to associate with Windows CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can't use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.

4. Save and apply your changes.

Installing shell CGI programs

Shell CGI is a server configuration that lets you run CGI applications using the file associations set in Windows NT.

For information on setting Windows file extensions, see your Windows NT documentation.

For example, if the server gets a request for a shell CGI file called `hello.pl`, the server uses the Windows NT file associations to run the file using the program associated with the `.pl` extension. If the `.pl` extension is associated with the program `C:\bin\perl.exe`, the server attempts to execute the `hello.pl` file as follows:

```
c:\bin\perl.exe hello.pl
```

The easiest way to configure shell CGI is to create a directory in your server's document root that contains only shell CGI files. However, you can also configure the server to associate specific file extensions with shell CGI by editing MIME types from the Server Manager.

Specifying a shell CGI directory

To create a directory for your shell CGI files:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. Go to the Server Manager and click Programs|Shell CGI Directory. The Shell CGI form appears.
3. In the URL Prefix field, type the URL prefix you want to associate with your shell CGI directory. For example, suppose you store all shell CGI files in a directory called `C:\docs\programs\cgi\shell-cgi`, but you want users to see the directory as `http://www.`*`yourserver`*`.com/shell/`. In this case, you would type `shell` as the URL prefix.
4. In the Shell CGI Directory field, type the absolute path to the directory you created.

Warning! The server must have read and execute permissions to this directory. For Windows NT, the user account the server runs as (for example, `LocalSystem`) must have rights to read and execute programs in the shell CGI directory.

5. Make sure that any files in the shell CGI directory also have file associations set in Windows NT. The server returns an error if it attempts to run a file that has no file-extension association.

Specifying shell CGI as a file type

You can use the Server Manager's MIME Types page to associate a file extension with the shell CGI feature. This is different from creating an association in Windows NT.

To associate a file extension with the shell CGI feature in the server, for example, you can create an association for files with the `.pl` extension. When the server gets a request for a file with that extension, the server knows to treat the file as a shell CGI file by calling the executable associated in Windows NT with that file extension.

To associate a file extension as a shell CGI file:

1. Create the shell directory on your computer. This directory doesn't have to be a subdirectory of your document root directory.
2. From the Server Manager, choose Server Preferences|MIME Types.
3. On the MIME Types page, add a new MIME type. Set the Type field to `type`, and the content type to `magnus-internal/shellcgi`.
4. In the File Suffix field, enter the file suffixes that you want the server to associate with shell CGI. If you activated CGI, WinCGI, and shell CGI file types, you must specify a different suffix for each type of CGI. For example, you can't use the suffix `.exe` for both a CGI program and a shell CGI program. If you need to, you can edit the other MIME type fields on the page so that the suffixes are unique.
5. Save and apply your changes.

Using the query handler

You can specify a default query handler CGI program. A query handler processes text sent to it via the `ISINDEX` tag in an HTML file.

`ISINDEX` is similar to a form text field in that it creates a text field in the HTML page that can accept typed input. Unlike the information in a form text field, however, the information in the `ISINDEX` box is immediately submitted when the user presses Return. When you specify your default query handler, you tell your server to which program to direct the input. For an in-depth discussion of the `ISINDEX` tag, see an HTML reference manual.

To set a query handler, do the following:

1. Choose Programs|Query Handler.
2. Use the Resource Picker to select the resource you want to set a default query handler for. If you choose a directory, the query handler you specify runs only when the server receives a URL for that directory or any file in that directory.
3. In the Default Query Handler field, enter the full path for the CGI program you want to use as the default for the resource you chose.
4. Click OK.
5. Save and apply your changes.

Enabling WAI services

Web Application Interface (WAI) services are a kind of plug-in that uses the Common Object Request Broker Architecture (CORBA).

Before you use WAI services, you must enable them on your server. Enabling WAI services essentially turns on Internet Inter-ORB Protocol (IIOP) support in the server. You may have other (non-WAI) applications that need this support. If you need IIOP support, enable WAI services.

To enable WAI services on your server:

1. From the Server Manager, choose Programs|WAI Management.
2. The WAI Administration page appears. To enable WAI services, click the Yes radio button.
3. Save and apply your changes.

For more information about WAI, see *Writing Web Applications with WAI* on Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

Installing server-side JavaScript programs

To install server-side JavaScript programs, you need to activate server-side JavaScript for your server and use the Application Manager. This section includes information on accessing and using the Application Manager to install server-side JavaScript applications, as well as perform other functions.

For more information about writing JavaScript applications, see *Writing Server-Side JavaScript Applications* on Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

Activating server-side JavaScript

If you are using server-side JavaScript applications, you must first activate server-side JavaScript for your server.

To enable server-side JavaScript:

1. From the Server Manager, choose Programs|Server Side JavaScript.
2. Choose Yes under "Activate the Server Side JavaScript application environment."
3. If you want to require the administration server username and password before allowing access to the Application Manager, choose Yes. For more information, see "Securing the Application Manager" on page 143.
4. Click OK.
5. Save and apply your changes.

For applications written in server-side JavaScript, you can perform many administrative tasks with the server-side JavaScript Application Manager. Using the Application Manager, you can do the following:

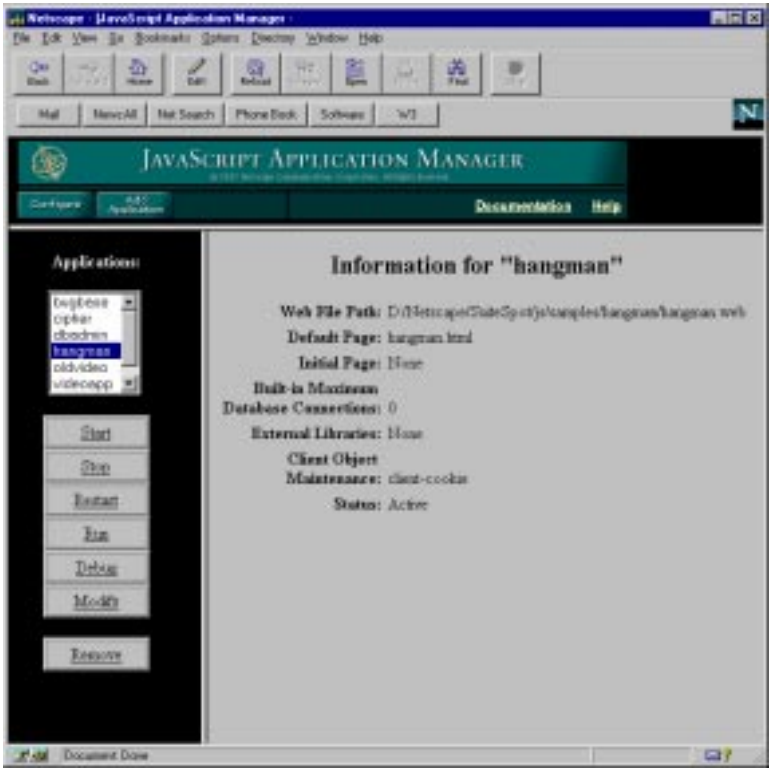
- Install a new JavaScript application. You must add an application before users can run it.
- Modify any of the attributes of an installed application (for example, its default home page, path to the `.web` file, and type of client-object maintenance).
- Stop, start, and restart an installed application.
- Run and debug an active application.
- Remove an installed application.

Running the Application Manager

To run the Application Manager from the Server Manager, choose Programs | Server Side JavaScript and click the link to the Application Manager. You can also run the Application Manager by loading the following URL in Navigator: `http://`*`server.domain`*`/appmgr`.

You see the following screen.

Figure 8.1 The Application Manager



The Application Manager displays all applications currently installed on the server in a scrolling list in the left frame. Select an application by clicking its name in the scrolling list.

For the selected application, the right frame displays

- The application name at the top of the frame.
- The path of the application `.web` file on the server.
- The default and initial pages for the application.
- The number of built-in database connections allowed.

- The external libraries used by the application (if any).
- The client object maintenance technique.
- The status of the application: active or stopped. Users can run only active applications. Stopped applications are not accessible.

Click the task buttons in the left frame to perform the indicated action on the selected application. For example, to modify the installation parameters of the selected application, click Modify.

Click Configure to configure the default settings for Application Manager. Click Add Application to add/install a new JavaScript application. Click Documentation for further documentation on server-side JavaScript. Click Help for instructions on using Application Manager.

Securing the Application Manager

Warning! Your Application Manager runs on your web server (rather than on the administration server). The Application Manager is installed into the `js\appmgr` directory. You can access it without the Server Manager with this URL: `http://`*`yourserver`*`.domain.com/appmgr`.

Consequently, you may want to restrict access to the Application Manager URL and the application URI so that only you and any other trusted administrators can access them. If you don't restrict access to the Application Manager, anyone can add, remove, modify, start, and stop applications on your server.

If you want to require the administration server user and password for access to the Application Manager, follow these steps:

1. From the Server Manager, choose Programs|Server Side JavaScript.
2. Under "Require administration server password for Server Side JavaScript Application Manager" choose the Yes radio button.
3. Save and apply your changes.

You then must enter the administration server username and password to use the Application Manager.

If your server does not use the Secure Sockets Layer (SSL), the username and password for the Application Manager are transmitted unencrypted over the network. Any intruder who intercepts this data may be able to access the Appli-

cation Manager. If you use the same password for your administration server, the intruder can also control your server. For security reasons, do not use the Application Manager from outside of your firewall unless you are using SSL.

Installing server-side JavaScript applications

You can install up to 120 JavaScript applications on one server.

You must install (add) an application with the Application Manager before you can run it. To install a new application:

1. From the Server Manager, choose Programs|Server Side JavaScript.
2. Click the link to the Application Manager.
3. Click Add Application from the top of the page. The Add Application page appears.
4. In the Name field, type the name of the JavaScript application. This name defines the application URL. For example, the name of the Hello World application is "world," and its application URL is `http://server.domain/world`. This is a required field, and the name you type must be different from all other application names on the server. The name must include only alphanumeric characters and cannot include spaces. For more information on application URLs, see "Application URLs" on page 145.
5. In the Web File Path field, type the absolute path to the `.web` file for the application. This is a required field.
6. In the Default Page field, note what file to send to a client who does not indicate a specific page for the application. This page is analogous to `index.html` for a standard URL. This is a required field.
7. In the Initial field, specify a page to run when the application is first started. This page only runs once during the life of the application and is used to initialize values and establish database connections. This is an optional field.
8. In the Built-in Maximum Database Connections field, specify the maximum number of database connections that this application can have at one time if you are using the built-in database object.

9. In the External Libraries field, specify the absolute paths of any libraries to be used with the application. This is an optional field. Libraries installed for one application can be used by all applications on the server.

10. In the Client Object Maintenance field, specify the mode for maintaining the client object. This can be client cookie, client URL, server IP, server cookie, or server URL.

11. After you have entered all the required information, click OK to install the application, Reset to clear all the fields, or Cancel to cancel the operation.

Note Don't give any JavaScript applications the same names as any subdirectories of your primary document directory. If you do, the server will no longer correctly process requests from the directory. For example, if you have a directory *server_root*`/docs/bug`, and a JavaScript application named `bug`, all requests for any files in the bug directory (or any of its subdirectories) will attempt to launch the JavaScript application `bug`. The JavaScript application URI takes precedence.

Application URLs

When you install a server-side JavaScript application, you must enter a name for it. This name determines the *application URL,* the URL that clients use to access a JavaScript application. Application URLs are of the form `http://`*server.domain/appName*`/page.html`.

where *server* is the name of the HTTP server, *domain* is the Internet domain (including any subdomains), *appName* is the application name you enter when you install it, and *page* is the name of a page in the application.

For example, if your server is named `myserver` and your domain name is `mozilla.com`, the application URL for the Hello World sample application is `http://`*myserver*`.mozilla.com/world/hello.html`.

When a client requests this URL, the server generates HTML for the specified page in the application and sends it to the client.

Important Before you install an application, make sure the application name you choose does not usurp an existing URL on your server. All client requests for URLs that match the application URL are routed to the directory specified for the `.web` file, circumventing the server's normal document root.

Using the previous example, any requests for URLs that begin with `http://myserver.mozilla.com/world` will look for documents in the `js\samples\world` directory and not in your server's normal document root.

Controlling access to a server-side JavaScript application

When you install an application, you may want to restrict its use to only certain users. You can do this by applying a configuration style to the application. For more information, see "Working with configuration styles" on page 64. For more information on restricting access to part of your server, see "Controlling access to your server" on page 91.

Modifying installation parameters

To modify an application's installation parameters, select the application name in the left frame of the Application Manager and click Modify.

You can change any of the parameters defined when you installed the application, except the application name. To change the name of an application, you must remove the application and then reinstall it.

If you modify the parameters of a stopped application, the Application Manager automatically starts it. When you modify parameters of an active application, Application Manager automatically stops and restarts it.

Removing a server-side JavaScript application

To remove the selected application, click Remove. This action removes the application from the Application Manager but does not delete files from the server. At this point, clients can no longer access the application.

If you delete an application, and you subsequently want to run it, you must install it again.

Starting, stopping, and restarting a server-side JavaScript application

To start an installed application that is stopped, click Start. If the application starts successfully, clients can run the application.

To stop an active application, click Stop. The application's status changes to "stopped," and clients can no longer run the application. You must stop an application if you want to move the `.web` file or update an application from a development server to a deployment server.

To restart a running application, click Restart. For any changes you have made to take effect, you must restart an application after you compile it.

You can also start, stop, and restart an application by entering a special URL of the form:

`http://`*`server.domain`*`/appmgr/control.html?name=`*`appName`*`&cmd=`*`action`*

where *appName* is the application name and *action* is either `stop`, `start`, or `restart`.

Running a server-side JavaScript application

There are two ways to run an installed application:

- Select the application name in the Application Manager, and click Run. A new Navigator window accesses the application.
- Type the application URL in Navigator.

If you attempt to run a stopped application (one that is not active), then the Application Manager tries to start it first.

Configuring default settings

Click Configure to configure default parameter settings for the Application Manager. When you install a new application, the default installation parameters are used for the initial settings.

You can specify the following default settings:

- Installation parameters: `.web` file path, default page, initial page, maximum number of built-in database connections, external libraries, and client object maintenance technique. You can specify a default directory path for your development area and native executables libraries.
- Whether you are prompted to confirm your action when you remove, start, stop, or restart an application.

- When debugging an application, whether the application trace appears in the same window as the application but in another frame, or in a window separate from the application.

Installing client-side programs

Installing client-side programs in your server is relatively easy. There are two types of client-side programs: Java applets and JavaScript programs. Client-side Java applets are executable files identified in an HTML document, retrieved from the server, and executed on the client. The applets can reside anywhere under your server's primary document root. Client-side JavaScript programs are embedded in HTML files and executed on the client.

Installing client-side Java applets

Instructions for creating client-side Java applets are outside the scope of this book. After you've created an applet, to install it you must copy it into a directory from which your server can deliver it. To send it to a client, it must be referenced in an HTML file.

Installing client-side JavaScript programs

Client-side JavaScript programs are created by lines of JavaScript code embedded in HTML files. The HTML files travel from the server to the client. Once the files reach the client, Navigator interprets the JavaScript code and performs the specified actions.

With LiveConnect you can connect server-side Java and JavaScript applications, or client-side Java and JavaScript applications. For more information on LiveConnect, on embedding JavaScript in HTML, and on using client-side JavaScript with other programs, refer to the documentation on the DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

Chapter

9

Monitoring the server

You can monitor your server's activity using several different methods. You can view the server's status in real time by using the Hypertext Transfer Protocol (HTTP) or the *Simple Network Management Protocol* (SNMP). You can also monitor your server by recording and viewing log files or by using the performance monitoring tools provided with your operating system.

Working with log files

Server log files record your server's activity. You can use these logs to monitor your server and to help you when troubleshooting. The `error` log file, located in `httpd-`*`servername`*`\logs` in the server root directory, lists all the errors the server has encountered. The access log, located in `httpd-`*`servername`*`\logs` in the server root directory, records information about requests to the server and the responses from the server. You can use the Server Manager to specify what to include in the `access` log file. Use the log analyzer to generate server statistics. You can back up server error and access log files by archiving them.

Viewing an access log file

You can view the server's active and archived access log files from the Server Manager.

To view an access log:

1. From the Server Manager, choose Server Status | View Access Log. The View Access Log form appears.
2. Choose the access log file you want to see. Active log files for resources and archived log files appear in the list.
3. To limit how much of the access log you see, type the number of lines you want to see in the "Number of entries" field. The order of the log entries on the screen is the order in which they were recorded in the log.
4. If you'd like to filter the access log entries for a particular word, type the word in the "Only show entries with" field. Case is important; make sure the case for your entry matches the case of the word you're searching for. (For example, if you want to see only those access log entries that contain "POST," type POST.)
5. Click OK.

Here is an example of an access log in the Common Logfile Format:

```
wiley.a.com - - [16/Feb/1996:21:18:26 -0800] "GET / HTTP/1.0" 200 751
wiley.a.com - - [17/Feb/1996:1:04:38 -0800] "GET /docs/grafx/icon.gif HTTP/1.0" 204 342
wiley.a.com - - [20/Feb/1996:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
arrow.a.com - john [29/Mar/1996:4:36:53 -0800] "GET /help HTTP/1.0" 401 571
```

Table 9.1 describes the last line of the sample access log.

Table 9.1 The fields in the last line of the sample access log file

Access Log Field	Example
Hostname or IP address of client	arrow.a.com. (In this case, the hostname is shown because the web server's setting for DNS lookups is enabled; if DNS lookups were disabled, the client's IP address would appear.
RFC 931 information	- (RFC 931 identity not implemented)
Username	john (username entered by the client for authentication)

Table 9.1 The fields in the last line of the sample access log file

Access Log Field	Example
Date/time of request	29/Mar/1996:4:36:53 -0800
Request	GET /help
Protocol	HTTP/1.0
Status code	401
Bytes transferred	571

Here is an example of an access log using the flexible logging format:

```
wiley.a.com - - [25/Mar/1996:12:55:26 -0800] "GET /index.htm HTTP/1.0" "GET" "/?-" "HTTP/
   1.0" 304 0 - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/1996:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-" "HTTP/1.0" 304 0
   - Mozilla/2.0 (WinNT; I)
wiley.a.com - - [25/Mar/1996:12:55:26 -0800] "GET / HTTP/1.0" "GET" "/?-" "HTTP/1.0" 304 0
   - Mozilla/2.0 (X11; I; IRIX 5.3 IP22)
```

The access log in the flexible logging format looks similar to the access log using the Common Logfile Format.

Viewing the error log file

The error log file contains errors the server has encountered since the log file was created; it also contains informational messages about the server, such as when the server was started. Incorrect user authentication is also recorded in the error log. Use the error log to find broken URL paths or missing files.

To view the error log file from the Server Manager:

1. From the Server Manager, choose Server Status|View Error Log. The View Error Log form appears.

2. If you want to see more or less than 25 lines of the error log, use the "Number of errors to view" field to enter the number of lines you'd like to see. The order of the log entries on the screen is the order in which they were recorded in the log.

3. If you'd like to filter the error messages for a particular word, type the word in the "Only show entries with" field. Case is important; make sure the case for your entry matches the case of the word you're searching for. (For example, if you want to see only those error messages that contain "warning," type `warning`.)

4. Click OK.

Here is an example of an error log:

```
[13/Feb/1996:16:56:51] info: successful server startup
[20/Mar/1996 19:08:52] warning: for host wiley.a.com trying to GET /report.html,
    append-trailer reports: error opening C:/Netscape/Server/docs/report.html
    (ERROR_FILE_NOT_FOUND)
[30/Mar/1996 15:05:43] security: for host arrow.a.com trying to GET /, basic-ncsa
    reports: user jane password did not match database C:/Netscape/Server/authdb/mktgdb
```

In this example, the first line is an informational message—the server started up successfully. The second log entry shows that the client `wiley.a.com` requested the file `report.html`, but the file wasn't in the primary document directory on the server. The third log entry shows that the password entered for the user `jane` was incorrect.

Setting log preferences

During installation, an access log file named `access` was created for the server. You can customize access logging for any resource by specifying whether to log accesses, what format to use for logging, and whether the server should spend time looking up the domain names of clients when they access a resource.

Server access logs can be in Common Logfile Format, flexible log format, or your own customizable format. The Common Logfile Format is a commonly supported format that provides a fixed amount of information about the server. The flexible log format allows you to choose (from the Server Manager) what to log. A customizable format uses parameter blocks that you specify to control what gets logged. Once an access log for a resource has been created, you can't change its format unless you archive it or create a new access log file for the resource.

To set access logging preferences:

1. From the Server Manager, choose Server Status|Log Preferences. The Log Preferences form appears.
2. Use the Resource Picker to choose the resource you'd like to apply custom logging to.
3. Select whether to log client accesses.
4. Type the full path for the log file.

   As a default, the log files are kept in the `logs` directory in the server root directory. If you specify a partial pathname, the server assumes the path is the `logs` directory in the server root.
5. Choose whether to record domain names or IP addresses in the access log.
6. Choose the format in which the log file should be: Common Logfile Format, or flexible log format (Only log radio button), or custom format. If you click Only log, you can choose any or all of the following flexible log format items:

   - Client hostname—The hostname (or IP address if DNS is disabled) of the client requesting access.
   - Authenticate username—If authentication was necessary, you can have the authenticated username listed in the access log.
   - System date—The date and time of the client request.
   - Full request—The exact request the client made.
   - Status—The status code the server returned to the client.
   - Content length—The content length, in bytes, of the document sent to the client.
   - HTTP header, "referer"—The referer specifies the page from which the client accessed the current page. For example, if a user was looking at the results from a text search query, the referer would be the page from which the user accessed the text search engine. Referers allow the server to create a list of backtracked links.

- HTTP header, "user-agent"— The user-agent information—which includes the type of browser the client is using, its version, and the operating system it's running on—comes from the User-agent field in the HTTP header information the client sends to the server.
- Method—The request method used.
- URI—Universal Resource Identifier. The location of a resource on the server. For example, for `http://www.a.com:8080/special/docs`, the URI is `special/docs`.
- Query string of the URI—Anything after the question mark in a URI. For example, for `http://www.a.com:8080/special/docs?find_this`, the query string of the URI is `find_this`.
- Protocol—The transport protocol and version used.

If you choose a custom format; type your custom format in the Custom format field. For more information about the parameters you should use, see Netscape's DevEdge online documentation web site at `http://developer.netscape.com/library/documentation/`.

7. If you don't want to log client access from certain hostnames or IP addresses, type them in the Hostnames and IP Addresses fields. Type a wildcard pattern of hosts the server should ignore when recording accesses. For example, use `*.netscape.com` if you don't want to log accesses from people whose domain is `netscape.com`. You can type wildcard patterns for hostnames, IP addresses, or both.
8. Choose whether to include the format string in the logfile. If you are using the proxy server's log analyzer, you should include a format string. If you are using a third-party analyzer, you may not want to include a format string in your log file.
9. Click OK.

Archiving log files

You can archive the access and error log files and have the server create new ones.

When you archive log files, the server renames the current log files and then creates new log files with the original names. You can back up or archive (or delete) the old log files, which are saved as the original filename followed by the date and time the file was rotated. For example, `access` might become `access.24Apr-04AM`.

You can archive log files immediately or have the server archive log files at a specific time on specific days. The information about when to archive log files is stored in the `cron.conf` file in the `admin-serv` directory in the server root directory; the server's cron configuration options are stored in `ns-cron.conf` in the `admin-serv` directory.

Note Before running the log analyzer, you should archive the server logs.

To archive log files:

1. From the Server Manager, choose Server Status|Archive Log. The Archive Log Files form appears.
2. Click Archive if you want to rotate the log files immediately.

   If you want archiving to occur at specific times on specific days, click the "Rotate log at" button, choose a time from the pull-down menu, and select the days for archiving to occur.
3. Click OK.

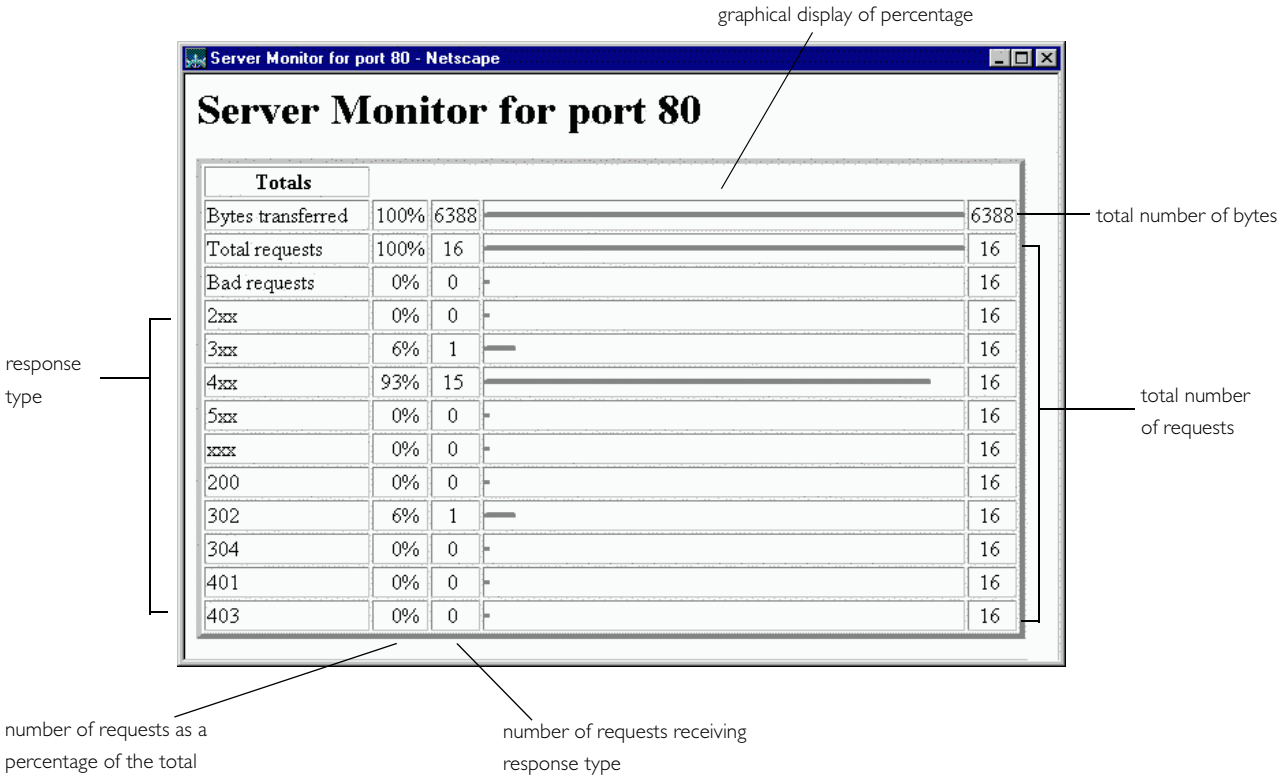
Monitoring the server using HTTP

You can monitor your server's usage with the interactive server monitor. You can see how many requests your server is handling and how it is handling these requests. If the interactive server monitor reports that the server is handling a great number of requests, you may need to adjust the server configuration or the system's network kernel to accommodate the requests. The interactive server monitor is shown in Figure 9.1.

To monitor your server from the Server Manager:

1. From the Server Manager, choose Server Status|Monitor Current Activity.
2. Click "Monitor current activity on port *port_number*". The interactive server monitor shown in Figure 9.1 appears.

Figure 9.1 The interactive server monitor



The interactive server monitor reports the totals for the following server values:

- Bytes transferred - the number of bytes the server is transferring
- Total requests - the number of requests the server is handling
- Bad requests - the number of bad requests the server is handling
- 2xx - the number of status codes ranging from 200 to 299 that the server is handling
- 3xx - the number of status codes ranging from 300 to 399 that the server is handling
- 4xx - the number of status codes ranging from 400 to 499 that the server is handling
- 5xx - the number of status codes of 500 and higher that the server is handling
- xxx - the total number of 2xx, 3xx, 4xx, and 5xx status codes the server is handling minus timeouts and other errors that did return an HTTP status code
- 200 - the number of successful transactions the server is processing
- 302 - the number of relocated URL status codes the server is processing
- 304 - the number of requests for which the server tells the client to use a local copy of a URL instead of retrieving a newer version from the server
- 401 - the number of unauthorized requests the server is handling
- 403 - the number of forbidden URL status codes the server is handling

Working with the log analyzer

Use the log analyzer to generate statistics about your server, such as a summary of activity, most commonly accessed URLs, times during the day when the server is accessed most frequently, and so on. You can run the log analyzer from the Server Manager or the command line.

Note Before running the log analyzer, you should archive the server logs. For more information about archiving server logs, see "Monitoring the server using SNMP" on page 162.

Running the log analyzer from the Server Manager

To run the log analyzer from the Server Manager:

1. From the Server Manager, choose Server Status|Generate Report. The Generate Report form appears.
2. Type the name of your server; this name appears in the generated report.
3. Choose whether the report will appear in HTML or plain text format.
4. Select the log file you want to analyze.
5. If you want to save the results in a file, type an output filename in the Output file field. If you leave the field blank, the analyzer prints results on the screen. For large log files, you should save the results to a file because printing the output to the screen might take a long time.
6. Select whether to generate totals for certain server statistics. You can generate the following totals:
   - Total hits—The total number of hits the server received since access logging was enabled.
   - 304 (Not Modified) status codes—The number of times the requesting client used a local copy of the requested document, rather than retrieving it from the server.
   - 302 (Redirects) status codes—The number of times the server redirected to a new URL because the original URL moved.
   - 404 (Not Found) status codes—The number of times the server couldn't find the requested document or the server didn't serve the document because the client was not an authorized user.

- 500 (Server Error) status codes—The number of times a server-related error occurred.
- Total unique URLs—The number of unique URLs accessed since access logging was enabled.
- Total unique hosts—The number of unique client hosts who have accessed the server since access logging was enabled.
- Total kilobytes transferred—The number of kilobytes the server transferred since access logging was enabled.

7. Select whether to generate general statistics. You can generate the following general statistics:

- Top number of one-second periods—You can generate the number of one-second periods during which requests were highest.
- Top number of one-minute periods—You can generate the number of one-minute periods during which requests were highest.
- Top number of one-hour periods—You can generate the number of one-hour periods during which requests were highest.
- Top number of users—You can generate the top number of users that accessed your server, provided that you included this as an item to log when you enabled access logging.
- Top number of referers—You can generate the number of referers that appear in your log analysis, provided that you included this as an item to log when you enabled access logging.
- Top number of user agents—You can generate the number of user agents that appear in your log analysis, provided that you included this as an item to log when you enabled access logging.
- Top number of miscellaneous logged items—You can generate the number of miscellaneous logged items that appear in your log analysis, provided that you included this as an item to log when you enabled access logging. These miscellaneous items include the request method, the URI, and the URI query.

8. Select whether to generate a list of server access statistics. You can generate a list of the following:

   - Most commonly accessed URLs—You can have the log analyzer show the most commonly accessed URLs or URLs that were accessed more than a specified number of times.
   - Hosts most often accessing your server—You can have the log analyzer show the hosts most often accessing your server or hosts that have accessed your server more than a specified number of times.

9. Specify the order in which you want to see the results.

10. Click OK.

Running the log analyzer from the command line

To analyze access log files from the command line, run the tool, `flexanlg`, which is in `extras/flexanlg` in your server root directory.

To run `flexanlg`, type the following command and options at the command prompt:

```
flexanlg [ -P ] [-n name] [-x] [-r] [-p order] [-i file]* [ -m metafile ]*
[ o file][ c opts] [-t opts] [-l opts]
```

The following describes the syntax. (You can get this information online by typing `flexanlg -h`.)

```
-P: proxy log format                                    Default: no
-n servername: The name of the server
-x : Output in HTML                                     Default: no
-r : Resolve IP addresses to hostnames                  Default: no
-p [c,t,l]: Output order (counts, time stats, lists)    Default: ctl
-i filename: Input log file(s)                          Default: none
-o filename: Output log file                            Default: stdout
-m filename: Meta file(s)                               Default: none
-c [h,n,r,f,e,u,o,k,c,z]: Count these item(s) -         Default: hnreuokc
    h: total hits
    n: 304 Not Modified status codes (Use Local Copy)
    r: 302 Found status codes (Redirects)
    f: 404 Not Found status codes (Document Not Found)
    e: 500 Server Error status codes (Misconfiguration)
    u: total unique URL's
    o: total unique hosts
    k: total kilobytes transferred
    c: total kilobytes saved by caches
    z: Do not count any items.
-t [sx,mx,hx, xx,z]: Find general stats - Default:s5m5h24x10
    s(number): Find top (number) seconds of log
    m(number): Find top (number) minutes of log
    h(number): Find top (number) hours of log
    u(number): Find top (number) users of log
    a(number): Find top (number) user agents of log
    r(number): Find top (number) referers of log
    x(number): Find top (number) for miscellaneous keywords
    z: Do not find any general stats.
-l [cx,hx]: Make a list of -                            Default: c+3h5
    c(x,+x): Most commonly accessed URLs
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    h(x,+x): Hosts (or IP addresses) most often accessing your server
             (x: Only list x entries)
             (+x: Only list if accessed more than x times)
    z: Do not make any lists
```

Monitoring the server using SNMP

You can monitor your server in real-time by using the *Simple Network Management Protocol* (SNMP). SNMP is a protocol used to exchange data about network activity. With SNMP, data travels between a managed device and a network management station (NMS) where users remotely manage the network.

A managed device is anything that runs SNMP (for example, hosts, or routers). Your Enterprise Server is a managed device. An NMS is usually a powerful workstation with one or more network management applications installed. A network management application graphically shows information about managed devices (which device is up or down, which and how many error messages were received, and so on).

Every managed device contains an SNMP *agent* that gathers information regarding the network activity of the device. This agent is known as the subagent. Each Netscape server (except the administration server) has a subagent.

Another SNMP agent exchanges information between the subagent and NMS. This agent is called the master agent. A master agent runs on the same host machine as the subagents it talks to. You can have multiple subagents installed on a host machine. All of these subagents can communicate with the master agent.

Values for various variables that can be queried are kept on the managed device and reported to the NMS as necessary. Each variable is known as a managed object, which is anything the agent can access and send to the NMS. All managed objects are defined in a management information base (MIB), which is a database with a tree-like hierarchy. The top level of the hierarchy contains the most general information about the network. Each branch underneath is more specific and deals with separate network areas.

How does SNMP work?

SNMP exchanges network information in the form of protocol data units (PDUs). PDUs contain information about various variables stored on the managed device. These variables, also known as managed objects, have values and titles that are reported to the NMS as necessary. Communication between an NMS and managed device can take place in one of two forms:

NMS-initiated communication: NMS-initiated communication is the most common type of communication between an NMS and a managed device. In this type of communication, the NMS either requests information from the managed device or changes the value of a variable stored on the managed device.

These are the steps that make up an NMS-initiated SNMP session:

1. The NMS searches the server's MIB to determine which managed devices and objects need to be monitored.

2. The NMS sends a PDU to the managed device's subagent through the master agent. This PDU either requests information from the managed device or tells the subagent to change the values for variables stored on the managed device.

3. The subagent for the managed device receives the PDU from the master agent.

4. If the PDU from the NMS is a request for information about variables, the subagent gives information to the master agent and the master agent sends it back to the NMS in the form of another PDU. The NMS then displays the information textually or graphically.

   If the PDU from the NMS requests that the subagent set variable values, the subagent sets these values.

Managed device-initiated communication: This type of communication occurs when the managed device needs to inform the NMS of an event that has occurred. A managed device such as a terminal would initiate communication with an NMS to inform the NMS of a shut down or start up. Communication initiated by a managed device is also known as a "trap."

These are the steps that make up a managed device-initiated SNMP session:

1. An event occurs on the managed device.
2. The subagent informs the master agent of the event.
3. The master agent sends a PDU to the NMS to inform the NMS of the event.
4. The NMS displays the information textually or graphically.

For information on setting up and configuring your server to use SNMP, see .

The Enterprise Server MIB

Each Netscape server has its own MIB (management information base). The Enterprise Server's MIB is a file called `netscape-http.mib`. This MIB contains the definitions for various variables pertaining to network management for the Enterprise Server. These variables are known as managed objects. Using the Enterprise Server MIB and network management software, such as HP OpenView, you can monitor your web server like all other devices on your network.

The Enterprise Server MIB has an object identifier of *netscape 1* (that is, `http OBJECT IDENTIFIER : := { netscape 1 }`) and is located in the *`server_root`*`/plugins/snmp` directory.

You can see administrative information about your web server and monitor the server in real time using the Enterprise Server MIB. Table 9.2 lists and describes the managed objects stored in the `netscape-http.mib`.

Table 9.2 `netscape-http.mib` managed objects and descriptions

Managed object	Description
httpEntityDescr	A description of the server (includes operating system information).
httpEntityId	The enterprise subtree for vendors (for example, Netscape's MIB has an object identifier of 1.3.6.1.4.1.1450).
httpEntityProtocol	The HTTP version number.
httpEntityVersion	The server software version number.
httpEntityOrganization	The organization responsible for the server.
httpEntityLocation	The full path for the server.

Table 9.2 `netscape-http.mib` managed objects and descriptions

Managed object	Description
httpEntityContact	The person(s) responsible for the server and contact information.
httpEntityAddress	The IP address of the machine the server is running on.
httpEntityPort	The port number on which the server is listening.
httpEntityName	The server's identifier name (for example, server2.a.com).
httpEntityType	The type of server.
httpEntityMethods	The methods supported by the server (for example, GET, POST, PUT).
httpEntityMaxProcess	The maximum number of active processes on the server.
httpEntityMinProcess	The minimum number of active processes on the server.
httpEntityMaxThread	The maximum number of active threads on the server.
httpEntityMinThread	The minimum number of active threads on the server.
httpStatisticsPort	The port number on which this server is listening.
httpStatisticsAddress	The IP address to which this server is bound.
httpStatisticsStatus	The status of the server (up or down).
httpStatisticsUptime	The uptime of the server since it was started.
httpStatisticsNumProcessIdle	The number of idle threads.
httpStatisticsNumProcessProc	The number of threads that are processing requests.
httpStatisticsNumProcessDns	The number of threads resolving host names.
httpStatisticsRequests	The total number of requests received and generated.
httpStatisticsRequestError	The total number of request errors detected.
httpStatisticsInUnknowns	The total number of unknown messages received/generated.
httpStatisticsInBytes	The total number of bytes received.
httpStatisticsOutBytes	The total number of bytes sent by the server.
httpStatisticsTimeOut	The total number of times the server has timed out.
httpStatisticsProcessNum	The number of running processes.
httpStatisticsThreadNum	The number of running threads.
httpStatisticsNumBytes	The total number of bytes sent by the server.

Table 9.2 `netscape-http.mib` managed objects and descriptions

Managed object	Description
httpStatisticsNum2xx	The number of 200-level status requests handled by the server.
httpStatisticsNum3xx	The number of 300-level status requests handled by the server.
httpStatisticsNum4xx	The number of 400-level status requests handled by the server.
httpStatisticsNum5xx	The number of 500-level status requests handled by the server.
httpStatisticsNum200	The number of 200 (Transfer OK) requests.
httpStatisticsNum302	The number of 302 (Moved Temporarily) requests.
httpStatisticsNum304	The number of 304 (Not Modified) requests.
httpStatisticsNum401	The number of 401 (Unauthorized) requests.
httpStatisticsNum403	The number of 403 (Forbidden) requests.

Enabling the subagent

Before you can monitor your server with SNMP, you need to enable the subagent that comes with your server. The subagent will then be able to communicate with the master agent built into the Windows NT operating system. You can enable the subagent via the Server Manager.

To enable the SNMP subagent,

1. From the Server Manager, choose Server Status|SNMP Subagent Configuration. The SNMP Configuration form appears.
2. Type the name of the system that has the master agent installed on it.
3. Type a description.
4. Type your organization name.
5. Type the web server's location.
6. Type the contact person for the web server.
7. Click the On radio button.
8. Click OK.

Using Performance Monitor

You can also monitor your server by using the Windows NT Performance Monitor, which graphically shows information about your computer's performance. Use Performance Monitor to see performance data about the Netscape Enterprise Server.

To monitor Netscape Enterprise Server performance using Performance Monitor:

1. Select the Performance Monitor icon in the Administrative Tools program group.
2. Choose Edit|Add to Chart. The Add to Chart window appears.
3. If the Netscape Enterprise Server you want to monitor is on a remote system, type its name in the Computer field.
4. Choose Netscape Server from the Object pull-down menu.
5. Choose the instance you want to see. (If you have multiple servers installed, you can choose multiple instances.)
6. Choose the counters you want to see in your chart. The following counters are available:
    - Server Conn/sec—Rate of incoming connections per second
    - Server Throughput (Kb/sec)—Rate of outgoing data from the server
    - Server Total Bytes—Total bytes sent by the server
    - Server Total Errors—Number of errored requests handled by the server
    - Server Total Requests—Total requests handled by the server
    - Status: 403 Forbidden—Number of "Forbidden" requests
    - Status: 200 level—Number of 200-level status requests handled by the server
    - Status: 200 OK—Number of "OK" requests

- Status: 300 level—Number of 300-level status requests handled by the server
- Status: 302 Moved Temporarily—Number of "Moved Temporarily" requests
- Status: 304 Not Modified—Number of "Not modified" requests
- Status: 400 level—Number of 400-level status requests handled by the server
- Status: 401 Unauthorized—Number of "Unauthorized" requests
- Status: 500 level—Number of 500-level status requests handled by the server

To see the counter definition online, click the Explain button.

7. Click Add.
8. To monitor other computers or objects, repeat steps 1 through 7 for each item you want to monitor.
9. Click Done. The Performance Monitor shows a chart with your selected items. A legend at the bottom of the window shows your choices.

For more information about Performance Monitor, see the documentation for your operating system.

Viewing events

In addition to logging errors to the server error log (see "Viewing the error log file" on page 151), the Netscape Enterprise Server logs severe system errors to the Event Viewer.

Using the Event Viewer

The Event Viewer lets you monitor events on your system. Use the Event Viewer to see errors resulting from fundamental configuration problems, which can occur before the error log can be opened.

To use the Event Viewer:

1. Select the Event Viewer icon in the Administrative Tools program group.
2. Choose Log|Application. The Application log appears in the Event Viewer.

   Errors from the administration server or Netscape Enterprise Server have a source label of `NetscapeAdministration` or `NetscapeHttps`.
3. Choose View|Find to search for one of these labels in the log. Choose View|Refresh to see updated log entries.

For more information about Event Viewer, see your system documentation.

Chapter

10

Using search

The Netscape Enterprise Server search function provides you with the ability to search the contents and attributes of documents on the server. As the server administrator, you can create a customized text search interface that's tailored to your user community.

Server documents can be in a variety of formats, such as HTML, Microsoft Word, Adobe PDF, and WordPerfect. The server converts many types of non-HTML documents into HTML as it indexes them so that users can use your web browser to view the documents that are found for their search.

Users can search through server documents for a specific word or attribute value, obtaining a set of search results that list all documents that match the query. They can then select a document from the list to browse it in its entirety. This provides easy access to server content.

As the server administrator, you can restrict which users and groups are authorized to use text search and which documents they can access, you can modify the configuration files that govern how text search operates, and you can customize the search query and results pages.

To enable searching capability on your server, you begin by identifying the special configuration needs of your server and using the several search configuration forms to input these. Then you need to identify the directory or directories of documents that you want prepared for searching and index the

document information into a searchable database, called a *collection*. The next several sections discuss the details of configuring search and indexing collections.

Configuring text search

You can configure several aspects of the search function for your specific server, some of which are collection-specific and others apply across all collections during a search. Collection-specific configuring affects how documents are indexed into a particular collection, so you must define these before creating the collection. Other configuring actions can be defined at any time because they only affect the searches themselves.

Collection-specific configuration actions:

- define URL mappings for the document directories to be indexed
- define the pattern files to display for searches on a particular collection

Configurations that affect all collections:

- establish access control for files and directories
- define any words you want dropped from the search
- define the search parameters
- turn the search function off and on
- restrict the amount of memory available for indexing operations

Controlling search access

The search function accesses the ACL database that is the default for your server. You can restrict access to the documents and directories on your server by defining explicit access control (ACL) rules or you can rely on the default access control definitions. You can add users to your server's access control database through the Administration Server's Users & Groups function. See Chapter 6, "Controlling access to your server" for more information about setting access control.

You can set your server to check access permissions before displaying search results (through the Agents & Search|Search Configuration form discussed in "Configuring the search parameters"). When this is set, before returning the results of a search query, the server checks a user's access privileges and challenges the user to identify themselves before displaying any results.

Mapping URLs

When users search through a collection's files, the documents that are returned as search results use a partial URL, called a *URI* (or *Uniform Resource Identifier*), to identify them. This is a security feature that prevents users from knowing the complete physical pathname for a file. A URI is set up by mapping a URL to an additional document directory.

For example, if the path for a file is *server_root*`\Docs\marketing\bizplans\planB.doc`, you could set up a mapping that prevents users from seeing all but the last directory by defining a URL prefix of `plans` and mapping it to *server_root*`\Docs\marketing\bizplans`. From then on, users need only type `\plans\planB.doc` to locate the file. For more information, see Chapter 4, "Managing server content."

The Enterprise server provides four default mappings:

- `/`—the primary document directory (sometimes called the *document root*), which initially maps to *server_root*`/docs`
- `/search-ui`—the directory for most of the search interface files
- `/webpub-ui`—the directory for most of the Web Publisher interface files
- `/publisher`—the directory for most of the Web Publisher files, which includes the online Web Publisher help files

When you create a collection, you must specify which document directory to index. You can only choose a directory that has a URL mapping or a subdirectory within such a mapped directory. You can create your own mappings to define specific directories. To do this, follow these steps:

1. From the Server Manager, choose Content Management.
2. Click the Additional Document Directories link.

3. Type in a nickname that maps the URL to the additional document directory you want to define. For example, type in the word `plans`.

4. Type the absolute physical path of the directory you want the URL mapping to map to. For example, `C:/Netscape/SuiteSpot/Docs/marketing/bizplans`.

5. If you want to apply a style to the directory, select the style in the Apply Style drop-down list. See Chapter 4, "Managing server content" for more information about styles.

6. Click OK to create the additional document directory.

Note Once you create a collection based on an additional document directory, you cannot change the URL mapping or the collection's entries will target the URL mapping to the wrong physical file location.

Deciding which words not to search

In the *server_root*`\plugins\search\common\english` directory, there is a file named `vdk20.stp`, which contains words the search engine does not index or search against in the English language. These are sometimes referred to as *stop words* or *drop words* and typically includes such words as *at*, *and*, *be*, *for*, and *the*.

For languages other than English, this file is in the directory for that language. This file applies to all collections in the chosen language. You can use a text editor to add or delete words you don't want the search engine to index or check against during a search.

Turning search on or off

You can turn search capabilities on and off for your server. Turning search off for a server where users do not use this function can improve server performance. You may also want to turn off the search function at certain times when you know the server will have heavy traffic, reserving this function for times when traffic is lighter.

If you turn it off, the search plug-in is not loaded when the HTTP server starts up. The default is for search to be turned on. To turn off the search function, use these steps:

1. From the Server Manager, choose Agents & Search.
2. Click the Search State link.
3. To turn the search function off, click the Off button.
4. Click OK to turn search off.

You can turn search back on with these steps:

1. From the Server Manager, choose Agents & Search.
2. Click the Search State link.
3. To turn the search function on, click the On button.
4. Click OK to turn search on.

Configuring the search parameters

As server administrator, you can set the default parameters that govern what users see when they get search results.

To configure search parameters:

1. From the Server Manager, choose Agents & Search.
2. Click the Search Configuration link.
3. Type the default maximum number of search result items displayed to users at a time. This cannot be larger than the value for the largest possible result set size, as defined in Step 4. The default is 20.
4. Type the maximum number of items in a result set. The default is 5000. For example, if you type 250 as the value, and there were 1000 documents that match the search criteria, users would only be able to see the first 250 or the 250 top-ranked documents (for searches that rank their results).

5. Type the format of the date/time string in Posix format. This is how the search results are displayed to users in the search results page. For example, the format %b-%d-%y %H:%M produces Oct-1-97 14:24. You can use the symbols listed in Table 10.1.

Table 10.1 Common Posix date and time formats

Format	Displayed result (example)
%a	Abbreviated week day (for example, Wed)
%A	Full week day (for example, Wednesday)
%b	Abbreviated month (for example, Oct)
%B	Full month (for example, October)
%c	Date and time formatted for current locale
%d	Day of the month as a decimal number (for example, 01-31)
%H	Hour as a decimal number, 24-hr military format (for example, 00-23)
%m	Month as a decimal number (for example, 01-12)
%M	Minute as a decimal number (for example, 00-59)
%x	Date
%X	Time
%y	Year without century (for example, 00-99)
%Y	Year with century (for example, 1999)

6. Type a default title for the document that is to be used if the document's author has not included a title as part of the document, tagged with the HTML `Title` tag. The typical default is (Untitled), which appears in the search results page for HTML files.

7. If you want the user's access permission to be checked before displaying the search results, click the Yes radio button under the label "Check access permissions before displaying search results?"

   If you click Yes, the server checks the user's access privileges for each file before displaying the documents found as a result of the search. Only the documents that you have permission to view are displayed. .

8. Click OK to set your new search configuration.

Configuring your pattern files

Pattern files are HTML files that define the layout of the text search interface. You can associate a pattern file with a search function and a set of pattern variables to create a specific portion of the interface. In the pattern file, you define the look, feel, and function of the text search interface. Pattern files use pattern variables that you can use to customize background color, help text, banners, and so on. In some cases, the values are pathnames to the files that contain the actual text and graphics that these variables represent; in other cases, the values represent text and HTML.

You can use the default pattern files, or you can create your own customized set of files and point to them from here. See "Customizing the search interface" for more information about how to change the user interface.

To define where the search function is to look for default pattern files associated with a particular search request, you have to specify the paths for the files.

1. From the Server Manager, choose Agents & Search.
2. Click the Search Pattern Files link.
3. Type the absolute path for the directory where you store your pattern files. The default start (header), end (footer), and query page pattern files are located in this directory.
4. Type in the relative pathname for the default pattern file you want to use for the top of the search results page when a collection has no defined header file or when more than one collection is being searched. Specify the path relative to the pattern file directory, as defined in Step 3.
5. Type in the relative pathname for the default pattern file you want to use for the footer of the search results page when for a collection has no defined footer file or when more than one collection is being searched. Specify the path relative to the pattern file directory, as defined in Step 3.
6. Type in the relative pathname for the pattern file you want to use for the search query page that appears when you start up the search function. Specify the path relative to the pattern file directory, as defined in Step 3.
7. Click OK to configure your search pattern files.

Configuring manually

The search function examines several configuration files to determine how search is configured on your server. These files define system settings, user-defined variables, and information about your search collections. You normally change this information through the Server Manager's Agents & Search forms, but you can also modify the files manually with your own text editor. Some of the implications of changing the configuration files in order to customize the user interface are discussed in "Customizing the search interface."

Note It is not recommended that you make any manual modifications to your configuration files, but if you do, you must restart the server for your modifications to take effect.

The configuration files

The configuration files that govern searching are:

- `webpub.conf`—This system configuration file contains system settings and file paths. In your server's `obj.conf` file, the search system initialization is mapped to the `webpub.conf` file. When you use the Search Configuration and Search Pattern Files forms, the data you input is reflected in the `webpub.conf` file. You can customize your server's search configuration by changing some of the settings in the `webpub.conf` file, but in general, you can make the changes you need through the Server Manager's forms.

- `userdefs.ini`—This user definitions file defines the user-defined pattern variables. In the `webpub.conf` file, this is mapped to the `userdefs.ini` file for your language (English, German, Japanese, and so on).

  You can customize a search interface by creating and defining your own pattern variables in the `userdefs.ini` file that can be used throughout your pattern files (See "User-defined pattern variables" for details).

- `dblist.ini`—This collection contents file describes collection-specific information. In the `webpub.conf` file, this is mapped to the `dblist.ini` file. When you create and maintain collections, the `dblist.ini` file is updated for you with information about your collections.

Adjusting the maximum number of attributes

Collections have different sets of default attributes that depend on which file format they are. For example, HTML files have `Title` and `SourceType`. You can also define META-tagged HTML attributes in your HTML files. Some file formats, such as PDF, have a great many default attributes. See "About collection attributes" and Table 10.2 for more information about the attributes for each format.

You can use the Add Custom Property form to add additional properties. These are the default maximum settings:

- Text (a maximum of 30, including all META-tagged attributes)
- Numeric (a maximum of 5)
- Date (a maximum of 5)

You can change the maximum settings for these in the `webpub.conf` file, although larger sets of attributes impact the performance of your server. You cannot set the maximums beyond 100 for text and 50 for dates and numbers.

To do this, you need to manually edit the `[NS-loader]` section of the `webpub.conf` file to define maximum numbers of attributes. For example, to change all three values, you could use these lines:

```
NS-max-text-attr = 50
NS-max-numeric-attr = 10
NS-max-date-attr = 10
```

Note You cannot use the additional attributes in existing collections, only in subsequently created collections. To use them in a search collection, you must use the Agents & Search|Maintain Collection form to remove the collection and then use the| New Collection form to create a new collection. If you want to use the new attributes in the web publishing collection, you must use your file system to remove both the `web_htm` and `link_mgr` collection files from the search collections directory and then restart your server.

Restricting memory for indexing

You can set a limit on the amount of RAM available for indexing operations. To do this, you need to manually edit the `[NS-loader]` section of the `webpub.conf` file to add a line defining a maximum memory amount. For example:

```
NS-max-memory = 32000000
```

The default is for the server to use all of the available memory that the system can offer. Most typically, you need to limit the RAM used for indexing in these two cases:

- The Enterprise Server 3.0 is installed on a machine that has less than the suggested minimum RAM requirement, 32MB.
- For server administrators on Windows NT servers that require a great deal of indexing but who wish to set aside some memory for other server operations.

Restricting your index file size

You can limit how much disk space an index file can consume. To do this, you need to manually edit the `[NS-loader]` section of the `webpub.conf` file to define a maximum index file size. For example,

```
NS-max-idx-file-size = 1500000
```

Typically, an indexing operation requires approximately 1.5MB per file, and since there are two files, one of which is temporary, you may need as much as 3MB of disk space for indexing. Setting the file size to 1.5MB per file puts a cap on how large each file can become.

Indexing your documents

Before users can execute searches, they need a database of searchable data against which they can target their searches. To do this, you create a database, called a *collection,* that indexes and stores information about the documents such as their content and file properties.

Searches require collections of files upon which to perform their searches. Once the documents are indexed, their contents and file properties, such as their titles, creation dates, and authors, are available for searching.

You can add or delete documents from a collection: optimizing, updating, and managing your collections as needed.

About collections

When your server administrator indexes all or some of a server's documents, information about the documents is stored in a collection. Collections contain such information as the format of the documents, the language they are in, their searchable attributes, the number of documents in the collection, the collection's status, and a brief description of the collection. For more details, see the section "Displaying collection contents."

When you create a collection, you indicate the type of files that it contains: HTML, ASCII, news, email, PDF, or multiple formats. This determines what happens during indexing: which attributes are indexed and what, if any, file conversion has to be done. Files in multiformat collections are converted to HTML and PDF files are converted to ASCII. You can index all the files in a directory or only those with a specific extension—for example, all the HTML, PDF, or `*.doc` documents.

A collection has records with information about each document that has been indexed. If the document is deleted from the collection, only the collection's entry for that document is removed. The original document is not deleted.

About collection attributes

Server documents can be in a variety of formats, such as HTML, Microsoft Excel, Adobe PDF, and WordPerfect. If there is a conversion filter available for a particular file format, the server converts the documents into HTML as it indexes them so that you can use your web browser to view the documents that are found for your search.

There are conversion filters for documents in these formats:

- HTML
- ASCII
- MS Rich Text Format (RTF)
- Interleaf 5.2-6.0
- MS Word (DOS) 3.0-6.0
- MS Word (Macintosh) 3-6
- MS Word (Windows) 2.0, 6.0, 7.0
- MS Excel 2-5
- MS Excel (Macintosh) 3-4
- MS PowerPoint 7.0
- Adobe PDF to ASCII
- Adobe FrameMaker (MIF) 3.0-5.0
- Ami Pro 1.x-3.1
- WordPerfect (Macintosh) 2-3.5
- WordPerfect (Windows) 5.x-6.1
- news and mail file formats

Certain file formats have a default set of attributes that are indexed for files of that type, as shown in Table 10.2.

Table 10.2 The default attributes indexed for each file format

File format	Attribute	Type	Description
ASCII	(none)	-	-
HTML	`Title`	text	The user-defined title of the file.
	`SourceType`	text	The original format of the document.
NEWS	`From`	text	The source userID of the news item.

Table 10.2 The default attributes indexed for each file format

	Subject	text	The text from the subject field of the news item.
	Keywords	text	Any keywords defined for the news item
	Date	date	The date the news item was created.
EMAIL	From	text	The source userID of the email.
	To	text	The destination userID of the email.
	Subject	text	The text from the email's subject field.
	Date	date	The date the email was created.
PDF	InstanceID	text	An internal ID number.
	PermanentID	text	An internal ID number.
	NumPages	integer	The number of pages in the document.
	DirID	text	The directory where the PDF file exists.
	FTS_ModificationDate	date	The document's last modification date.
	FTS_CreationDate	date	The document's creation date.
	WXEVersion	integer	The version of Adobe Word Finder used to extract the text from the PDF document.
	FileName	text	The Adobe filename specification.
	FTS_Title	text	The document's title.
	FTS_Subject	text	The document's subject.
	FTS_Author	text	The document's author.
	FTS_Creator	text	The document's creator.
	FTS_Producer	text	The document's producer.
	FTS_Keywords	text	The document's keywords.
	PageMap	text	The page map, describing the word instances for the page.

By default, HTML collections have `Title` and `SourceType` attributes, but they can be indexed to permit searching and sorting by up to 30 file attributes tagged with the HTML `<META>` tag. You can change the maximum settings for file attributes in `webpub.conf`, as discussed in "Adjusting the maximum number of attributes."

For example, a document could have these lines of HTML code:

```
<META NAME="Writer" CONTENT="J. S. Smith">
<META NAME="PubDate" CONTENT="07-24-97">
<META NAME="Product" CONTENT="Communicator">
```

If this document was indexed with its META tags extracted, you could search it for specific values in the writer, publication date, or product fields. For example, you could enter this query: `Writer <contains> Smith` or `PubDate > 1/1/97`.

Note Any attribute values in META-tagged fields are text strings only, which means that dates and numbers are sorted as text, not as dates or numbers. Also, illegal HTML characters in a META-tagged attribute are replaced with a hyphen.

Creating a new collection

You can create a collection that indexes the content of all or some of the files in a directory. You can define collections that contain only one kind of file or you can create a collection of documents in various formats that are automatically converted to HTML during indexing. When you define a multiple format collection (with the auto-convert option), the indexer first converts the documents into HTML and then indexes the contents of the HTML documents. The converted HTML documents are put into the `html_doc` directory in the server's search collections folder.

Note You need to have at least 3MB of available disk space on your system to create a collection. For information on how you can restrict the size of the index files, see "Restricting your index file size."

To create a new collection, follow these steps:

1. From the Server Manager, choose Agents & Search.
2. Click the New Collection link.
3. The Directory to Index field displays the currently defined document directory and provides a drop-down list of all the additional document directories defined for the server. See "Mapping URLs" for more information about additional document directories. You can select any of the items in the drop-down list as a starting point for finding the directory you want to index.

If you want to index a different subdirectory, click the View button to see a list of resources. You can index any directory that is listed or you can view the subdirectories in a listed directory and index one of those instead.

Once you click the index link for a directory, you return to the Create Collection form and the directory name appears in the Directory to Index field.

4. You can index all files in the chosen directory by leaving the default `*.html` pattern in the "Documents matching" field or you can define your own wildcard expression to restrict indexing to documents that match that pattern. For example, you could enter `*.html` to only index the content in documents with the `.html` extension, or you could use either of these patterns (complete with parentheses) to index all HTML documents:

   ```
   (*.htm|*.html)
   ```
   or
   ```
   *(.htm|.html)
   ```

   You can define multiple wildcards in an expression. See Chapter 3, "Managing your server" for details of the syntax for wildcard patterns.

5. To also index the subdirectories within the specified directory, click the "Include Subdirectories" checkbox.

6. In the Collection Name field, type a name for your collection. The collection name is used for collection maintenance. This is the physical file name for the file, so follow the standard directory-naming conventions for your operating system. You can use any characters up to a maximum of 128 characters. Spaces are converted to underscores.

7. In the optional Collection Label field, type a user-defined name for your collection. This is what users see when they use the text search interface. Make your collection's label as descriptive and relevant as possible. You can use any characters except single or double quotation marks, up to a maximum of 128 characters.

   Using single or double quotation marks prevents agent services from operating. If you know that you are not going to use agent services, you can use these quotation marks, but it is good practice to avoid using them.

8. In the optional Description field, type a description for your collection up to a maximum of 1024 characters.

9. Select the type of files the collection is to contain: ASCII, HTML, news, email, PDF, or multiple document formats. The kind of file format you choose indicates which default attributes are used in the collection and which, if any, automatic HTML conversion of the content is done as part of indexing. See "About collection attributes" and Table 10.2 for information about the attributes for each format.

   If you choose HTML as the file type and also try to index non-HTML files, the server creates the collection with the HTML set of default attributes and does not attempt to convert any non-HTML file it indexes. If you index HTML files into an ASCII collection, even the HTML markup tags are indexed as part of the file's contents and when you display the files, the contents are displayed as raw text. Regardless of the file type chosen, the content of the file is always indexed.

10. Select whether or not to extract META-tagged attributes from HTML files during indexing. If you extract these attributes, you can search on their values. You can index on a maximum of 30 different user-defined META tags in a document. You cannot use this option for multiple-format collections.

11. Select the collection's language from the drop-down list. The default is English, labeled "English (ISO-8859-1)." For more information on character sets, see Chapter 4, "Managing server content."

12. Click OK to create a new collection.

Configuring an existing collection

After you have initially created a collection, you can modify some of the initial settings for the collection. This data resides in the collection information file, `dblist.ini`, and when you reconfigure a collection, the `dblist.ini` file is updated to reflect your changes. See "Configuring manually" for more information about the configuration files. You can revise the description, change its label, define a different URL for its documents, and define how to indicate highlighting in displayed documents, which pattern files to use, and how to format dates.

Note This form allows you to modify some of the settings for the web publishing default collection, `web_htm,` because you are not changing actual collection data. Avoid making unnecessary making changes to this collection's settings.

To configure a collection, follow these steps:

1. From the Server Manager, choose Agents & Search.
2. Click the Configure Collection link.
3. In the optional Description field, you can type a description for your collection up to a maximum of 1024 characters.
4. In the optional Collection Label field, you can type a user-defined name for your collection. This is what users see when they use the text search interface. Make your collection's label as descriptive and relevant as possible. You can use any characters except single or double quotation marks, up to a maximum of 128 characters.

   Using single or double quotation marks prevents agent services from operating. If you know that you are not going to use agent services, you can use these quotation marks, but it is good practice to avoid using them.
5. In the URL for Documents field, you can type in the new URL mapping for the collection's documents if that has changed. That is, if you originally indexed the directory of files that corresponded to those defined by the URL mapping `/publisher/help`, and you have changed that mapping to the simpler `/helpFiles`, you would replace the URL of `/publisher/help` with the `/helpFiles` in this field. See "Mapping URLs" for more information about additional document directories.
6. In the Highlight Begin and Highlight End fields, you can type in the HTML tagging you want the server to use when highlighting a search query word or phrase in a displayed document. The default is to use bold, with the `<b>` and `</b>` tags, but you can add to this or change it. For example, you could add `<blink><FONT COLOR = #FF0000>` and the corresponding `</blink></FONT>` to highlight with blinking bold red text.
7. You can define different default pattern files for displaying the search results: how the search result's header, footer, and list entry line are formatted, respectively. Initially, the pattern files are in the *`server_root`*`\plugins\search\ui\text`.

8. In the Result Pattern File field, you can enter the name of the pattern file you want to use when displaying a single highlighted document from the list of search results.

9. In the Date Format field, you can specify how you want input dates to be interpreted when using this collection: MM/DD/YY, DD/MM/YY, or YY/MM/DD.

10. Click OK to change the collection configuration.

Updating an existing collection

After you have initially created a collection, you may want to add or remove files. If you are adding documents, the files' contents are indexed (and converted if necessary), when their entries are added to the collection. If you are removing documents, the entries for the files are removed from the collection along with their metadata. This function does not affect the original documents, only their entries in the collection.

Note If you selected the Extract Metatags option when you created this collection, then the META-tagged HTML attributes are indexed whenever you add new documents to this collection.

To update a collection, follow these steps:

1. From the Server Manager, choose Agents & Search.

2. Click the Update Collection link.

3. Select the collection you want to update from the drop-down list.

   The scrollable list of documents in the center of the form shows you what documents have index entries in the currently selected collection. The list holds 100 records, and the Prev and Next buttons get the previous (or next) set of 100 files for collections that have more than 100 files in them.

4. In the Documents Matching field, you can type in a single filename or you can use wildcards to specify the type of files you want added to or removed from the collection. If you enter a wildcard such as `*.html`, only files with this extension are affected. You can indicate files within a subdirectory by

typing in the pathname as it appears in the list of files. For example, you could delete all the HTML files in the `/frenchDocs` directory by typing in (no slash before the directory name): `frenchDocs/*.html`

Note: Be careful how you construct wildcard expressions. For example, if you type in `index.html`, you can add or remove the index file from the current collection. If instead you type in the expression `*/index.html`, you can add or remove all `index.html` files in the collection.

5. Select whether to index and add all matching documents from the subdirectories of the document directory that was originally defined for the collection. That is, if the collection originally indexed the `/publisher` directory, this option looks for documents matching the new pattern within all the subdirectories within `/publisher`. This does not apply for removing documents.

6. Click AddDocs to add the indicated files and subdirectories.

7. Click RemoveDocs to remove the indicated files.

Maintaining an existing collection

Periodically, you may want to maintain your collections. With normal usage, these tasks may not be necessary, but if you do a great deal of indexing and updating of collections, you may want to use some of these functions occasionally. You can perform the following collection management tasks:

- Optimize collections—You can optimize a collection to improve performance if you frequently add, delete, or update documents or directories in your collections. An analogy is defragmenting your hard drive. Optimizing is done automatically when you reindex or update a collection, so you should not need to do additional optimizing. One situation when you might want to optimize a collection is just before publishing it to another site or before putting it onto a read-only CD-ROM.

- Reindex—You can reindex a collection, which locates each file that already has an entry in the collection and reindexes its attributes and contents, extracting the META-tagged attributes if that option was selected when the files were originally indexed into the collection. This does not return to the original criteria for creating the collection, say `*.html`, and add any new

documents that fit the original criteria. This option also removes collection entries when the source documents have been deleted and can no longer be found.

- Remove—You can remove a collection. This only removes the collection, not the original source documents.

To perform any of the collection management tasks:

1. From the Server Manager, choose Agents & Search.
2. Click the Maintain Collection link.
3. Select the collection you want to manage from the scrollable list and information about the collection you selected is displayed.
4. To optimize a collection, click Optimize. To remove a collection, click Remove. To reindex a collection, click Reindex.

Scheduling regular maintenance

You can schedule collection maintenance at regular intervals. You can set up separate maintenance schedules for optimizing and reindexing. With normal usage, these tasks may not be necessary, but if you do a great deal of indexing and updating of collections, you may want to use some of these functions occasionally. For example, some very active web sites may require frequent reindexing if new documents are added on a daily basis.

You can optimize a collection to improve performance if you frequently add, delete, or update documents or directories in your collections. An analogy is defragmenting your hard drive. Optimizing is done automatically when you reindex or update a collection, so you should not need to do additional optimizing. One situation when you might want to optimize a collection is just before publishing it to another site or before putting it onto a read-only CD-ROM.

You can reindex a collection, which locates each file that has an entry in the collection and reindexes its attributes and contents, extracting the META-tagged attributes if that option was selected when the files were originally indexed into the collection. This does not return to the original criteria for creating the collection, say `*.html`, and add any new documents that fit the original criteria.

1. From the Server Manager, choose Agents & Search.
2. Click the Schedule Collection Maintenance link.
3. Choose a collection from the drop-down list. This lists all the collections that you have created.
4. Choose an action from the drop-down list: Reindex or Optimize. You can set up separate schedules for reindexing and optimizing the same collection.
5. In the Schedule Time field, type in the time of day when you want the scheduled maintenance to take place. Use a military format (HH:MM). HH must be less than 24 and MM must be less than 60. You must enter a time.
6. In the section labeled "Schedule Day(s) of the Week," check one or more of the day checkboxes. You can select all days. You must select at least one day.
7. Click OK to schedule the maintenance.

Unscheduling collection maintenance

If you have scheduled regular reindexing or optimizing of a collection, you can remove the scheduled maintenance when you no longer want the collection to be maintained at regular intervals. To do this:

1. From the Server Manager, choose Agents & Search.
2. Click the Remove Scheduled Collection Maintenance link.
3. Choose a collection from the drop-down list for Choose Collection. This lists all your collections for which you have set up regular maintenance.
4. Choose an action from the drop-down list: Reindex or Optimize.
5. In the lower part of the frame, you can see the time and days of the week when the scheduled maintenance is currently scheduled to take place.
6. Click OK to remove the scheduled maintenance.

Performing a search: the basics

Users are primarily concerned with asking questions of the data in the search collections and getting a list of documents in return. When you install the Enterprise server, a default set of search query and result forms are included. These allow users a simple method of accessing the search function.

There are four parts to text searching:

- making a query—you enter your search criteria.
- displaying search results—the server displays a list of the documents that match your criteria.
- viewing a document—you can view a specific highlighted document from the search results list.
- viewing the contents of a collection—you can look at the information that is maintained for each of your collections.

Search home page

The search home page, at `http://search-ui/examples`, provides individual links to each of the three search query interfaces as well as an online QuickStart tutorial on customizing the interface. The tutorial discusses the various pattern files and gives examples of how they can be changed to produce different results.

A search query

The default installation of Netscape Enterprise Server includes three search query pages: standard and advanced HTML queries and a Java-based guided query.

On the standard search query, you select a collection to search against and type in a word or phrase to search for using the query language operators.

On the guided Java-based search interface, you can use the many drop-down lists to easily construct a query. You can only obtain this interface when Java is enabled for your browser.

On the advanced HTML page, you have the additional options of selecting multiple collections to search through, establishing a sort sequence for the results, and defining how many documents are to be displayed on a page at a time (clicking the Prev and Next arrows moves you through the pages of results).

To perform a standard search, follow these steps:

1. Type this URL in the location field in your web browser:

   `http://yourServer/search`

2. In the search query page that appears, choose the collection you want to search through from the drop-down list in the Search In field.

3. Enter the word or phrase for your search query in the For field. You can create complex queries by combining operators. See "Query operators: a reference" for details about the search operators.

4. Click the Search button to execute your query.

Guided search

You can choose to use the Java-based guided search interface, which helps you construct the query. This is especially useful if you want to build a query that has several parts, say searching for a word in the documents' content as well as a specific attribute value.

Note Make sure Java is enabled for your browser. To do this, use the Languages option preferences menu command.

There are two ways to obtain the guided search page: through the Search home page or through the standard search query page.

To access guided search through the Search home page, follow these steps:

1. Type this URL in the location field in your web browser:

   `http://yourServer/search-ui/examples`

2. Click the Guided Search link on the home page.

To access guided search through the standard search query page, follow these steps:

1. Go to the standard search query page by typing this URL in the location field in your web browser:

   `http://yourServer/search`

2. Click Guided Search on the standard search page and the guided Java-based query page is displayed.

3. Choose the collection you want to search through from the drop-down list in the Search In field.

4. Use the For drop-down list to select the type of element you wish to search for. In this example, choose Words.

5. In the blank text field, type in the word you want to search for. See "Query operators: a reference" for details about the search operators.

6. Click Add Line to add the first part of the query. The word appears in the large text display box at the bottom of the form.

7. To add to your query, choose another element from the drop-down list. In this example, choose Attribute.

8. A new drop-down list appears on the right side of the form, listing all attributes that are available for the chosen collection. Choose the attribute you want to search against.

9. From the drop-down list above the text input field, choose a query operator (Contains, Starts, Ends, Matches, Has a substring) or logical operator (=, <, >, <=, >=) for your query.

10. In the blank text field, type in the attribute value you want to search for.

11. Click Add Line to add another line for your query. You can click Undo Line to remove the last line you added or Clear to remove the entire query.

12. Click the Search button to execute the search.

Advanced search

You can choose to use the advanced HTML search interface, which helps you construct the query. This is especially useful if you want to create a query that searches through more than one collection or that produces results sorted by a specific attribute value.

There are two ways to obtain the advanced HTML search page: through the Search home page or through the standard search query page.

To access advanced HTML through the Search home page, follow these steps:

1. Type this URL in the location field in your web browser:

   `http://yourServer/search-ui/examples`

2. Click the Advanced HTML Search link on the home page.

To access advanced HTML search through the standard search query page, follow these steps:

1. Go to the standard search query page by typing this URL in the location field in your web browser:

   `http://yourServer/search`

2. Disable Java for your browser. To do this, use the Languages option preferences menu command.

3. Click Guided Search on the standard search page and the advanced HTML query page is displayed.

4. In the For field, type in the word or phrase you want to search for. You can create complex queries by combining operators. See "Query operators: a reference" for details about the search operators.

5. You can type in one or more attributes to sort the results by. The default is an ascending sort order, but you can indicate a descending sort order with a minus. (See "Sorting the results" for more information about sorting).

6. Depending on how many fields are listed for each document in the search results page or how many you want to see at a time, you can expand or limit the number of matching documents you want the search to return at a time. The Prev and Next buttons allow you access to additional pages of documents if there are too many to fit on a page at once.

7. Use the drop-down list in the Search In field to choose the collection you want to search through. You can select more than one collection by holding down the Ctrl key as you click on another collection. All collections in a query must be in the same language.

8. Click the Search button to execute your query.

The search results

There are two standard types of search results: a list of all documents that match the search criteria and the text of a single document that you selected from the list of matching documents.

Listing matched documents

In the default installation of the Netscape Enterprise Server, when you execute a search from either the simple or advanced search query pages, you obtain a list of the documents that match your search criteria. The list gives some standard information about each file, depending on the collection's format. For example, the default results page for email collections give subject, to, from, and date for each entry and news collections give subject, from, and date for each entry.

The kind of file format in the collection indicates which default attributes are available for searching. See "About collection attributes" and Table 10.2 for information about the attributes for each format.

For entries resulting from a search that checks for comparative proximity of words to each other or for the exactness of the match, the file's ranking can be provided by showing a score.

If there are more matching documents than can fit on a page, click Next to see the next batch. You can always execute a new search by entering new query data and clicking Search.

Sorting the results

By default, or if you don't enter anything in the Sort By field on the advanced HTML query page, all documents matching the search are output according to their relevance ranking (for queries that consider this) or their position in the server file database (for other queries).

If you enter an attribute name in the Sort By field, the documents are displayed in an ascending sort sequence. You can list the documents in a descending sort sequence by adding a minus sign (-) prefix to the attribute, as in `-keywords` or `-title`. You can do a multiple sort, by typing in more than one field, as in `Author,-PubDate`.

In a short query, sort order usually isn't critical, but in queries that result in a great many matches, you may want to set a sort value in order to obtain useful search results. Note, however, using a special sort sequence may impact the search's performance.

Note Attribute values in META-tagged fields are text strings, which means that dates and numbers are sorted as text, not as dates or numbers. To convert the value into a date or number, you can create a new property in the Web Publishing|Add Custom Property form and check the box that marks this property as a META-tagged attribute.

Displaying a highlighted document

In the default installation of Netscape Enterprise Server, when you obtain a list of the documents that match your search criteria, you can select a single document to view in your web browser. Depending on how the pattern files are set up, the word you entered as your original search query can be highlighted in the displayed document with color, boldface text, or blinking.

To view a highlighted document, you click on a link in the document's entry in the search results. The field you use to access the highlighted document depends on how your search interface has been designed, but in the default installation, you click the document's title. When you click the title's hypertext link, there is additional code defined behind that link to format the displayed document with the search query highlighted. There is another link in the default search result entries that provides the file's URL, but clicking this is the equivalent of opening the file in your browser without any special highlighting.

In the case of documents that have been converted into HTML, the URL points you to the original document. To get to the converted HTML document, click the document's title.

Displaying collection contents

You can display the contents of your collection database to see which attributes are set for each collection. The default installation of Netscape Enterprise Server uses the `HTML-description.pat` file to display information about each of your collections that have been defined as displayable (`NS-display-select = YES`) in the `dblist.ini` file. The collection contents typically include these items:

- collection name, label, and description
- collection format
- number of attributes in the collection and a list of their names
- number of documents in the collection
- collection size and status
- language and character set
- input and output date formats

To display your collection database contents, type this line in the web browser's URL location field:

`http://`*`yourServer`*`/search?NS-search-page=c`

Using the query operators

To perform an effective search, you need to know how to use the query operators. You can only do Boolean searches, so all the subsequent information is based on Boolean search rules.

Note The query language is not case-sensitive. The examples use uppercase for clarity only.

The search engine interprets the search query based on a set of syntax rules. For example, by entering the word *region*, the actual word *region* and all its stemmed variations (such as *regions* and *regional*) are found. The search results are ranked for "importance," which means how close the matched word comes to the originally input search criteria. In the example above, *region* would rank higher than any of the stemmed variants.

Not all queries rank their results. For example, queries that check whether a given string matches the value in a field cannot perform a comparison: either the string matches the value or it doesn't. The same is true for checking whether a string is contained in a field, or begins or ends a field.

Default assumptions

The search query language has some implicit defaults and assumptions that dictate how it interprets your input. In some cases, you can circumvent the defaults, but here is how the search engine decides what you want as the search results:

`<STEM>`—Search finds all documents that contain any stemmed variant of the search word or phrase. The search engine looks at the meaning of the word, not just its spelling. For example, if you want to search on *plan*, the results would include documents that contain *planning* and *plans*, but not those that contain *plane* or *planet*.

`<MANY>`—Search considers how often the search word or phrase appear in the found documents and ranks the results for frequency (or *relevancy*).

`<PHRASE>`—Search considers words separated by spaces to be part of a phrase. For example, *Monterey otter* is interpreted as a phrase and both must be present and together to be found. Such a search would not find documents containing *sea otter* or *Monterey Bay*.

Note In any case where it's not clear that two words are to be considered as a phrase, you can use parentheses for clarity. For example, *<PHRASE> (rise "and" fall)*.

`OR`—Search considers each word or phrase in the query separated by a comma to be optional, although at least one must be present. In effect, this is an implicit OR operation. For example, *Monterey, otter* is interpreted as find documents that contain either *Monterey* or *otter*. Note that angle brackets are not require for `OR`.

Search rules

To create complex searches, you can combine query operators, manipulate the query syntax, and include wildcard characters.

Angle brackets

With the exception of the `AND`, `OR`, `NOT`, and the date and numeric comparison operators, you need to enclose query operators in angle brackets, as in `<CONTAINS>` and `<WILDCARD>` .

Combining operators

You can combine several query operators into a single query to obtain precise results. For example, you can input the following query to limit your search to those documents that have *Bay* and *Monterey* but excludes those that also mention *Aquarium*

```
Monterey AND Bay NOT <CONTAINS> Aquarium
```

You can achieve even greater precision by including some implicit phrases, as in the following query that finds documents that refer to the *Monterey Bay Aquarium* by its full name and also mention *otters* but do not refer to *shark*:

```
Monterey Bay Aquarium AND otter AND NOT shark
```

Using query operators as search words

You can use any of the query operators as a search word, but you must enclose the word in quotation marks. For example, you could search for documents about the *ebb and flow* of the tides with the following query:

```
<CONTAINS> ebb "and" flow
```

Canceling stemming

You can cancel the implicit stemming by using quotation marks around a word. For example, you can be exact by using a query such as this:

```
"plan"
```

This search only results in documents that contain the exact word *plan*. It ignores documents with *plans* or *planning*.

Modifying operators

You can use AND, OR, and NOT to modify other operators. For example, you may want to exclude documents with titles that contain the phrase *theme park*. A query such as this would solve this problem:

```
Title NOT <CONTAINS> theme park
```

Determining which operators to use

Use the following reference to help determine which operators to use. Note that the query language is not case-sensitive, so <starts> and <STARTS> are equivalent. This document uses uppercase for clarity only.

Table 10.3 Deciding which operator to use

Type of Search	Valid Operators	Examples
Finding documents by date or numeric value comparison.	is equal to (=), greater than (>), greater than or equal to (>=), less than (<), less than or equal to (<=)	`DATE >= 06-30-96` Finds documents created on or after June 30, 1996.
Finding words or phrases in specific document fields or in specific locations in the field.	<STARTS>, <CONTAINS>, <ENDS>, is equal to (=)	`Title <STARTS> Help` Finds documents with titles that start with *Help*.
Finding two or more words in a document.	AND, <NEAR/1>	`specifications AND review` Finds documents that contain both *specifications* and *review*.

Query operators: a reference

The following table describes some commonly used operators and provides examples of how to use each one. All are relevance ranked except where explicitly noted.

Table 10.4 Query language operators

Operator	Description	Examples
`AND`	Adds mandatory criteria to the search. Finds documents that have all of the specified words.	`Antarctica AND mountain climb` Finds only documents containing both *Antarctica* and *mountain climb* plus all the stemmed variants, such as *mountain climbing*.
`<CONTAINS>`	Finds documents containing the specified words in a document field. The words must be in the exact same sequential and contiguous order. You can use wildcards. Only alphanumeric values. Does not rank documents for relevance.	`Title <CONTAINS> higher profit` Finds documents containing the phrase *higher profit* in the title. Ignores documents with *profits higher* in the title.
`<ENDS>`	Finds documents in which a document field ends with a certain string of characters. Does not rank documents for relevance.	`Title <ENDS> draft` Finds documents with titles ending in *draft*.
equals (=)	Finds documents in which a document field matches a specific date or numeric value.	`Created = 6-30-96` Finds documents created on June 30, 1996.
greater than (>)	Finds documents in which a document field is greater than a specific date or numeric value.	`Created > 6-30-96` Finds documents created after June 30, 1996.
greater than or equal to (>=)	Finds documents in which a document field is greater than or equal to a specific date or numeric value.	`Created >= 6-30-96` Finds documents created on or after June 30, 1996.
less than (<)	Finds documents in which a document field is less than a specific date or numeric value.	`Created < 6-30-96` Finds documents created before June 30, 1996.

Table 10.4 Query language operators

Operator	Description	Examples
less than or equal to (<=)	Finds documents in which a document field is less than or equal to a specific date or numeric value.	`Created <= 6-30-96` Finds documents created on or before June 30, 1996.
<MATCHES>	Finds documents in which a string in a document field matches the character string you specify. Ignores documents that contain partial matches. Does not rank documents for relevance.	`<MATCHES> employee` Finds documents containing *employee* or any of its stemmed variants such as *employees*.
<NEAR>	Finds documents that contain the specified words. The closer the terms are to each other in the document, the higher the document's score.	`stock <NEAR> purchase` Finds any document containing both *stock* and *purchase*, but gives a higher score to a document that has *stock purchase* than to one that has *purchase supplies and stock up*.
<NEAR/N>	Finds documents in which two or more specified words are within N number of words from each other. N can be an integer up to 1000. Also ranks the documents for relevance based on the words' proximity to each other.	`stock <NEAR/1> purchase` Finds documents containing the phrases *stock purchase* and *purchase stock*. Ignores documents containing phrases like *purchase supplies and stock up* because *stock* and *purchase* do not appear next to each other. When N is 2 or greater, finds documents that contain the words within the range and gives a higher score for documents which have the words closer together.
NOT	Finds documents that do not contain a specific word or phrase. **Note:** You can use `NOT` to modify the `OR` or the `AND` operator.	`surf AND NOT beach` Finds documents containing the word *surf* but not the word *beach*.

Table 10.4 Query language operators

Operator	Description	Examples
`OR`	Adds optional criteria to the search. Finds any document that contains at least one of the search values.	`apples OR oranges` Finds documents containing either *apples* or *oranges*.
`<PHRASE>`	Finds documents that contain the specified phrase.A phrase is a grouping of two or more words that occur in a specific order.	`<PHRASE> (rise "and" fall)` Finds documents that include the entire phrase *rise and fall*. The *and* is in quotes to force the search to interpret it as a literal, not as an operator.
`<STARTS>`	Finds documents in which a document field starts with a certain string of characters. Does not rank documents for relevance.	`Title <STARTS> Corp` Finds documents with titles starting with *Corp*, such as *Corporate* and *Corporation*.
`<STEM>` (English only)	Finds documents that contain the specified word and its variants.	`<STEM> plan` Finds documents that contain *plan*, *plans*, *planned*, *planning*, and other variants with the same meaning stem. Ignores similarly spelled words such as *planet* and *plane* that don't come from the same stem.
`<SUBSTRING>`	Finds documents in which part or all of a string in a document field matches the character string you specify. Similar to `<MATCHES>`, but can match on a partial string. Does not work with wildcards. Does not rank documents for relevance.	`<SUBSTRING> employ` Finds documents that can match on all or part of *employ*, so it can succeed with *ploy*. **Note:** This works with literals only. If you input `web*`, the asterisk does not work as a wildcard, so the search succeeds only with the exact "web*" string.

Table 10.4 Query language operators

Operator	Description	Examples
`<WILDCARD>`	Finds documents that contain the wildcard characters in the search string. You can use this to get words that have some similar spellings but which would not be found by stemming the word. Some characters, such as * and ?, automatically indicate a wildcard-based search, so you don't have to include the word `<WILDCARD>`.	`<WILDCARD> plan*` Finds documents that contain *plan*, *plane*, and *planet* as well as any word that begins with *plan*, such as *planned*, *plans*, and *planetopolis*. See the next section for more details and examples.
`<WORD>`	Finds documents that contain the specified word.	`<WORD> theme` Finds documents that contain *theme*, *thematic*, *themes*, and other words that stem from *theme*.

Using wildcards

You can use wildcards to obtain special results. For example, you can find documents that contain words that have similar spellings but are not stemmed variants. For example, *plan* stems into *plans* and *planning* but not *plane* or *planet*. With wildcards, you can find all of these words.

Some characters, such as * and ?, automatically indicate a wildcard-based search and do not require you to use the <WILDCARD>operator as part of the expression.

Table 10.5 Wildcard operators

Character	Description
*	Specifies 0 or more alphanumeric characters. For example, `air*` finds documents that contain *air*, *airline*, and *airhead*. Cannot use this wildcard as the first character in an expression. This wildcard is ignored in a set of ([ ]) or in an alternative pattern ({ }). With this wildcard, the<WILDCARD> operator is implicit.
?	Specifies a single alphanumeric character, although you can use more than one ? to indicate multiple characters. For example, ?at finds documents that contain *cat* and *hat*, while ??at finds documents that contain *that* and *chat*. This wildcard is ignored in a set of ([ ]) or in an alternative pattern ({ }). With this wildcard, the<WILDCARD> operator is implicit.
{}	An alternative pattern that specifies a series of patterns, one for each pattern separated by commas. For example, <WILDCARD> 'Chat{s, ting, ty}' finds documents that contain *chats*, *chatting*, and *chatty*. You must enclose the entire string in back quotes and you cannot have any embedded spaces.
[ ]	A set that specifies a series of characters that can be used to find a match. For example, <WILDCARD> '[chp]at' finds documents that contain *cat*, *hat*, and *pat*. You must enclose the entire string in back quotes and you cannot have any embedded spaces.

Table 10.5 Wildcard operators

Character	Description
^	Specifies one or more characters to exclude from a set. For example, `<WILDCARD> 'C[^io]t'` finds documents that contain *cat* and *cut*, but not *cot*. The caret (^) must be the first character after the left bracket.
-	Specifies a range of characters in a set. For example, `<WILDCARD> 'Ch[a-j]t'` finds documents that contain any four-letter word from *chat* to *chjt*.

Non-alphanumeric characters

You can only search for non-alphanumeric characters if the `style.lex` file used to create the collection is set up to recognize them. This file is in the HTML, news, and mail subdirectories in the *`server_root`*`\plugins\common\` directory.

Wildcards as literals

Sometimes you may want to search on characters that are normally used as wildcards, such as *or ?. To use a wildcard as a literal, you must precede it with a backslash. In the case of asterisks, you must use two backslashes. For example, to search on a magazine with a title of Zine***, you would type:

```
<WILDCARD>Zine\\*\\*\\*
```

Several characters have special meaning for the search engine and require you to use back quotes to be interpreted as literals. The special search characters are listed here:

- comma ,
- left and right parentheses ( )
- double quotation mark "
- backslash \
- at sign @
- left curly brace {

- left bracket [
- back quote ‘ (**Note:** You can only search on back quotes as literals if the `style.lex` file has been set up to recognize it.)

For example, to search for the string "a{b", you would type

```
<WILDCARD>‘a{b‘
```

For another example, if you wanted to search on the string "c‘t", which contains a back quote, you would type

```
<WILDCARD>‘c‘‘t‘
```

Customizing the search interface

As server administrator, you can customize the search interface to meet specific user requirements. All of the HTML-based forms that the user sees are defined through a set of pattern files that set up display formats for the search results page header and footer as well as each search result record listed in response to a query. There are a set of pattern variables that you can use to construct the forms used for search input and output. Many of the variables are defined in the system and user configuration files (`userdefs.ini`, `webpub.conf`, and `dblist.ini`, which are discussed in “Configuring manually.”)

Note The search home page, at `http://`*`yourServer`*`/search-ui/examples`, also provides an introduction to the search interface as well as an online QuickStart tutorial on customizing the interface. The tutorial discusses the various pattern files and gives examples of how they can be changed to produce different results.

HTML pattern files

A good place to begin customizing the interface is by modifying the existing pattern files. After you see how they work and you understand pattern variables, you can create your own pattern files and change the configuration files and other pattern files to point to them. In the default installation of Netscape Enterprise Server, the pattern files are in this directory: *`server_root`*`\plugins\search\ui\text`. (Make copies of your original pattern files so you can restore them afterwards.)

There are pattern files for different kinds of collections: email, news, ASCII, PDF, and HTML as well as one for the web publishing collection. (The web publishing pattern file is a special case, using a great many collection-specific attributes as variables in the `dblist.ini` file.) There are several general types of pattern files, each of which has a particular use:

- `query.pat` displays the standard and advanced query pages
- `tocstart.pat` displays the header across the top of the search results page
- `tocrec.pat` displays each document listed on the search results page
- `tocend.pat` displays the footer across the bottom of the search results page
- `record.pat` displays a single highlighted document from the search results page (See "Displaying a highlighted document" for more information.)
- `descriptions.pat` displays the collection contents

The pattern files contain HTML formatting instructions, which define how elements look, and HTML search arguments and variables, which define the text label or value that is displayed.

There are three kinds of pattern variables (discussed further in "Using pattern variables"):

- user defined, in the `userdefs.ini` file, with a `$$` prefix (See "User-defined pattern variables").
- defined in the configuration files, `webpub.conf` and `dblist.ini` files, with a `$$NS-` prefix (See "Configuration file variables," Table 10.8, and Table 10.9).
- search macros and variables generated by a pattern file, with a `$$NS-` prefix (See "Macros and generated pattern variables" and Table 10.10).

To see how these work together, here are some lines from the standard query pattern file, `NS-query.pat`:

```
<input type="hidden" name="NS-max-records" value="$$NS-max-records">

<td align=left colspan=2>$$logo</td>
<td align=right><h3>$$sitename</h3></td>
```

```
<td align=right><b>$$queryLabel</b></td>
<td align=left> <input name="NS-query" size=40 value="$$NS-display-query"></td>
```

Each line contains standard HTML tags and one or more variables with the `$$` or `$$NS-` prefix. Examining each line more closely requires looking at the configuration files mentioned in "Configuring manually."

- `NS-max-records`: Defined in the `webpub.conf` file. Because this field is hidden, users cannot change this value, which defines how many matching documents to return at a time. In the advanced HTML query pattern file, `NS-advquery.pat`, this is a user-modifiable input field.
- `$$NS-max-records`: The search generates a variable from this field that can be used in subsequent searches to calculate how many result records to display at a time. Because this field is not modifiable here, the value is set to that in `webpub.conf` file. In the advanced query, this value could vary for each query.
- `$$logo`: Defined in the `userdefs.ini` file. This could be any image or text the user wanted to display on the form.
- `$$sitename`: Defined in the `userdefs.ini` file as the server's host name that is provided by the `$$NS-host` search macro.
- `$$queryLabel`: Defined in the `userdefs.ini` file as a text label for the query input field. In this case, the label on the form is the word "For:"
- `NS-query`: Defined in this pattern file as the name of the input field.
- `$$NS-display-query`: Defined in the `userdefs.ini` file. The search generates a variable from this field that can be used in subsequent searches to determine which word or phrase to highlight when an entire matching document is displayed.

Search function syntax

The search function uses standard URL syntax with a series of name-value pairs for the search arguments. This is the basic syntax:

`http://`*`yourServer`*`/search?`*`name`*`=`*`value`*`[&`*`name`*`=`*`value`*`][&`*`name`*`=`*`value`*`]`

As you use the HTML search query and results pages, you can see search functions and arguments displayed in the URL field of your browser. When entered directly into the URL field, these are sometimes called *decorated URLs*. You can also embed them in your pattern files with the HREF tag.

You can create a complete search function as an HREF element within a pattern file. The example given is from the HTML-descriptions.pat file, which defined how collection information is displayed. The following lines produce a heading for each collection for the label ("Collection:") and provides a link to the actual collection file through the collection's label (`NS-collection-alias`) that was defined in the `dblist.ini` file.

```
<td colspan=6><font size=+2><b>$$collectionLabel</b>
<a href=$$NS-server-url/search?NS-collection=$$NS-collection>$$NS-collection-alias</a>
</font></td>
```

The HREF contains a complete search function by using the following elements:

- `$$NS-server-url`: A search macro that determines the user's server URL.
- `/search`: The search command itself.
- `?`: The query string indicator. Everything after the ? is information used by the search function.
- `NS-collection=$$NS-collection`: This uses the search macro `$$NS-collection` to define the collection's filename.

You can set up a search to use a variable conditionally so that if there is no value associated with the variable, nothing is displayed. The syntax is as follows:

```
variableName[conditionalized output]
```

For example, you could request that the document's title be output if it exists. If there is no title for this document, not even the label "Title:" is to be displayed. To do this, you would use code like this:

```
$$Title[<P>Title: <B>$$Title</B>]
```

URL encodings

When you construct HTML instructions, whether in decorated URLs or within a pattern file, you need to follow the rules for URL encoding. Any character that might be misunderstood as part of an URL should be encoded with a code in

the format of *%nn*, where *nn* is a hexadecimal code. Blanks are converted to the + symbol (plus sign) in queries or to %20 in output. Table 10.6 shows the most commonly used URL codes.

Table 10.6 Common URL encodings

Character	Description	Code
	Space	%20
;	Semicolon	%3B
/	Slash	%2F
?	Question mark	%3F
:	Colon	%3A
@	At sign	%40
=	Equal sign	%3D
&	Ampersand	%26

Required search arguments

Although you can customize almost every aspect of query and result pages, there are some arguments required for search functions to display the different types of search pages. These arguments are required whether the search function is in a decorated URL or embedded as an HREF in a pattern file.

Search functions that display the search query page require these arguments:

- search query (the word, phrase, or attribute you want to search on)
- collection (can specify more than once for multiple-collection searches)

Search functions that display the search results page require these arguments:

- `NS-search-page=results` (or `r`, in upper- or lowercase)
- collection (can be specified more than once for multiple-collection searches)
- search query

Search functions that display a highlighted document require these arguments:

- `NS-search-page=document` (or `d`, in upper- or lowercase)
- document path
- collection (can be specified only once)
- search query (necessary if you want to highlight the query data)

Search functions that display the collection contents require only this argument:

- `NS-search-page=contents` (or `c`, in upper- or lowercase)

Using pattern variables

By using pattern variables, you can customize the search text interface and eliminate the need to update the actual HTML pages as user requirements change. For example, if the interface has graphics or text elements that change periodically, you can define a pattern variable that points to a pathname where that graphic or text is maintained and stored.

There are three categories of pattern variables:

- variables defined in the `userdefs.ini` file, to which are added a `$$` prefix in decorated URLs and pattern files. For example, `uidir`, `logo`, and `title` become `$$uidir`, `$$logo`, and `$$title`.
- variables defined in the configuration files, `webpub.conf` and `dblist.ini` files, which have a `NS-` prefix where they are defined in the configuration file and which have a `$$NS-` prefix when they are used in decorated URLs and pattern files. For example, `NS-max-records`, `NS-doc-root`, and `NS-date-time` become `$$NS-max-records`, `$$NS-doc-root`, and `$$NS-date-time`.
- search macros and variables generated by a pattern file, which always have a `$$NS-` prefix. For example, `$$NS-host`, `$$NS-get-next`, and `$$NS-sort-by`.

User-defined pattern variables

You can create any number of your own user-defined pattern variables in the user definitions file, `userdefs.ini`, or you can modify existing definitions. When one of these variables is used in a pattern file, the $$ prefix is added to it.

Variable names can have up to 32 characters or digits, or combinations of both. Characters can be letters A-Z in upper or lower case, hyphens (-), and underscores (_). Names are case sensitive.

The default `userdefs.ini` file included with Netscape Enterprise Server contains variables that are used to define the search query page (labeled `[query]` in the file, the results listing (labeled `[toc]`), the document display page, (labeled `[record]`), and the collection contents page (labeled `[contents]`). Each line begins with a variable name and is followed by a definition for that variable. Many are labels for screen elements, some are paths to other files, and some have more complex contents. For example, the following lines are from the query section of that file.

```
[query]
help=/publisher/help/srchhelp.html
title=ES3.0 Sample Search Interface
queryLabel=Search for:
collectionLabel=Collection:
booleanLabel=Boolean:
sortByLabel= Sort for:
copyright = Copyright &#169; 1997 Netscape Communications Corporation.
All Rights Reserved.
```

The file also includes references to search macros, such as $$NS-server-url, and can also refer to other user-defined variables, as in the following lines:

```
uidir = $$NS-server-url/search-ui
icondir = $$uidir/icons
```

Search macros are described further in the section "Macros and generated pattern variables."

You can use any supported HTML character entity in your variable definitions. You can use entity names that are defined in the *&name;* format as well as those defined with the three-digit code in the *&#nnn;* format. In the `userdefs.ini` code sample, the entity ` ` inserts a nonbreaking space and `&#169;` inserts a copyright symbol. Some of the more commonly used entities are in Table 10.7.

Table 10.7 Common HTML character entities

Numeric code	Entity name	Description
&#032;		Space
&#034;	&quot;	Quotation mark
&#036;	$	Dollar sign

Table 10.7 Common HTML character entities

Numeric code	Entity name	Description
&#058;	-	Colon
&#060;	&lt;	Less than
&#062;	&gt;	Greater than
&#153;	-	Trademark symbol
		Nonbreaking space
&#169;	&copy;	Copyright symbol
&#174;	&reg;	Registered trademark

Configuration file variables

Some variables are defined in the system configuration and the collection configuration files. These use a prefix of `NS-` in the configuration file to differentiate them from other markup tags in an HTML page. To use these variables as arguments to the search function, you add another prefix `$$` to the variable, as in `$$NS-date-time` and `$$NS-max-records`.

Variables that define defaults for all searches on a server are defined in the system configuration file, `webpub.conf`. For example, the default installation of Netscape Enterprise Server includes the following variables in the `webpub.conf` file:

```
NS-max-records = 20
NS-query-pat = /text/NS-query.pat
NS-ms-tocstart = /text/HTML-tocstart.pat
NS-ms-tocend = /text/HTML-tocend.pat
NS-default-html-title = (Untitled)
NS-HTML-descriptions-pat = /text/HTML-descriptions.pat
NS-date-time = %b-%d-%y %H:%M
```

Although installations may vary depending on how each server is configured, the most commonly found variables from the `webpub.conf` file are listed in Table 10.8.

Table 10.8 Commonly found variables defined in webpub.conf

Variable	Description
`NS-default-html-title`	The name given to HTML documents that do not contain a user-defined title. Typically set to "(Untitled)."
`NS-date-time`	The date and time format to use when displaying results.
`NS-date-input-format`	The format for inputting dates (the default is MMDDYY).
`NS-HTML-descriptions-pat`	The pattern file to use when displaying the contents of the collections.
`NS-largest-set`	The maximum number of records that can be handled as matching the search criteria. The records are displayed in groups of `NS-max-records.`
`NS-max-records`	The maximum size of the result set displayed at one time.
`NS-ms-tocend`	The pattern file to use for the footer at the bottom of the search results page when searching multiple collections.
`NS-ms-tocstart`	The pattern file to use for the header at the top of the search results page when searching multiple collections.
`NS-query-pat`	The query pattern file used when creating a query page.
`NS-search-type`	The type of search to perform. Only Boolean is permitted.

Collection-specific variables are defined in the `dblist.ini` file. For example, the default installation of Netscape Enterprise Server includes variables for the web publishing collection. Among the variables defined there are:

```
NS-collection-alias = Web Publishing
NS-doc-root = C:/Netscape/SuiteSpot/docs
NS-url-base = /
NS-display-select = YES
```

The variables in your `dblist.ini` file may differ according to the type of collections you are using, Table 10.9 contains some of the more commonly found collection-specific variables.

Table 10.9 Commonly found variables in dblist.ini

Variable	Description
`NS-collection-alias`	The collection's label. Can be specified more then once to search multiple collections.
`NS-doc-root`	The root directory for the documents in the collection.
`NS-display-select`	This indicates whether the collection is displayed as part of the collection information listing, when `NS-search-page=contents`. The default is YES.
`NS-highlight-start`	Begin highlighting at this point in the displayed document. Typically this highlights the search query criteria.
`NS-highlight-end`	End highlighting at this point in the displayed document.
`NS-language`	The language of the documents in the collection.
`NS-record-pat`	The pattern file to use when displaying a highlighted document page.
`NS-tocend-pat`	The footer pattern file associated with a collection to be used when formatting the search results.
`NS-tocrec-pat`	The record pattern file associated with a collection to be used when formatting the search results.
`NS-tocstart-pat`	The header pattern file associated with a collection to be used when formatting the search results.
`NS-url-base`	The base URL used when constructing the link used to locate the file.

Macros and generated pattern variables

There are some search macros that you can use in your pattern files or decorated URLs, and the search function itself generates some pattern variables that you can use in subsequent search requests to define how the later output is to be displayed. These macros and variables have a prefix of `$$NS-` to indicate their use.

For example, after doing an initial search query that results in 24 documents on the results page, you can reuse the search-generated `$$NS-docs-matched` and the `$$NS-doc-number` variables to help define a document page displaying one of the documents in detail. In this way, you can tell the user that this document is number 3 of 24 documents returned for the original search.

The search macros and the generated variables that you can use in a subsequent pattern file or decorated URL are listed in Table 10.10.

Table 10.10 Macros and generated pattern variables

Variable	Description
`$$NS-collection-list`	An HTML multiple select list of all the collections in `dblist.ini` where `NS-display-select` is set to YES.
`$$NS-collection-list-dropdown`	An HTML drop-down list version of `NS-collection-list.`
`$$NS-collections-searched`	The number of collections searched for this request.
`$$NS-display-query`	The HTML-displayable version of the query that is generated for a results page.
`$$NS-doc-href`	The HTML HREF tag for the document. This provides a URL to the original source document. For email, this is in the form `mailbox:/boxname?id-messageID` and for news, it is in the form `news:messageID`.
`$$NS-doc-name`	The document's name.
`$$NS-doc-number`	The sequence number of the document in the results page list.
`$$NS-doc-path`	The absolute path to the document.
`$$NS-doc-score`	The ranked score of the document (ranges 0 to 100).
`$$NS-doc-score-div10`	The ranked score of the document (ranges 0 to 10).
`$$NS-doc-score-div5`	The ranked score of the document (ranges 0 to 5).
`$$NS-doc-time`	The creation time for a document in the results list. To obtain this value, you must set `NS-use-system-stat = YES` in the `webpub.conf` file. By default it is set to NO, since system statistics are expensive.
`$$NS-doc-size`	The size of the document rounded to the nearest K. To obtain this value, you must set `NS-use-system-stat = YES` in the `webpub.conf` file. By default it is set to NO, since system statistics are expensive.
`$$NS-docs-found`	The actual number of documents that the search engine found for this request.
`$$NS-docs-matched`	The number of documents returned from the search (up to `NS-max-records`) for this request.
`$$NS-docs-searched`	The number of documents searched through for this request.

Table 10.10 Macros and generated pattern variables

Variable	Description
`$$NS-get-highlighted-doc`	This provides the URL for a highlighted document in order to be able to display the document as HTML text with highlights.
`$$NS-get-next`	This variable gets the next set of search results to be displayed. The set is equal to `NS-max-records` and is positioned by using `NS-search-offset`.
`$$NS-get-prev`	This variable gets the previous set of search results that has been displayed. The set is equal to `NS-max-records` and is positioned by using `NS-search-offset`.
`$$NS-host`	The host name.
`$$NS-insert-doc`	A placeholder used in the `NS-record-pat` pattern files for HTML to indicate where the source document is to be inserted.
`$$NS-rel-doc-name`	The relative name of the document to display creating a document page.
`$$NS-search-offset`	The offset into the set of records returned as search results. Used to determine which set of records are displayed when you use `NS-get-next` and `NS-get-prev`.
`$$NS-server-url`	The URL for the server.
`$$NS-sort-by`	The sort sequence for the items on the results page. You can select one or more of the available attributes for the collection. The default is an ascending sort.

Chapter

11

Using agents

Netscape Enterprise Server allows you to use server-based agents to manage server files and folders. Agents act as watchdogs for you, watching for a specific event or time, and then performing a task for you. For example, you could set up a document agent to notify you when a specific URL has been updated, or you could have a search agent execute every week at the same time to pull a list of all web publishing documents that have been updated during the week. The notification could be an email message or a posting to a newsgroup.

An agent is stored on the server, so you must be connected to the server when you create the agent. The agent resides on the server until it is deleted or completes the assigned task. The server only allows users with the correct permissions, as recorded on the access control list (ACL), to submit an agent. An agent can only perform operations that you are authorized to perform, but it cannot access authenticated sites because agents don't send authorization data with their requests.

As server administrator, you can configure how your server manages agents. For example, you can define who has access to specific agent events and you can restrict who can create or disable agents.

One of the options you have as server administrator is to turn web publishing off. When you do so, all agents for the server are also turned off and clients cannot use Netscape Web Publisher to access agent services. When you turn

web publishing back on, agents that were turned off because you turned off web publishing are also turned back on. Agents that were disabled for other reasons are still disabled.

Note Because agents are stored on the server, you must have sufficient disk space available for all agents created on your server. A general rule of thumb is to allow 512 bytes per agent, which calculates out to approximately 70-100MB of space for 100,000 agents.

Types of agents

There are several types of agents, each of which has a particular use. When an event occurs that an agent is monitoring or when the specified time for activation occurs, the agent activates and begins to perform its assigned actions, such as:

- Sending an email message
- Posting a news article to one or more newsgroups
- Performing an HTTP operation to post to a URL or to get a URL (advanced options only)

Timer agents

Timer agents respond to time-related events that occur as a result of the date or time. You can submit an agent to activate:

- On a specified date and time
- At recurring time (for example, every Tuesday at 10 a.m.)
- At periodic intervals (for example, every five hours)

Document agents

Document agents respond to document-related events that take place when something has occurred to a document on the server. Some examples are:

- A document is changed, moved, copied, or deleted
- Someone views or modifies a document attribute
- A document is locked or unlocked

Directory agents

Directory agents respond to directory-related events that take place when something has occurred to a directory on the server. Some examples are:

- A directory is changed, moved, copied, or deleted
- A directory is added
- Someone lists a directory
- Someone views or modifies a directory attribute

Search agents

Search agents execute periodically, notifying the client of any documents that have been modified since the last time the search agent executed. Search agents can also check the content of documents, pulling a list of all documents in the chosen collection that contain the specified search criteria or text string.

You can limit the content search to recently modified documents or you can extend the search to include all server documents. Some examples of search agent tasks are:

- Check the server at 5 a.m. every Monday morning for all documents that have been modified in the preceding week.
- Check the server at 5 a.m. every Monday morning for all documents that contain the string "JavaScript" that have been modified in the preceding week.
- Check the server at 5 a.m. on the first of each month for all new documents with the word *Netscape* in their title.

Creating authorized users

Only users that you have added to an access-control database are permitted to use agents. You must use the Administration Server to add users and groups.

Agents access the ACL database that is the default for your server. The local LDAP database is typically set as your default during server installation, but you can direct agent services to access a different ACL database. See Chapter 6, "Controlling access to your server" for more information about setting access control.

Configuring agent services

As server administrator, you must begin by enabling and configuring agent services. You can enable or disable agent services as well as define many default characteristics of individual agents.

Note Before you can use agent services on your server, you *must* define the MTA (mail) and NNTP (news) hosts for your server. To do this, use the Server Manager for your Enterprise Server. Use the Server Preferences|Network Settings link and enter values for the MTA Host and NNTP Host fields.

To set up agent services for your server, follow these steps:

1. From the Server Manager, choose Agents & Search.
2. Click the Agent Management link.
3. Click the Yes radio button under the Enable Agent Services label.
4. Type in the full path of the agent directory in the Agent Directory field. This is where files containing information about agents are kept. The default is *server_root*\https-*yourServer*\agents-db. If you want to specify a different directory, make sure you include a full path.
5. Type in the maximum number of agents for all users that are allowed on your server. The number you specify must be an integer greater than 0.
6. Type in the maximum number of agents an individual user can have. The number you specify must be an integer greater than 0.
7. Type in the maximum number of days any agent can exist. The number you specify must be an integer greater than 0. This provides a calculated expiration date for your agents. Clients cannot input a longer agent life for a particular agent.
8. Type in the maximum times an agent can be activated. The number you specify must be an integer greater than 0. Clients cannot input a larger amount of activations for a particular agent.
9. Type in the agent's server administrator's email address. This is the "From" address included on any emails that are sent from the server. This can also be used to identify agents that encounter error conditions.
10. Type in the name of your organization. This identifies the organization associated with this server.
11. Type in the minimum timer resolution, in minutes. This must be an integer and must be greater than or equal to 5. This limits the period that clients can input as the interval at which periodic timer agents can activate.
12. Click OK to apply your changes.

Agent information in the configuration files

There are several configuration files that govern how agent services operate. In general, you don't access these files, but this section briefly introduces them just in case you do need to know something about them.

The system configuration file is mapped to the `agent_system.ini` file in the `obj.conf` file. This defines your system and data directories.

This in turn points to the `agent_string.ini` file that contains the text strings that are used to create the HTML agent services forms.

The information that you enter through the Agents & Search |Agent Management form (See "Configuring agent services" for more information), is reflected in the `agent.conf` file.

The access control list (ACL) data is configured in the `magnus.conf` file, which points to another file, *`server_root`*`\httpacl\generated.https-`*`yourServer`*`.acl`.

Recovering agent files

Agent information is stored in a set of database files on the server. It is possible for the data in these files to become inconsistent or corrupted. If this happens, you can use the command-line utilities (provided with the web server) to salvage and recover agent information from the corrupted files.

The next sections explain how agent information is stored and how to recover this information if file corruption occurs.

How agent information is stored

Agent information is stored in this database file: *`server_root`*`\https-`*`yourServer`*`\agents-db\ns-agent-base`

In order to provide quicker access to information in this file, the web server also creates and uses three index files.

- `ns_agent_base.idx` is an index to the main agent database. It provides the other index files with quick access to the agent database.
- `ns_agent_user.idx` provides the server with a quick way to find agents based on username. (Essentially, this file contains usernames and pointers to locations in the main database file. Given a username, the server can use this index file to "look up" that user's agents in the main database file.)
- `ns_agent_class.idx` provides the server with a quick way to find agents based on class (or type). (This file contains classes and pointers to locations in the main database file. Given a class, the server can use this index file to "look up" agents of that class in the main database file.)

Fixing inconsistencies and file corruption

The set of files that store agent information are in one of the following three states:

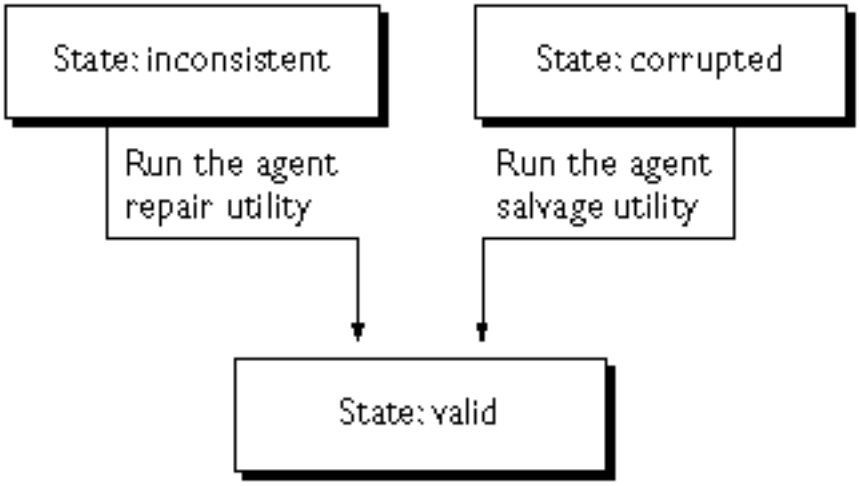
- Valid (nothing is wrong with the files)
- Inconsistent (the index files are not in sync with the main database file)
- Corrupted (the main database file has become corrupted)

If the files are in the "Inconsistent" state, you need to run the `agent_repair` command-line utility to fix the index files. For details, see "Recovering from inconsistencies" on page 228.

If the files are in the "Corrupted" state, you need to run the `agent_salvage` command-line utility to recover the data. For details, see "Recovering from file corruption" on page 229.

Figure 11.1 illustrates these different states and the utilities that you need to use to return the files to a "Valid" state.

Figure 11.1 Possible states of the files storing agent information



Recovering from inconsistencies

In some cases, the index files develop inconsistencies and the server cannot be able use them to find entries in the main database file (`ns_agent_base`). When this situation occurs, the message "Agents database has become internally inconsistent" is logged to the `agent.log` file, and the server displays the following error message:

```
Agent store files are out of sync
```

Because both index files consist of data that is already stored in the main database file, you can recover the index files from the data in the main database file.

To recover the index files, run the `agent_repair.exe` utility, which is located in the *`server_root`*`\Netscape\SuiteSpot\plugins\agents\bin` directory. Unlike with corrupted files, inconsistent files are repaired in place, with the "new" files overwriting the original files.

To run this utility:

1. Begin by shutting down your web server.
2. On the command line, type in this command:

```
agent_repair filePathname/filePrefix
```

filePathname is the pathname for the agent file directory. The default is *server_root*\https-*yourServer*\agents-db.

filePrefix is the prefix that is common to the names of the database and index files. Typically, this prefix is ns_agent.

For example, you would execute a command similar to the following command:

```
agent_repair server_root\https-yourServer\agents-db\ns_agent
```

Recovering from file corruption

In some cases, the main database file (ns_agent_base) might get corrupted. When this situation occurs, the server displays the following error message:

```
Agent store files are corrupted
```

The message "Agents database has become corrupted" is logged to the agent.log file. You can change this message by changing the text in the agent_strings.ini file. If you do so, the new message appears instead of this standard one.

If this happens, you need to recover the data from the main database file. (Unlike the problem with inconsistent indexes, corruption in the main database file may result in data loss.)

To recover the data, run the agent_salvage.exe utility, which is located in the *server_root*\Netscape\SuiteSpot\plugins\agents\bin directory. This utility retrieves data from the corrupted files and creates new files for the data. Unlike with inconsistent files, corrupted files cannot be repaired in place, so new files are created in addition to the original files rather than overwriting them.

To run this utility:

1. Begin by shutting down your web server.
2. On the command line, type in this command:

   ```
   agent_salvage filePathname/filePrefix newFilePrefix
   ```

filePathname is the pathname for the agent file directory. The default is *server_root*\https-*yourServer*\agents-db.

filePrefix is the prefix that is common to the names of the database and index files. Typically, this prefix is ns_agent.

newFilePrefix is a prefix that you assign to the newly created files that contain the recovered data.Typically, this prefix is ns_agent_recovered.

For example, suppose you run the following command:

```
agent_salvage server_root\https-yourServer\agents-
db\ns_agent ns_agent_recovered
```

The agent_salvage utility retrieves data from the following corrupted files:

```
ns_agent_base

ns_agent_user.idx

ns_agent_class.idx
```

The utility then creates the following new files in the same directory as the original files and saves the data to these files:

```
ns_agent_recovered_base

ns_agent_recovered_user.idx

ns_agent_recovered_class.idx
```

The utility also displays the following message to the user console when it is finished:

```
Agents database has been repaired
```

Accessing agent services

There are two ways to access the agent services user interface: you can go directly through the server or through Netscape Web Publisher.

To access agent services through the server, type in the following URL:

http://*yourServer*/agents

To access agent services through Web Publisher, you can either of these methods:.

1. In the Web Publisher applet window, from the Services menu, choose Agent Services to go to the agent services page.

2. From the Netscape Web Publisher Services window, click the Agents link.

The Netscape Enterprise Server Agent Services page appears. The lower left frame contains the links you need to create and view agents. The lower right frame describes agent services and will be used to display information about a specific agent once you have selected one.

For further information, see the *Web Publisher User's Guide,* which is available through the online help system in user component such as agent services, search, and Web Publisher. To access help information, you can use the Help menu command in Web Publisher, or you can click the Help button on the Agent Services page, on the Web Publisher Services page, or on any of the search interface forms.

Chapter

12

Configuring web publishing

Netscape Enterprise Server 3.0 clients can use Netscape Web Publisher to collaborate on projects by directly accessing, editing, and managing file on remote servers. Web Publisher provides sophisticated features for server clients, such as file management, editing and publishing, document version control, search, agent services, access control, and link management.

As the server administrator, you can set many options that define how web publishing works for your server clients and how your server's web publishing data is maintained. One of the most important functions you can perform for your users is to create a database of searchable web publishing data. This requires using the Index and Update Properties function to index a set of documents and directories so that when users start up Web Publisher, they can search on the contents and properties of these files.

Other web publishing setup and configuration functions are:

- turn web publishing on and off for your server
- set the language that Web Publisher uses
- turn off the link management component of web publishing, or you can turn off or on only the automatic link update feature
- define the archive directory for version-controlled files
- unlock files that a client may have locked, thereby making them available again to other users

- add and manage custom web publishing file properties
- maintain web publishing data

For further information about Netscape Web Publisher, see the online *Web Publisher User's Guide*. This is available through the online help system in user components such as agent services, search, and Web Publisher. To access the help system, you can use the Help menu command in Web Publisher, or you can click the Help button on one of the search interface forms, on the Agent Services page, or on the Web Publisher Services page.

Setting access control for owner

The access control system supports a special user called *owner*. When an ACL rule designates the user to be the owner, the permissions defined by this rule apply to the owner assigned by Web Publisher for each document. For example:

```
allow (write, delete) user = owner;
```

Note Do not create a user with the username of `owner`.

Ownership of web publishing documents can be assigned either through the Web Publishing|Index and Update Properties form or through Web Publisher. The Index and Update Properties form allows you to do a bulk assignment of ownership to a set of documents and Web Publisher performs individual assignments of file ownership to a user when the user publishes or uploads the file.

Only the owner can modify the access control (ACL) rules for a file. These rules define the actions users can perform on the file, such as moving, copying, renaming, or deleting it. An owner can reassign ownership of a file to another user, and if a file has no owner, anyone with a valid username can identify themselves as its owner. Because the username identified as the owner of a file can change, any access control that you place on a file should target the owner of a file rather than a specific username.

If the default access control (ACL) that governs your server is not restrictive or flexible enough for your web publishing needs, you can use the Server Preferences|Restrict Access function to create an ACL that is more appropriate for web publishing.

For example, you could create an ACL like this:

```
acl "uri=/publisher/";
allow (read, execute, list, info) user = anyone;
allow (write, delete) user = owner;
```

This ACL sets a restriction such that only the owner of a file within the additional document directory of `/publisher` can modify or delete the file.

See Chapter 6, "Controlling access to your server" for more information about setting access control.

Indexing and updating properties

Before users can perform a search across a set of documents and directories, information about the documents and directories needs to be indexed into the web publishing database. The web publishing database is stored as a search collection and is created as part of the server installation process. Initially it contains no data and must be populated by indexing the documents in the document directories.

The Web Publisher window lists the files and folders that are in the document directory selected when a user starts up Web Publisher, but the data initially is not indexed (and therefore is not available for searching) and the files have no owners (so anyone can define their username as the owner of a file, and thereby be able to set the access control for a file).

You can use the Index and Update Properties form to perform bulk indexing of documents to create searchable web publishing data and you can also use it to do a bulk assignment of owner for the files included in the collection. You can restrict or expand the scope of documents and directories to be indexed, and you can index just the file properties, called *metadata*, or you can also index the documents' contents. If you choose to index the contents of the files, you can search on any word in the documents although publishing and uploading files with Web Publisher may be slightly slower.

1. From the Server Manager, choose Web Publishing.
2. Click the Index and Update Properties link.

3. The Document Directory field displays the currently selected directory. You can index documents in the primary document directory, an additional document directory, or in a subdirectory.

   If you want to index a different directory, click the View button to see a list of directories. You can index any directory that is listed or you can view the subdirectories in a listed directory, and index one of those instead.

   Once you click the index link for a directory, you return to the Index and Update Properties form and the directory name appears in the Document Directory field.

4. To also index the subdirectories within the specified directory, click the Include Subdirectories checkbox.

5. You can index all files in the chosen directory by leaving the default *.* pattern in the "Include files matching pattern" field or you can define your own wildcard expression to restrict indexing to documents that match that pattern. For example, you could enter `*.html` to only index the content in documents with the `.html` extension, or you could use this pattern (complete with parentheses) to index all HTML documents:

   ```
   (*.htm|*.html)
   ```

   You can define multiple wildcards in an expression. See Chapter 3, "Managing your server" for details of the syntax for wildcard patterns.

6. If this is the first time you index web publishing documents, check the "Index unindexed documents" checkbox. In subsequent indexing operations, you can uncheck it or you may leave it checked to index any new documents that have been added to the document directory.

7. If you want to make a change to files that have already been indexed, you can use the "Update previously indexed documents" option to do a bulk ownership assignment or to index the content of files that did not have this option set when they were first indexed. These options are useful when you change many files at once. You can use the Web Publisher client to index and update individual files.

8. To do a bulk assignment of ownership to all files that match your criteria, you can check the "Set document owner to" checkbox and type in a username. Be sure to type in a valid username because the server does not perform any validity checks on the name. This updates the owner property in each file's collection entry.

9. To index the document content, check the "Index document contents" checkbox. You can choose to index the documents' contents as well as their file metadata.

10. Click OK to begin indexing and updating web publishing.

A summary of the indexing operation is displayed in the web browser window. The information is also logged to a local log file.

Note Once you have indexed documents into the web publishing collection, you should not change any document directory's URL mapping or the collection's entries will target the URL mapping to the wrong physical file location. If you have to change a document directory, you need to reindex the documents in the new location. You can use the Repair function to remove the indexed data from the old directory mapping.

Changing the web publishing state

You can deactivate web publishing and you can turn it back on. If you turn off web publishing, you also turn off link management. Documents that are subsequently moved or renamed may have incorrect links, and the link status database may not be up to date. The solution is to use the Web Publishing|Link Management function to manually turn link management off and then turn it back on again. This starts the link management function up again with an empty link status database. See "Changing the link management state" for more details of link management.

Note If you turn web publishing off, all agents for the server are also turned off and clients cannot use Netscape Web Publisher to access agent services. When web publishing is turned back on, agents that were turned off solely because web publishing was turned off are turned back on. Agents that were disabled for other reasons are still disabled.

To change the web publishing state:

1. Select your server on the Server Administration page.
2. From the Server Manager, choose Web Publishing.
3. Click the Web Publishing State link.
4. To turn web publishing on, click the On radio button. To turn it off, click the Off radio button. The default value is On.
5. Click OK to change the state of web publishing on your server.

Setting the web publishing language

You can change the web publishing language to any language supported by the user's installation, and these are listed for the server administrator in a drop-down list on the form.

Note Be cautious when using this function. If you change the language of a collection, the system deletes all the existing data in the collection.

1. From the Server Manager, choose Web Publishing.
2. Click the Web Publishing Language link.
3. Select a language from the drop-down list. The default is English.
4. Click OK to set the language.

Your changes are reflected in the `language.conf` file, which is located in the *`server_root`*`\plugins\search\admin` directory.

After you change the web publishing language, your server is automatically restarted to apply the change.

Maintaining web publishing data

Web Publisher maintains multiple sets of data about the documents that are in the web publishing collection. When all web publishing data is synchronized, each file in the chosen document directory has a record in the web publishing collection and every property record in the collection has a corresponding file in the document directory.

Although you can limit the scope of the Repair and Report functions to checking only the files in a particular document directory for collection records, every property record in the collection is checked for a corresponding source document regardless of which directory the file might be in.

Occasionally, these can become out of synch. You can obtain a report on the state of your web publishing files, and then repair one or more directories as needed. For example, if a document that was indexed into a collection is deleted, there is a record in the collection that no longer has any corresponding source document. Repairing removes the collection records for any such document.

You can perform these functions to maintain your web publishing data:

- Report on the collection's data—You can produce a report on the current logical consistency of the web publishing collection's data. This lists all the files in the selected document directory and also lists all the records in the web publishing collection, regardless of which directory the collection data corresponds to. The report indicates which files are not yet indexed (and therefore don't have records in the web publishing collection) and which records have no source document (and therefore should be repaired). The report highlights errors and indicates what the result of the repair would be. For example, "Repair will delete Properties Record."

  The report provides a short summary at the end of the log file, indicating how many directories and files have been checked, how many repairs are recommended, and how many errors have been encountered.

- Repair the collection—You can repair the web publishing collection's logical consistency. This function repairs the files in the selected document directory and produces a report similar to that from the Report function. The Repair function indicates on the report which repairs have been completed and what the repair accomplished. For example, "Repair: Removing Properties Record."

- Optimize the collection—You can optimize the web publishing collection to improve performance if you frequently add, delete, or update documents or directories in your collections. An analogy is defragmenting your hard drive. Optimizing is done automatically when you reindex or update a collection, so you should not need to do additional optimizing. One situation when you might want to optimize a collection is just before publishing it to another site or before putting it onto a read-only CD-ROM.

Periodically, you may want to maintain your web publishing collections. You can perform the following collection management tasks:

1. From the Server Manager, choose Web Publishing.
2. Click the Maintain Web Publishing Data link.
3. You can define the scope of the Repair and Report functions by choosing the document directory to check through. If you want to use a different directory, click the View button to see a list of directories. You can report on or repair any directory or subdirectory that is listed.

   Once you click the link for a directory, you return to the Maintain Web Publishing Data form and the directory name appears in the Document Directory field.
4. To also report on or repair the subdirectories within the specified directory, click the Include Subdirectories checkbox.
5. To report on the collection, click the Report button. This reports on the selected document directory.
6. To repair the collection, click the Repair button. This repairs inconsistencies in the selected document directory.
7. To optimize the collection, click the Optimize button. This optimizes the entire web publishing collection.

Changing the link management state

At times, you may not need automatic link checking and updating. At these times, you can turn link management off to conserve resources and to improve searching and indexing performance. When you turn link management off, Web Publisher stops doing automatic link checking and you cannot use the Check Links function from the Web Publisher Services page.

You can also use this form to selectively turn the automatic link update feature on and off. When automatic link updating is on, Web Publisher changes the outgoing and incoming links in a file to keep them up to date as files are moved and renamed in Web Publisher. Because this revises the modification date for any file that has updated links, this feature is off by default.

Note The automatic link update feature only affects links outgoing to or incoming from moved or renamed files. It does not affect HTML files that are being uploaded or published. Provided that link management is on, the links in these files are always updated as part of the upload or publish operation.

For further information about link management in Web Publisher, access the online *Web Publisher User's Guide* through the Help menu command in Web Publisher or the Help button on the search interface, the Agent Services page, or the Web Publisher Services page.

1. From the Server Manager, choose Web Publishing.
2. Click the Link Management link.
3. To change the state of link management, select the On or Off radio button.

   To deactivate link management, select the Off radio button. This clears the link status information so that when you try to check links in Web Publisher, you get an error message, and you cannot access any link status information.

   To reactivate link management, select the On radio button. This starts link management up again, which creates a new empty link status database. To get link status information, you must again check links for all your files. Links that have changed status since you turned link management off may have to be manually fixed.
4. To turn automatic link updating on, select the On radio button. You can only turn this on when link management is on.

This starts up automatic link updating, which revises links from or to files that are subsequently moved or renamed. It does not, however, affect the links in any files that were moved or renamed while automatic link updating was turned off.

5. Click OK to apply your change.

Setting the version control archive

Netscape Web Publisher includes a version control system for keeping track of files and documents as they are updated and changed. Web Publisher manages version control for you, allowing you to compare different versions of a file, providing version history for any file under version control, and automatically incrementing version numbers for files edited under version control.

For further information about version control in Web Publisher, access the online *Web Publisher User's Guide* through the Help menu command in Web Publisher or the Help button on the search interface, the Agent Services page, or the Web Publisher Services page.

Files under version control are stored in an archive directory. To specify which directory you want Web Publisher to use as the version control archive directory, follow these steps:

1. From the Server Manager, choose Web Publishing.
2. Click the Version Control link.
3. Type the full path for the archive directory in the Archive Path field. The path in the default installation is *server_root*`\plugins\content_mgr\archive`. Web Publisher uses this archive to store all files under version control.

   If you are changing the archive directory but keeping the version history intact, you *must* have (a) already created the new directory, (b) moved the version history files to the new directory, and (c) deleted the old archive directory. If you don't want to keep the old version history, you don't need to move the files to the new directory, but you must do the other two steps (a and b) or this function will fail.
4. Click OK to set the archive directory.

Unlocking files

If a file that has been locked in Web Publisher is required for another user, you can unlock it. This is true for files that were locked manually by the client or automatically by Web Publisher as part of an edit or download operation.

For further information about locking and unlocking files in Web Publisher, access the online *Web Publisher User's Guide* through the Help menu command in Web Publisher or the Help button on the search interface, the Agent Services page, or the Web Publisher Services page.

Be cautious in using this function because by unlocking a file that was locked, you are forcing the file to be available for editing by other users. This is contrary to the intent of the lock owner, who may not know of the unlocking operation.

To unlock a file:

1. From the Server Manager, choose Web Publishing.
2. Click the Unlock File link.
3. The Choose field displays the currently selected file or directory.

   If you want to unlock a different file or a file from another directory, click the View button to see a list of resources. You can unlock files that are listed or you can view the files in a listed directory, and select one of those files.

   Once you click the unlock link for a file, you return to the Unlock File form and the filename appears in the Choose field.
4. Click OK to unlock the file.

After you unlock a file, your server is automatically restarted to incorporate the lock change.

Note If you want to unlock a file that begins with a period, as in `.jshrc`, you cannot use this form to perform the unlocking. You will have to log into Web Publisher as the user and unlock the file there.

Adding custom properties

As server administrator, you can add your own custom Web Publisher file properties. These properties are added to the default set of file properties stored in the web publishing collection. Server clients can view visible custom properties in Web Publisher and use them in their document searches.

For further information about viewing and modifying properties in Web Publisher, access the online *Web Publisher User's Guide* through the Help menu command in Web Publisher or the Help button on the search interface, the Agent Services page, or the Web Publisher Services page.

Note If you want to add another custom property after creating the maximum number of custom properties for a given type, you cannot remove an existing custom property and "reuse" the property's slot in the collection by adding a new custom property of the same type. For example, if you want to add a numeric property after 5 have already been created, you cannot delete one of the existing 5 numeric properies and add another numeric property in its place. The only way to use the new property is to remove the entire collection and recreate it with the new property.

To add a custom file property:

1. From the Server Manager, choose Web Publishing.
2. Click the Add Custom Properties link.
3. Type a name in the Property Name field. The name has these restrictions:
    - It cannot duplicate an existing Web Publisher property name.
    - It cannot exceed 128 characters.
    - It cannot be "." or ".." or contain spaces.
    - It cannot begin with an underscore or have an underscore as the third-to-last character.

4. Select the property's type from the Property Type scrollable list. This value is not modifiable. There is a limit to the number of each type you can have. These are the default settings:

   - Text (a maximum of 30).
   - Numeric (a maximum of 5).
   - Date (a maximum of 5). Dates are formatted as month/day/year, and year can be two or four digits.

   You can change the maximum settings for these in the `webpub.conf` file, although larger sets of attributes impact the performance of your server.

Note You cannot use the additional attributes in the existing web publishing collection. If you want to use the new attributes in the web publishing collection, you must use your file system to remove both the `web_htm` and `link_mgr` collection files from the search collections directory and then restart your server. See "Configuring manually" in Chapter 10, "Using search" for details on how to change the `webpub.conf` file.

5. Click one of the Permissions buttons, either Read only or Modifiable. By default, this is set to Modifiable.

Note For modifiable custom properties defined as META-tagged attributes, the value in the document is extracted only the first time the document is indexed. Because users can input a different value in the attribute field through the Web Publisher Services Properties page, the server ignores the META-tagged value in subsequent indexing. In this way, the user's value is not overwritten.

6. Click one of the Visible to User buttons, either Invisible or Visible. By default, this is set to Visible. This defines whether server clients can view the property through Web Publisher.

7. If the property you are adding is actually an HTML file attribute that has been tagged with the HTML META tag, you can check this checkbox. From this point onward, when files containing this attribute are indexed, the contents of the META attribute is used as the value of the property and you can search for files that contain this META-tagged property. The property must conform to the same conventions as property names.

Note Because all attributes tagged with META are defined as text, sorting operations on fields containing dates or numbers do not sort in the expected date or number order. With this feature, you can redefine META-tagged attributes to dates or numeric values to obtain valid sort sequences.

8. Click OK to create the new custom property.

Managing properties

You can list all the file properties that are available for use. These include the default set plus any new custom properties you have created. You can remove or edit only those properties that you have created. These have active Remove and Edit links in the first two columns.

To manage file properties:

1. From the Server Manager, choose Web Publishing.
2. Click the Manage Properties link to obtain a listing of all available properties.

To remove a custom property:

1. Click the Remove link for the property. The Remove Custom Property form appears.
2. Click OK to remove the property. Click Back to return to the Manage Properties page without removing the property.

To edit a custom property:

1. Click the Edit link for the property. The Edit Custom Property form appears.
2. Change the property as needed. You can only change the property's name, permissions, visibility and its option of whether to capture META-tagged attributes.
3. Click OK to update the property with your changes. Click Back to return to the Manage Properties page without editing the property. Click Reset to reset any property values you changed.

Chapter

13

Cataloging your web site

This chapter describes how you can automatically generate web pages that list and categorize the HTML files in your web site. The AutoCatalog feature lets you automatically provide your web users with easy access to your content by:

- listing all HTML documents in your web site
- generating HTML views of your web content organized by title, classification, author, and last-modified date
- generating automatic directory information (known as a resource description) for each HTML document in your document root

If your web server is one of many in a company, organization, or educational facility, you can also use the AutoCatalog feature to provide a resource description of your web site to any Netscape Catalog Server. Netscape Catalog Server can then provide a central server where users can find information on any of the individual web servers in your organization.

The AutoCatalog feature of Enterprise Server 3.0 provides only a subset of the functionality of Netscape Catalog Server. While the AutoCatalog feature can list and categorize the files on your web site, Netscape Catalog Server also indexes information, provides searching capabilities, and catalogs documents from multiple servers. Netscape Catalog server is also highly configurable, allowing users to write plug-in functions and define rules for gathering and categorizing

documents. If you would like a more robust cataloging tool, you may want to purchase Netscape Catalog Server to work in conjunction with your Enterprise Server.

What can AutoCatalog do for my web site?

If you have a large web site with many files and directories, it can be difficult to organize the content so that your users can quickly find specific information. If your web server also contains directories of information from various groups or people, the content may not be unified.

The AutoCatalog feature creates an organized catalog of all of the documents on your web server. It sorts the documents by title, classification, author, and last-modification time, as shown in Figure 13.1.

Figure 13.1 Users see your catalog as categorized links

How does AutoCatalog work?

The AutoCatalog feature is actually controlled by an agent process called the *catalog agent*. The catalog agent accesses your server through HTTP requests. You either set up the catalog agent to run at set times, or you can manually run it from a form in the Server Manager. The catalog agent sends requests to your server until the catalog agent determines there are no more files to catalog.

The catalog agent gathers information in a two-step process. First it *enumerates* (gathers) the URLs referenced in each HTML file and determines which of these URLs it should catalog. Then it generates a *resource description* that contains information about the HTML file.

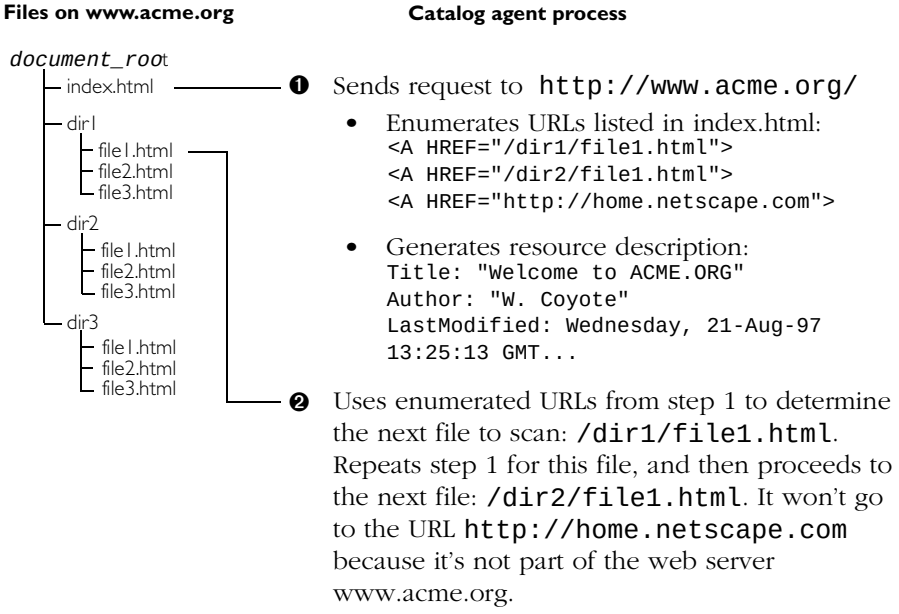
Enumerating the URLs

The catalog agent sends an HTTP request to your server and accesses the first URL you specify. Typically this is the URL to your home page, but you can set it to start in any directory or HTML file in your document root. The catalog agent gets the first HTML document and scans it for information to catalog.

Warning! If your server uses access control based on hostname or IP address, make sure you've allowed your local machine full access to your web site. Also, if your server is configured to use authentication through entries in a user database, make sure you give access to the catalog agent. See "The process.conf file" on page 260 for more information.

The first scan lists any URLs referenced in the HTML file. The second scan generates a resource description, as described in "Generating a resource description" on page 251. After the catalog agent enumerates the URLs and generates a resource description, it determines which HTML files to scan next. The catalog agent in Netscape Enterprise Server limits the URLs it traverses: it accesses only those HTML files located in your server. Figure 13.2 shows how the catalog agent scans the files on a sample web server.

Figure 13.2 The catalog agent enumerates URLS, and then generates resource descriptions



Generating a resource description

For each document found during enumeration, the catalog agent scans the HTML document for information to catalog. For example, the agent might gather the following information:

- **Title** is the text that appears within HTML `<TITLE>` tags.
- **Classification** is the text that appears in the CONTENT attribute for the HTML `<META>` tag. For example, "General HTML" is the Classification in the following text:

  ```
  <META NAME="Classification" CONTENT="General HTML">
  ```

- **Author** also appears in the CONTENT attribute for the HTML `<META>` tag. For example, "J. Doe" is the Author in the following text:

  ```
  <META NAME="Author" CONTENT="J. Doe">
  ```

- **Last-modified time** is the time the file was last saved. The catalog agent gets this information from the HTTP headers that the server sends, not from the HTML document itself.

After the catalog agent gathers this information from the first HTML file, it uses the enumerated URLs to choose which file to scan next.

Generating HTML catalog files

After the catalog agent gathers all of the information for your web server, it generates several HTML files that users will view to find information on your web site. These HTML files are kept in the `https-`*`identifier`*`/catalog` directory. Users can access the categorized information by going to the catalog directory on your server:

`http://`*`yourserver.org`*`/catalog`

You can restrict access to this directory and treat it as you do other parts of your web site. You can also turn off the catalog feature, which effectively means no one can access the catalog directory or any of its HTML files.

Using AutoCatalog

To use the AutoCatalog feature with your server, you first turn on the catalog agent and configure it to gather and sort the information about your web site. The catalog agent then collects information from HTML documents on your server, and it creates static HTML files that categorize your server's content in several ways.

Before any user can access the generated HTML files, you must turn on the catalog option. To let users access your server's catalog:

1. From the Server Manager, choose Auto Catalog|On/Off. The AutoCatalog On/Off form appears.
2. Click the Server On button.
3. Save and apply your changes.

See "Accessing catalog files" on page 258 for information on accessing the HTML files created by the catalog agent.

Configuring AutoCatalog

You can configure how the catalog agent accesses the content on your server. You can set directories in your document root where the catalog agent starts cataloging. That is, if you have several directories in your document root (the main directory where you keep your server's content files), you can set the catalog agent to access only certain directories and their subdirectories.

To configure the catalog agent:

1. From the Server Manager, choose AutoCatalog|Configure. The Configure Catalog Agent form appears.

To find your server's document root, choose the Primary Document Directory link in Content Mgmt.

2. Type the directories where you want the catalog agent to begin searching (it starts with the `index.html` file in that directory). For example, if your server's document directory has three subdirectories called *first*, *second*, and *third*, and you want the catalog agent to search only the second directory, type `/second` in the Starting Directories field.

If you leave the Starting Directories field blank, the catalog agent searches your home page first (this file is usually called `index.html`), and then it searches any URLs referenced in that file.

3. Select the speed at which the catalog agent should search your server's directories. The default is 7. The speed setting determines the number of "hits" the server will experience when the catalog agent is working. That is, when the catalog agent is searching through your server's files, it can simultaneously send the server one or more requests for documents. The catalog agent can also wait before sending a request to the server.

   In general, the speed setting should be appropriate for your server and its content. If you have a high-access server and up-to-date cataloging isn't very important, you should choose a low speed; if your server has low load times (perhaps in the early morning hours) and cataloging is very important to you, you should run the catalog agent at a high speed.

   Table 13.1 defines the speed settings.

Table 13.1 Speed settings

Speed setting	Simultaneous retrievals	Delay (seconds)
1	1	60
2	1	30
3	1	15
4	1	5
5	1	1
6	2	0
7	4	0
8	8	0
9	12	0
10	16	0

4. Enter the username and password that the agent will use to access any password-protected sources that are to be enumerated.

5. Click OK.

Scheduling the catalog agent

You can configure the catalog agent to run at specific times on specific days of the week. This feature is useful if your web site content changes frequently. For example, you might have a site where many people upload content and you don't directly monitor the content changes. Or, you might manage a site whose content is very dynamic and should be cataloged frequently.

Note If the content on your server changes infrequently or if all of the content changes simultaneously, you'll probably want to recatalog your content manually instead of scheduling times for recataloging. Manual recataloging minimizes the performance impact on your server.

To schedule the catalog agent:

1. From the Server Manager, choose Auto-Catalog|Schedule. The Schedule Catalog Agent form appears.
2. Select the hour and minute when you want the catalog agent to run. The drop-down lists let you choose a time in ten-minute increments.
3. Check the days of the week that you want the catalog agent to run. You can check one or more days.
4. Check Activate schedule. If you want to stop the server from cataloging your files on a schedule, check Deactivate schedule.
5. Click OK.

When the catalog agent runs, it logs its progress in a file called `robot.log`. This file appears in the `https-`*`identifier`*`/logs` directory under your server root directory. The log file contains the URLs retrieved, generated, and enumerated. This log file gives more detail than the status report (see "Getting a status report for the catalog agent" on page 256).

Controlling the catalog agent manually

You can control the catalog agent manually. This feature is useful for several reasons:

- You can improve performance by running the catalog agent only when it is necessary. For instance, you can run the catalog agent only after you've made significant changes to your server's content instead of running it on a weekly basis.
- You can minimize the catalog agent's impact on your server by running it during a low-access period. That is, you manually run the catalog agent when you know your server isn't being accessed by lots of people.
- You can stop the catalog agent if it is impacting your server's performance.

To manually control the catalog agent:

1. From the Server Manager, choose Auto Catalog|Manually Control. The Control Catalog Agent form appears.
2. Select one of the following buttons for controlling the catalog agent:

   Start starts the catalog agent using the settings in the Configure Catalog Agent form.

   Status displays the current status of the agent. See the following section for more information on status.

   Stop Enumeration stops the catalog agent from traversing files, but it continues generating the resource description for the file it's scanning.

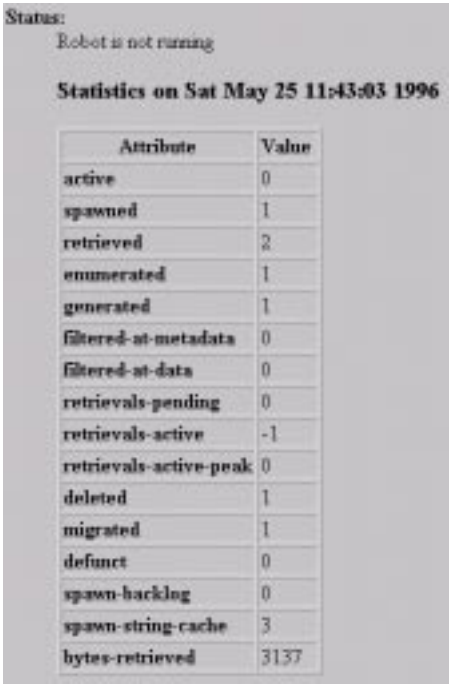
   Stop stops the catalog agent that you manually started. If the agent is in the middle of enumerating or generating a resource description, you'll lose that information, but the catalog agent will stop itself and clean up any temporary files it was using. You might use Stop Enumeration instead. The catalog agent will run again later if you scheduled the agent to run at specific times.

   Kill immediately stops the server. You'll lose any information the catalog agent was working on.

Getting a status report for the catalog agent

Whenever the catalog agent runs, you can get a status report that describes what the catalog agent is doing. To view the status report, click the Status button on the Control Catalog Agent form.

Figure 13.3 A sample status report for the catalog agent.


Status:

Robot is not running

Statistics on Sat May 25 11:43:03 1996

Attribute	Value
active	0
spawned	1
retrieved	2
enumerated	1
generated	1
filtered-at-metadata	0
filtered-at-data	0
retrievals-pending	0
retrievals-active	-1
retrievals-active-peak	0
deleted	1
migrated	1
defunct	0
spawn-backlog	0
spawn-string-cache	3
bytes-retrieved	3137

Table 13.2 defines all of the status attributes.

Table 13.2 Status attributes

Attribute	Description
active	The number of URLs the catalog agent is currently working on
spawned	The number of URLs the catalog agent has enumerated but hasn't yet retrieved
retrieved	The number of URLs retrieved through HTTP connections

Table 13.2 Status attributes

Attribute	Description
enumerated	The number or URLs enumerated so far
generated	The number or URLs generated so far
filtered-at-metadata	The number of URLs rejected by the catalog agent when scanning the META data in the HTML files
filtered-at-data	The number of URLs rejected by the catalog agent when scanning the data in the HTML files (for example, if the links reference an external host)
retrievals-pending	The number of URLs remaining that need to be retrieved
retrievals-active	The number of URLs the agent is currently retrieving
retrievals-active-peak	The highest number of URLs the agent simultaneously retrieved
deleted	The number of URLs filtered
migrated	The number of URLs enumerated but waiting to have resource descriptions processed
defunct	The number of URLs filtered
spawn-backlog	The number of URLs waiting to be processed by the catalog agent
spawn-string-cache	The number of unique host names that appeared in links
bytes-retrieved	The total number of bytes the catalog agent has retrieved, that is, the total number of bytes for all of the files the agent has retrieved through HTTP connections

Accessing catalog files

Once you have a working catalog, you can access the catalog main page at the following URL:

`http://yourserver.org/catalog`

Catalog files are kept on your server in a directory under the server root directory called `https-identifier/catalog`. Because this directory is outside your document root directory (where you keep all of your web content), the server creates an additional document directory that maps the URL prefix `/catalog` to the `https-identifier/catalog` directory on your hard disk. You can view this setting by choosing Content Mgmt | Additional Document Directories in the Server Manager.

Catalog configuration files

The catalog agent uses the following configuration files:

- The `filter.conf` file is used by the catalog agent to determine what data to save in the resource descriptions. This file also configures the catalog agent. You should modify this file only if you have Netscape Catalog Server. See that product's documentation for more information on this configuration file.
- The `process.conf` file configures the catalog agent and tells it where to send the resource descriptions that it generates. This file contains all of the catalog agent settings you specified in the Server Manager forms, including the URL where the catalog agent begins its enumeration.
- The `robot.conf` file specifies which `filter.conf` file the catalog agent uses.
- The `rdm.conf` file contains information for all catalogs served by the resource description server (RDS). RDSs collect resource descriptions from the robots that search the network and send this information to the catalog

server. The RDS is actually the back end of Netscape Catalog Server, and is not part of the autocatalog feature. You should modify this file only if you have Netscape Catalog Server and its documentation.

- The `csid.conf` file contains configuration information for the servers that the RDS catalogs.

The catalog agent also uses and obeys restrictions set in a file called `robots.txt`. You can use this file to restrict areas of your server from your catalog agent. This file is also used by any other robots or catalog agents that visit your web server.

The filter.conf file

The `filter.conf` file uses the same syntax as the `obj.conf` file. It is a series of directives and functions with attributes that define the rules the catalog agent follows (which directory to start cataloging) and how to generate the resource descriptions. The `filter.conf` file uses four directives:

- **Setup** initializes the catalog agent when the agent is started.
- **MetaData** filters the resource based on any meta-data listed in META tags in the HTML document. This filtering occurs once for each HTML file that the catalog agent retrieves.
- **Data** filters the HTML file based on the information sent in the HTTP headers for the file.
- **Shutdown** performs any functions needed before the catalog agent shuts down.

You should only modify this file if you plan to use your web server with Netscape Catalog Server. For more information on the configuration files, see the documentation for Netscape Catalog Server.

The process.conf file

The `process.conf` file configures the catalog agent. It includes information such as:

- Where to register results. A catalog service ID (CSID) points to the resource description server for your web server. The catalog agent sends the resource descriptions to this CSID. You can view your server's CSID on the AutoCatalog On/Off form.
- How fast to run. This is the speed setting you set in the Server Manager forms.
- How many system resources to use.
- A single username and password for authenticating to the server.

Example process.conf file

The following sample file shows how you can set a username and password that the catalog agent uses when authenticating to your server. The email address is also used to identify the catalog agent.

```
<Process csid="x-catalog://www.netscape.com:9999/AutoCatalog" \
   speed=10 \
   email="user@domain" \
   username="anonymous" \
   password="robin@" \
   http://www.netscape.com/
</Process>
```

The robots.txt file

The catalog agent is a type of *robot*—that is, it is a program that gathers information from your web site by recursively following links from one HTML file to another. There are many different kinds of robots that roam the World Wide Web looking for web servers to use for information gathering. For example, there are many companies that search the web, index documents, and then provide the information as a service to their customers (typically through searchable forms).

Robots are also sometimes called web crawlers or spiders.

Because some web administrators want to control what directories and files a robot can access, the web community designed a standard `robots.txt` file for excluding robots from web servers. The catalog agent was designed to follow instructions in a `robots.txt` file. However, not all web robots follow these guidelines.

You can use the `robots.txt` file to restrict your server's catalog agent, but if your web server is part of the World Wide Web, keep in mind that the `robots.txt` file might be used by other robots visiting your site.

Note The catalog agent, and any other robot, is restricted by access control settings and user authentication.

Format for robots.txt

The `robots.txt` file consists of one or more groups of lines with name-value pairs that instruct the robots. Each group of lines should describe the User-Agent type, which is the name of a particular robot. The Netscape catalog agent is called Netscape-Catalog-Agent/1.0. After you specify which User-Agents you want to configure, you include a Disallow line that lists the directories you want to restrict. You can include one or more groups in your `robots.txt` file.

Each line in the group has the format

```
"field:value"
```

The field name is not case-sensitive, but the value is case-sensitive. You can include comment lines by beginning the comment with the # character. The following example shows one group that configures all robots and tells them not to go into the directory called `/usr`:

```
# This is a sample robots.txt file
User-agent: *
Disallow: /usr
```

Example robot.txt files

The following example `robots.txt` file specifies that no robots should visit any URL starting with `/usr` or `/tmp`:

```
# robots.txt for http://www.mysite.com/
User-agent: *
Disallow: /usr
Disallow: /tmp
```

The next example restricts all robots from your web site except the Netscape catalog agent:

```
# robots.txt for http://www.site.com/
User-agent: *
Disallow: *
# Netscape catalog agent is a good robot
User-agent: Netscape-Catalog-Agent/1.0
Disallow:
```

The following example tells all robots, including the catalog agent, not to traverse your web site.

```
# No robots allowed!
User-agent: *
Disallow: /
```

Editing the robots.txt file

You can edit the `robots.txt` file manually or by using the online form. The Edit Robots.txt form will create a `robots.txt` file if one does not already exist. If you choose to edit the file manually, use the format described in "Format for robots.txt" on page 261. To edit the `robots.txt` file using the Edit Robots.txt form:

1. Choose Auto-Catalog|Edit Robots.txt in the Server Manager. The Edit Robots.txt form appears.

2. In the User-Agent field, enter the names of the User-Agents, or robots, you want to configure. Each User-Agent should be on a separate line in the text field. The User-Agents you list in this field are those for which you will be disallowing access to specific directories. The User-Agent names are case-sensitive.

   For example, if you want to configure the Netscape catalog agent, type `Netscape-Catalog-Agent/1.0` in the User-Agent field. If you want to configure all robots, you should enter `*`.

3. In the Disallow field, enter the names of the directories you want to restrict, listing each directory on a separate line. The directory names are case-sensitive also.

4. Click OK.

Appendix

A

HyperText Transfer Protocol

The HyperText Transfer Protocol (HTTP) is a protocol (a set of rules that describe how information is exchanged on a network) that allows a web browser and a web server to "talk" to each other using the ISO Latin1 alphabet, which is ASCII with extensions for European languages.

HTTP is based on a request/response model. The client connects to the server and sends a request to the server. The request contains the following: request method, URI, and protocol version. The client then sends some header information. The server's response includes the return of the protocol version, status code, followed by a header that contains server information, and then the requested data. The connection is then closed.

The Netscape Enterprise Server 3.0 supports HTTP 1.1. Previous versions of the server supported HTTP 1.0. The server is conditionally compliant with the HTTP 1.1 proposed standard, as approved by the Internet Engineering Steering Group (IESG) and the Internet Engineering Task Force (IETF) HTTP working group. For more information on the criteria for being conditionally compliant, see the Hypertext Transfer Protocol—HTTP/1.1 specification (RFC 2068) at:

```
http://www.ietf.org/html.charters/http-charter.html
```

This chapter provides a short introduction to a few HTTP basics. For more information on HTTP, see the IETF home page at `http://www.ietf.org/home.html`.

Requests

A request from a client to a server includes the following information:

- Request method
- Request header
- Request data

Request method

A client can request information using a number of methods. The commonly used methods include the following:

- `GET`—Requests the specified document
- `HEAD`—Requests only the header information for the document
- `POST`—Requests that the server accept some data from the client, such as form input for a CGI program
- `PUT`—Replaces the contents of a server's document with data from the client

Request header

The client can send header fields to the server. Most are optional. Some commonly used request headers are shown in Table A.1.

Table A.1 Common request headers

Request header	Description
`Accept`	The file types the client can accept.
`Authorization`	Used if the client wants to authenticate itself with a server; information such as the username and password are included.
`User-agent`	The name and version of the client software.

Table A.1 Common request headers

Request header	Description
`Referer`	The URL of the document where the user clicked on the link.
`Host`	The Internet host and port number of the resource being requested.

Request data

If the client has made a `POST` or `PUT` request, it can send data after the request header and a blank line. If the client sends a `GET` or `HEAD` request, there is no data to send; the client waits for the server's response.

Responses

The server's response includes the following:

- Status code
- Response header
- Response data

Status code

When a client makes a request, one item the server sends back is a status code, which is a three-digit numeric code. There are four categories of status codes:

- Status codes in the 100–199 range indicate a provisional response.
- Status codes in the 200–299 range indicate a successful transaction.
- Status codes in the 300–399 range are returned when the URL can't be retrieved because the requested document has moved.
- Status codes in the 400–499 range indicate the client has an error.
- Status codes of 500 and higher indicate that the server can't perform the request, or an error has occurred.

Table A.2 contains some common status codes.

Table A.2 Common HTTP status codes

Status code	Meaning
`200`	OK; successful transmission. This is not an error.
`302`	Found. Redirection to a new URL. The original URL has moved. This is not an error; most browsers will get the new page.
`304`	Use a local copy. If a browser already has a page in its cache, and the page is requested again, some browsers (such as Netscape Navigator) relay to the web server the "last-modified" timestamp on the browser's cached copy. If the copy on the server is not newer than the browser's copy, the server returns a 304 code instead of returning the page, reducing unnecessary network traffic. This is not an error.
`401`	Unauthorized. The user requested a document but didn't provide a valid username or password.
`403`	Forbidden. Access to this URL is forbidden.
`404`	Not found. The document requested isn't on the server. This code can also be sent if the server has been told to protect the document by telling unauthorized people that it doesn't exist.
`500`	Server error. A server-related error occurred. The server administrator should check the server's error log to see what happened.

Response header

The response header contains information about the server and information about the document that will follow. Common response headers are shown in Table A.3.

Table A.3 Common response headers

Response header	Description
`Server`	The name and version of the web server.
`Date`	The current date (in Greenwich Mean Time).

Table A.3 Common response headers

Response header	Description
`Last-modified`	The date when the document was last modified.
`Expires`	The date when the document expires.
`Content-length`	The length of the data that follows (in bytes).
`Content-type`	The MIME type of the following data.
`WWW-authenticate`	Used during authentication and includes information that tells the client software what is necessary for authentication (such as username and password).

Response data

The server sends a blank line after the last header field. The server then sends the document data.

Appendix

B

Using the internationalized server

The internationalized version of the Netscape Enterprise Server contains special features tailored for the non-U.S. environment. These features include a choice of user-interface language (Japanese, French, or German) and a choice of search engines that allow you to use text search on a variety of languages.

General information

The following information covers the international considerations for general server capabilities.

Installing the server

When you install the server, you choose what user-interface language to use, as well as what search engines to install.

For information on installing the international version of the server, see the `README` file.

Entering 8-bit text

If you want to type 8-bit data into the Server Manager or the administration server forms, you need to be aware of the issues in this section.

File or directory names

If a file or directory name is to appear in a URL, it cannot contain 8-bit or multi-byte characters.

LDAP users and groups

For email addresses, use only those characters permitted in RFC 822 (`ftp://ds.internic.net/rfc/rfc822.txt`). User ID and password information must be stored in ASCII.

If you use a local database, you can enter 8-bit and multi-byte characters, but you should standardize on one character set. If you use more than one character set in the same database, it can cause display and search problems.

If you must use 8-bit or multi-byte characters in your directory database, you should store them in UTF-8 for future compatibility with the Netscape Directory Server version 3.x. To make sure you enter characters in the correct format, use a UTF-8 form-capable client (such as Netscape Communicator) to input 8-bit or double-byte data.

If you let users access their own user and group information, they will need to use a UTF-8 form-capable client.

Using the accept language header

When clients contact a server using HTTP 1.1, they can send header information that describes the various languages they accept. You can configure your server to parse this language information.

For example, suppose this feature is set to *on*, and a client configured to send the accept language header sends it with the value `en,fr`. Now suppose that the client requests the following URL:

`http://www.`*`someplace`*`.com/`*`somepage`*`.html`

The server first looks for:

`http://www.someplace.com/en/somepage.html`

If it does not find that, it looks for:

`http://www.someplace.com/fr/somepage.html`

If *that* is not available either, and a `ClientLanguage` (call it `xx`) is defined in the `magnus.conf` file, the server tries:

`http://www.someplace.com/xx/somepage.html`

If none of these exist, the server tries:

`http://www.someplace.com/somepage.html`

For information about configuring the server to parse the accept language header, see "Parsing the accept language header" on page 55.

Language settings in configuration files

The following directives in the `magnus.conf` file affect languages:

Table B.1 International settings in magnus.conf

File	Directive	Values	Description
`magnus.conf`	`ClientLanguage`	`en`, `fr`, `de`, `ja`	Specifies the language in which client messages, such as "Not Found" or "Access denied" are to be expressed. This value is used to identify a directory containing `ns-https.db`.
`magnus.conf`	`DefaultLanguage`	`en`, `fr`, `de`, `ja`	Specifies the language used if a resource cannot be found for the client language or the administration language.
`magnus.conf`	`AcceptLanguage`	`on`, `off`	Enables or disables the Accept language header parsing.

The following directives in the `ns-admin.conf` file affect languages:

Table B.2 International settings in ns-admin.conf

File	Directive	Values	Description
`ns-admin.conf`	`ClientLanguage`	`en`, `fr`, `de`, `ja`	If the client does not send an accept language header, `ClientLanguage` defines the language of the Directory Server User Information and Password pages. The two-letter value code is used to find the directory containing `ns-admin.db`.
`ns-admin.conf`	`AdminLanguage`	`en`, `fr`, `de`, `ja`	Sets the language used for administrative pages that are accessed through the administration server.
`ns-admin.conf`	`DefaultLanguage`	`en`, `fr`, `de`, `ja`	The language used if a value cannot be found for the client or admin languages.

Server-side JavaScript information

When you use server-side JavaScript with the international version of the server, you have additional things to consider when compiling applications and using databases. For example, you can specify the language of the JavaScript application one of two ways: using the compiler, or using the HTML `<META>` tag.

Specifying the character set for the compiler

For the international version, the server-side JavaScript compiler (`jsac`) has a `-l` option called *`charSet`*. This option specifies the character set being used in the input HTML files. The value for *`charSet`* is one of the following character set names.

Table B.3 Valid values for charSet

Language	Value for *charSet*
Western European	`iso-8859-1`
Central European	`iso-8859-2`
Cyrillic	`iso-8859-5`
Japanese	`iso-2022-jp`, `x-sjis`, `x-euc-jp`
Korean	`iso-2022-kr`, `x-euc-kr`
Simplified Chinese	`x-gb2312`
Traditional Chinese	`x-big5`, `x-euc-ch`
Greek	`iso-8859-7`
Turkish	`iso-8859-9`

Usage To use this option, use the following format:

```
jsac [-cdv] [-l charSet] -o binaryFile [-i] inputFile1 [-i] inputFile2
...
jsac [-cdv] -o binaryFile -f includeFile
jsac -h
```

Options The following table shows the options for the compiler.

Table B.4 Options for the jsac compiler

Option	Usage
`-c`	Check only; do not generate *`binaryFile`*
`-v`	Enable verbose output
`-d`	Enable debug output

Table B.4 Options for the jsac compiler *(Continued)*

Option	Usage
`-o`	Name of *`binaryFile`* (output file).
`-i`	Name of *`inputFile`* (use if the input filename starts with a switch character)
`-f`	Name of *`includeFile`* (has input filenames, separated by white space)
`-l`	Name of *`charSet`* (for example, `iso-8859-1`, `x-sjis`, `euc-kr`)
`-h`	Display this help

The possible filename extensions are summarized in the following table:.

Table B.5 File extensions

Extension	File type
`.html` or `.htm`	HTML source file (may include JavaScript)
`.js`	JavaScript source file
`.web`	Binary output file

When you specify the language using the compiler option, you can only specify one language. If you want to specify multiple languages, you can use the `<META>` tag in the individual files.

Specifying the character set with the <META> tag

You can also use the `<META>` tag to specify the character set information. For example, if you put the following statement into the header (between `<HEAD>` and `</HEAD>`) in a JavaScript program, the server-side JavaScript compiler (`jsac`) considers the file to be written in `x-sjis`.

```
<META HTTP-EQUIV="Content-Type" CONTENT="test/html; CHARSET=x-sjis">
```

If the character set specified in the `<META>` tag is different from the character set specified by the compiler's *`charSet`* option, the character set specified by the compiler option is used.

Using server-side JavaScript with Oracle's Japanese database

To use server-side JavaScript with Oracle's Japanese database, follow these overall steps. Details for each step are in the following sections.

1. Install Oracle and set up your environment.
2. Verify the connection.
3. Verify the language setup.

Installing Oracle and setting up your environment

You must first install the Japanese Oracle database. For instructions, see the documentation that came with your database. Next, you must set up your environment variables using the following information. Environment variables for Oracle:

- `set ORACLE_HOME=`*`oracle_root`*
  for example, `D:\orant`
- `set ORACLE_SID=`*`oracle_service_ID`*
  for example, `ORCL`
- `set TNS_ADMIN=`*`path_to_tnsnames.ora`*
  for example, `D:\orant\...\tnsnames.ora`

Environment variables for NLS (National Language Support) in Oracle:

- `set NLS_LANG=`*`language_charset_info`*
  for example, `japanese_japan.JA16SJIS`

  (This example sets up `x-sjis`)

Environment variable for the path:

- `set PATH=`*`Enterprise_server_root`*`/bin/https;%ORACLE_HOME%/bin;%PATH%`

Restart the web server from the command line.

Verifying the connection

1. At the Application Manager, select and run `dbadmin`.
2. Click Connect to Database Server.
3. Enter the following information in the window, and click Connect. If your server identifier, user ID, or password is different from these default values, enter your actual values here.

Field	Value
Server Type	`ORACLE`
Server Identifier	`WG73`
User ID	`system`
Password	`manager`
Database	

Unless you see an error indicating otherwise, you are now connected.

Verifying the language setup

Use the `videoapp` sample application to verify the language setup.

1. If your ORACLE installation has a server identifier, user ID, or password that is different from the default values shown in the previous table, be sure to specify the actual values in the `start.htm` file at the following line:

```
project.sharedConnections.pool =
new DbPool("ORACLE","WG73", "system", "manager", "", 2, false)
```

2. Run the build script in the directory to recompile the JavaScript code.
3. At the Application Manager, select and run `vidoeapp`.
4. Click Add New Customer and enter data in the character set you specified.
5. Click Home to go back to the `videoapp` home page, and then click Save Changes.
6. Click Delete a Customer.

7. Check to see if the data you entered appears in the table. If the data appears in the database in the correct language, you've set up the languages correctly.

Putting the Oracle client and database server on separate hosts

To put the Oracle client (with server-side JavaScript database service) and the Oracle database server on separate hosts, follow these steps:

1. On the client side, define the `SERVER SID` alias to refer to the server in `tnsnames.ora`.

2. Set `the TWO_TASK` environment variable to the `SERVER SID` alias defined in the `tsnames.ora` file.

   For example:

   ```
   set TWO_TASK=SERVER SID alias
   ```

4. Set the `NLS_LANG` environment variable to the correct client language and character set information.

5. Using the sample application `videoapp`, edit the `start.htm` file as shown below. (In this example, assume that the `SERVER SID` alias is `remoteDB`.)

   ```
   project.sharedConnections.pool = new
   DbPool("ORACLE","remoteDB", "system", "manager", "",
   2, false)
   ```

6. Click Add New Customer and enter data in the character set you specified.

7. Click Home to go back to the `videoapp` home page, and then click Save Changes.

8. Click Delete a Customer.

9. Check to see if the data you entered appears in the table. If the data appears in the database correctly, you've configured your system properly.

Search information

Search capabilities are supported for the following languages:

- English
- German
- French
- Italian
- Spanish
- Swedish
- Dutch
- Japanese
- Korean
- Simplified Chinese
- Traditional Chinese

You choose which search engines to install when you install the international version of the server.

International search and auto catalog

If your server contains documents in various character set encodings, the search collections and/or auto catalog for the documents will inherit the same encodings as the originals. To view documents in different character set encodings, users must change the character set encoding for their browsers. In addition, since the text search and auto catalog features work with one character set encoding at a time, you might receive inaccurate results when using those features. Netscape recommends using one specific character set for all documents.

Searching in Chinese, Japanese, and Korean

The following information is specific to searching in Japanese, Korean, and Chinese.

Query operators

This release supports the following query operators for Japanese, Korean and Chinese languages:

Table B.6 Query operators for Japanese

Operator	J/C/K Character
AND	Yes
CONTAINS	No
ENDS	Yes
MATCHES	Yes
NEAR	Yes
NEAR/N	Yes
NOT	Yes
OR	Yes
PHRASE	Yes
STARTS	Yes
STEM	English only
SUBSTRING	Yes
WILDCARD *	Yes
WILDCARD ?	Yes
WILDCARD {}	No
WILDCARD []	No
WILDCARD ^	No

Table B.6 Query operators for Japanese

Operator	J/C/K Character
`WILDCARD -`	No
`WORD`	Yes

Document formats

This release supports the following document formats for the Japanese, Korean, and Chinese languages:

- HTML
- ASCII
- NEWS
- MAIL

Searching in Japanese

The following sections give additional information about searching in the Japanese character set.

Document codes

This release supports the following document codes for the Japanese language:

- `euc`
- `sjis`
- `jis` (7-bit)

Search words

This release supports the following search words:

- Kanji
- hirakana
- katakana (full-width and half-width)

- ascii-string (full-width and half-width)

The search engine translates half-width katakana to full-width katakana, and translates full-width ascii-string to half-width ascii-string. Users can use full-width and half-width as the same characters.

This release also supports phrase and sentence search.

Glossary

ACL Access Control List. A mechanism for defining which users have access to your server. You can define ACL rules that are specific to a particular file or directory, granting or denying access to one or more users and groups.

agent Software that runs the network-management software in a network device, such as a router, host, or X terminal. See also intelligent agents.

authentication Allows client to verify that they are connected to an SSL-enabled server, preventing another computer from impersonating the server or attempting to appear SSL-enabled when it isn't.

authorization The granting of access to an entire server or particular files and directories on it. Authorization can be restricted by criteria including hostnames and IP addresses.

browser See client.

cache A copy of original data that is stored locally. Cached data doesn't have to be retrieved from a remote server again when requested.

certification authority A third-party organization that issues digital files used for encrypted transactions.

certificate A nontransferable, nonforgeable, digital file issued from a third party that both communicating parties already trust.

CGI Common Gateway Interface. An interface by which external programs communicate with the HTTP server. Programs that are written to use CGI are called CGI programs or CGI scripts. CGI programs handle forms or parse output the server does not normally handle or parse.

ciphertext Information disguised by encryption, which only the intended recipient can decrypt.

client Software, such as Netscape Navigator, used to request and view World Wide Web material. Also known as a browser program.

collection A database that contains information about documents, such as word list and file properties. Collections are used by the search function to retrieve documents matching specified search criteria.

Common LogFile Format The format used by the server for entering information into the access logs. The format is the same among all major servers, including the Netscape FastTrack and Enterprise servers.

DHCP Dynamic Host Configuration Protocol. An Internet Proposed Standard Protocol that allows a system to dynamically assign an IP address to individual computers on a network.

DNS Domain Name System. The system that machines on a network use to associate standard IP addresses (such as 198.93.93.10) with hostnames (such as `www.netscape.com`). Machines normally get this translated information from a DNS server, or they look it up in tables maintained on their systems.

DNS alias A hostname that the DNS server knows points to a different host—specifically a DNS CNAME record. Machines always have one real name, but they can have one or more aliases. For example, an alias such as `www.`*`yourdomain.domain`* might point to a real machine called `realthing.`*`yourdomain.domain`* where the server currently exists.

document root A directory on the server machine that contains the files, images, and data you want to present to users accessing the server.

drop word See stop word.

encryption The process of transforming information so it can't be decrypted or read by anyone but the intended recipient.

expires header The expiration time of the returned document, specified by the remote server.

extranet An extension of a company's intranet onto the Internet, to allow customers, suppliers, and remote workers access to the data.

fancy indexing A method of indexing that provides more information than simple indexing. Fancy indexing displays a list of contents by name with file size, last modification date, and an icon reflecting file type. Because of this, fancy indexes might take longer than simple indexes for the client to load.

file extension The last part of a filename that typically defines the type of file. For example, in the filename `index.html` the file extension is `html`.

file type The format of a given file. For example, a graphics file doesn't have the same file type as a text file. File types are usually identified by the file extension (`.gif` or `.html`).

firewall A network configuration, usually both hardware and software, that protects networked computers within an organization from outside access. Firewalls are commonly used to protect information such as a network's email and data files within a physical building or organization site.

flexible log format A format used by the server for entering information into the access logs.

FTP File Transfer Protocol. An Internet protocol that allows files to be transferred from one computer to another over a network.

GIF Graphics Interchange Format. A cross-platform image format originally created by CompuServe. GIF files are usually much smaller in size than other graphic file types (BMP, TIFF). GIF is one of the most common interchange formats. GIF images are readily viewable on Unix, Microsoft Windows, and Apple Macintosh systems.

hard restart The termination of a service and its subsequent restart. See also soft restart.

home page A document that exists on the server and acts as a catalog or entry point for the server's contents. The location of this document is defined within the server's configuration files.

hostname A name for a machine in the form *`machine.domain.dom`*, which is translated into an IP address. For example, `www.netscape.com` is the machine `www` in the subdomain `netscape` and `com` domain.

HTML Hypertext Markup Language. A formatting language used for documents on the World Wide Web. HTML files are plain text files with formatting codes that tell browsers such as the Netscape Navigator how to display text, position graphics and form items, and display links to other pages.

HTTP HyperText Transfer Protocol. The method for exchanging information between HTTP servers and clients.

HTTP-NG The next generation of HyperText Transfer Protocol.

HTTPD An abbreviation for the HTTP service, a program that serves information using the HTTP protocol. The Netscape Enterprise Server is often called an HTTPD.

HTTPS A secure version of HTTP, implemented using the Secure Sockets Layer, SSL.

imagemap A process that makes areas of an image active, letting users navigate and obtain information by clicking the different regions of the image with a mouse. Imagemap can also refer to a CGI program called "imagemap," which is used to handle imagemap functionality in other HTTPD implementations.

intelligent agent An object within a server that performs various requests (such as HTTP, NNTP, SMTP, and FTP requests) on behalf of the user. In a sense, the intelligent agent acts as a client to the server, making requests that the server fulfills.

IP address Internet Protocol address. A set of numbers, separated by dots, that specifies the actual location of a machine on the Internet (for example, 198.93.93.10).

ISDN Integrated Services Digital Network.

ISINDEX An HTML tag that turns on searching in the client. Documents can use a network navigator's capabilities to accept a search string and send it to the server to access a searchable index without using forms. In order to use <ISINDEX>, you must create a query handler.

ISMAP ISMAP is an extension to the IMG SRC tag used in an HTML document to tell the server that the named image is an imagemap.

ISP Internet Service Provider. An organization that provides Internet connectivity.

Java An object-oriented programming language created by Sun Microsystems used to create real-time, interactive programs called applets.

JavaScript A compact, object-based scripting language for developing client and server Internet applications.

last-modified header The last modification time of the document file, returned in the HTTP response from the server.

MD5 A message digest algorithm by RSA Data Security. MD5 can be used to produce a short digest of data that is unique with high probability. It is mathematically extremely hard to produce a piece of data that produces the same message digest email.

MD5 signature A message digest produced by the MD5 algorithm.

MIB Management Information Base.

MIME Multi-Purpose Internet Mail Extensions. An emerging standard for multimedia email and messaging.

MTA Message Transfer Agent. You must define your server's MTA Host to use agent services on your server.

NNTP Network News Transfer Protocol for newsgroups. You must define your news server host to use agent services on your server.

NSAPI See Server Plug-in API.

primary document directory See document root.

protocol A set of rules that describes how devices on a network exchange information.

private key The decryption key used in public-key encryption.

public key The encryption key used in public-key encryption.

RAM Random access memory. The physical semiconductor-based memory in a computer.

redirection A system by which clients accessing a particular URL are sent to a different location, either on the same server or on a different server. This system is useful if a resource has moved and you want the clients to use the new location transparently. It's also used to maintain the integrity of relative links when directories are accessed without a trailing slash.

resource Any document (URL), directory, or program that the server can access and send to a client that requests it.

RFC Request For Comments. Usually, procedures or standards documents submitted to the Internet community. People can send comments on the technologies before they become accepted standards.

server daemon A process that, once running, listens for and accepts requests from clients.

Server Plug-in API An extension that allows you to extend and/or customize the core functionality of Netscape servers and provide a scalable, efficient mechanism for building interfaces between the HTTP server and back-end applications. Also known as NSAPI.

server root A directory on the server machine dedicated to holding the server program, configuration, maintenance, and information files.

simple index The opposite of fancy indexing—this type of directory listing displays only the names of the files without any graphical elements.

SNMP Simple Network Management Protocol.

SOCKS Firewall software that establishes a connection from inside a firewall to the outside when direct connection would otherwise be prevented by the firewall software or hardware (for example, the router configuration).

soft restart A way to restart the server that causes the server to internally restart, that is, reread its configuration files. A soft restart sends the process the `HUP` signal (signal number one). The process itself does not die, as it does in a hard restart.

SSL Secure Sockets Layer. A software library establishing a secure connection between two parties (client and server) used to implement HTTPS, the secure version of HTTP.

stop word A word identified to the search function as a word not to search on. This typically includes such words as *the*, *a*, *an*, *and*. Also referred to as *drop words*.

strftime A function that converts a date and a time to a string. It's used by the server when appending trailers. `strftime` has a special format language for the date and time that the server can use in a trailer to illustrate a file's last-modified date.

TCP/IP Transmission Control Protocol/Internet Protocol. The main network protocol for the Internet and for enterprise (company) networks.

telnet A protocol where two machines on the network are connected to each other and support terminal emulation for remote login.

timeout A specified time after which the server should give up trying to finish a service routine that appears hung.

top-level domain authority The highest category of hostname classification, usually signifying either the type of organization the domain is (for example, `.com` is a company, `.edu` is an educational institution) or the country of its origin (for example, `.us` is the United States, `.jp` is Japan, `.au` is Australia, `.fi` is Finland).

URI Uniform Resource Identifier. A file identifier that provides an additional layer of security by using an abbreviated URL. The first part of the URL is substituted with a URL mapping that hides the file's full physical pathname from the user. See also URL mapping.

URL Uniform Resource Locator. The addressing system used by the server and the client to request documents. A URL is often called a location. The format of a URL is *`protocol://machine:port/document`*.

A sample URL is `http://www.netscape.com/index.html`.

URL database repair A process that repairs and updates a URL database that has been damaged by a software failure, a system crash, a disk breakdown, or a full file system.

URL mapping The process of mapping a document directory's physical pathname to a user-defined alias so that files within the directory need only refer to the directory's alias instead of the file's full physical pathname. Thus, instead of identifying a file as `C:\Netscape\SuiteSpot\docs\index.html`, you could identify the file as `/myDocs/index.html`. This provides additional security for a server by eliminating the need for users to know the physical location of server files.

web publishing The capability of server clients to access and manipulate server files, editing and publishing documents remotely. Web publishing provides document version control, link management, search, access control, and agent services to server users.

Web Application Interface (WAI) An easy-to-program mechanism for extending the Enterprise server's functionality with CORBA-compliant services that are tightly integrated with the web server. WAI can be used to compose services in C, C++, and Java that customize the functionality of the server.

Windows CGI CGI programs written in a Windows-based programming language such as Visual Basic.

Index

Symbols

A

B

C

D

E

F

G

H

I

O

P

Q

R

S

T

U

V

W